A Literature Review of design trade offs in bioinformatics

Luke Darlow
Supervisor: Phillip Machanick

May 27, 2013

Computer Science Department, Rhodes University

Abstract

Bioinformatics is a fast advancing multi-faceted area of research. Many tool suites exist and are charged with the task of processing the accumulating wealth of knowledge welling up with the discovery of the DNA molecule. The importance of this discovery as an insight into the building blocks of life cannot be underestimated. The question of design trade-offs in bioinformatics web services is posed with consideration of the pros and cons of a number of tool suites and their use by anybody desiring to do efficient, reproducible, and reliable research. Solutions for advancement into the public EC2 cloud and possible use of distributed computing in a transparent fashion are discussed. Utilising web scraping techniques to harvest input sequence information from published research is discussed as a usability advance.

Contents

1 Introduction

1.1 What is biology to a computer scientist?

DNA (deoxyribonucleic acid) is a molecule consisting of a combination of paired bases bonded to a sugar phosphate.

There are four possible bases, namely:

- Adenine
- Theymine
- Cytosine
- Guanine

Most DNA is found in the nucleus of the cell. However, there is a small portion residing in the mitochondrial DNA. This mitochondrial DNA is found within the mitochondria of cells. Mitochondria are involved in energy production of cells [22].

Approximately 99% of DNA is identical between two human beings. The remaining 1% of dissimilar DNA is what is used in paternity tests and the like [12].

It is DNA that is the determining factor of heredity and consequently how an organism self-replicates in the monumental fashion resulting in the flow and balance of life. Most of DNA processing has been concerned with DNA in terms of protein construction. This is the main coding function of DNA, as understood presently. It cannot decisively be said this is the only function.

Another concept worth understanding is that of the **transcription factor**. A transcription factor is a protein acting on DNA to influence the flow of genetic information. This influences the production of RNA and therefore the proteins that make up an organism.

Bayat [10] defines bioinformatics as :

> "The application of tools of computation and analysis to the capture and interpretation of biological data."

DNA is comprised of the four letter alphabet (A, T, C and G). Protein is comprised of a twenty letter alphabet. Biologists perform a process known as sequencing to analyse DNA and/or protein to output a sequence. This sequence is linear and (since the representation is simply using an alphabet of letters) results in a large string. This is the point whereby computer science becomes useful. The problem at hand is essentially one

of string searching, known more specifically as the Approximate Common String (ACS) problem. The biological sequences are the strings for the ACS problem [3].

Computer scientists are able to develop the tools needed to do the large scale processing and management of raw data (produced by molecular biologists) into a manageable, useful and user-friendly form. This hastens the process of understanding in appropriate fields. Not only are the techniques and algorithms an important aspect of this, but so is the interpretation of data and the implementation of tools that make this information more available and user friendly.

1.2 Bioinformatics - what is it?

Computer science is a rapid growing discipline naturally lending itself to avenues of thought applicable in multiple other disciplines [18]. The fundamental principles of abstraction and automation [32] associated with computational thinking are inherently powerful devices for the solving of otherwise overwhelmingly complex problems. Problems involving massive data sets, pattern matching, large statistical analysis and multi-database lookups can be managed effectively and efficiently using tools developed within the realm of computer science. A solid core understanding of manipulating data and the processes and algorithms involved (in both the abstracted, higher level and the lower level) is crucial in elegantly using a computer to solve this sort of problem.

The style of problem able to be solved with thanks to computer science suites the style of problem present in many other sciences. Molecular biology is an example of a science marrying very well with computer science. Molecular biology concerns itself with the structure, characteristics and chemical processes involved in living cells. These chemical processes result in the formation and self-creation of an organism on a higher level (considering the "one-way street" flow widely believed to be followed in molecular construction; genes are transcribed into RNA and RNA is translated into DNA). The discovery of DNA by James Watson and Francis Crick published in a scientific paper in 1953 resulted in a new area of research with absolutely profound implications.

The key to the link between biology and computer science is the digital nature of the DNA molecule. These building blocks of life serve to comprise the RNA alphabet living organisms use to self-create. The presence of this along with the precise mapping between the amino acid building blocks of protein means computer science, in theory (considering the noise present through the biochemical processes involved), possesses the tools and consequently the potential for tools to store, analyze and process DNA. [13]

The emergence of informational biology and bioinformatics follows naturally as a result of the tools, techniques and processing ability needed from computer science in this quickly developing area of research. The Human Genome Project was completed in 2003 and became a major contributor to the swift advance of bioinformatics (in terms of popularity and tools). The goals of the Human Genome Project included (amongst others) the development and enhancement of sequencing technology. This can be attributed to sparking much research into novel, state-of-the-art techniques involved in bioinformatics.

Bioinformatics is a rapidly advancing area of research and requires the combined effort of members of different schools of study in order to not only gather and process the vast wealth of information constantly collected but also to do so in an efficient and conducive manner.

This is all good and well (considering purely academic endeavors) but when the question of functionality and use is considered, access to and easy use of bioinformatics tools is paramount. The MEME (Multiple EM for Motif Elicitation) Suite [6] tool set for motif discovery and searching is a standalone or web server based tool set designed for use by anybody wishing to do so. The Galaxy project acknowledges computation as an indispensable tool in life science research and provides an open platform for accessible and reproducible web-based tools [14]. RSAT (Regulatory Sequence Analysis Tools) is yet another example of a software suite tending the needs arising in bioinformatics.

These tools, constantly under scrutiny and consequently being improved, all have their flaws and space for growth. A common trend in the use of said tools is the gap of understanding between computer scientists (the designers and implementers of these tools) and biologists (the users). Whether considering the quirks of software use (command line parameters, for example) [14, 6] or the need for a specific format input (essentially a computer science 'speed bump' to be overcome), there is still much space for improvement.

2 The MEME Suite

What is the MEME Suite?

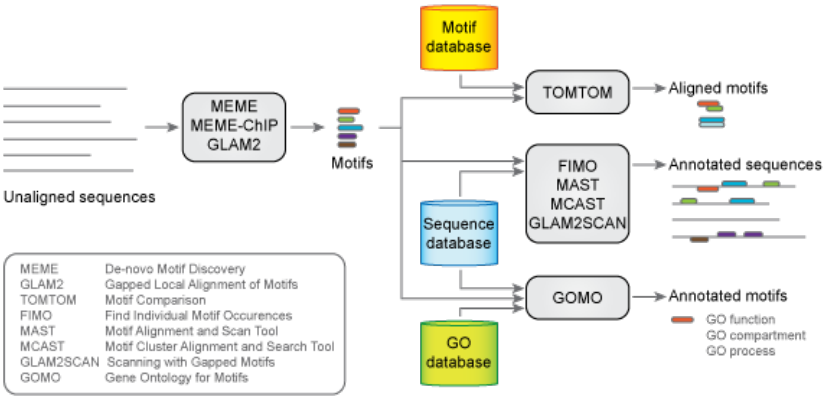

Figure 1: A visually descriptive representation of the MEME Suite outlining the use and workflow of the tools involved [6].

The work by Bailey et al. [6] of the MEME Suite is a collaborated collection of tools for discovering, analyzing, comparing and characterizing sequence motifs in DNA or protein sequences. A motif is a statistically significant reoccurring pattern within the sequence data and is functionally significant in molecular evolution. This suite of tools can be installed locally or on a web server. The flagship tool in the meme suite is MEME. MEME discovers gapless sequence motifs by searching for statistically relevant motifs within the supplied unaligned sequences. Other tools in the suite for motif discovery are DREME and Glam2.

Tools in the MEME Suite for use in motif search are FIMO, Glam2Scan, MAST, and MCAST. These (along with other, additional tools) will be discussed. Figure 1 shows a workflow diagram from the MEME web site depicting how these interrelated tools work together to provide useful information to the user.

2.1 MEME

MEME searches for statistically significant motifs in unaligned and related input sequences. It is one of the most widely used bioinformatics tools. The discovery of novel signals within these sequences has many uses in the academic and medical worlds - finding numerous similar motifs in multiple sequences is a good indication these sequences share some biological function [7]. The discovery of transcription factor binding sites is but one of many uses for this tool [9].

The MEME algorithm

MEME makes use of a statistical algorithm called Expectation Maximization. A motif length and initial estimation of the number of sites are given. Possible sites are found and sorted according to this Expectation Maximization. A quantity known as the E-value (a parameter indicative of the probablity of finding a hit at random - p-value scaled by the size of the data) is then computed for each site. This is repeated for different input parameters (motif length and number of occurrences).

The MEME algorithm has the advantages of not requiring each sequence to contain a motif and does not need a prior classification of what motif(s) a sequence contains in order to find motifs. MEME can still find a motif if only 20% of the sequences contain the motif. This unsupervised learning characteristic of MEME makes it a flexible and powerful tool.

The motif with the lowest E-value is output and then taken out of the training set (making use of a soft-erase function; probabilistically removing the occurrences of the found motif). This is repeated until such a point as either the number of motifs is reached or the minimum E-value reaches threshold [4, 7].

A downfall of this algorithm is its inability to find gapped motifs. This would be much more computationally intensive and is not warranted as of yet. Another possible downfall is inherent in statistical expectation maximization algorithms. This is their tendency to converge on local optima. MEME uses heurisitics to overcome this problem [7].

MEME input

The data submission form for MEME is shown in Figure 2. Along with an email address and the file containing the input sequences, command line paramenters for the MEME algorithm can be set here. The necessity for sequences to be input in the FASTA format and the limited functionality of this is a hinderance because many papers publish data as genomic coordinates rather than sequences in this format.

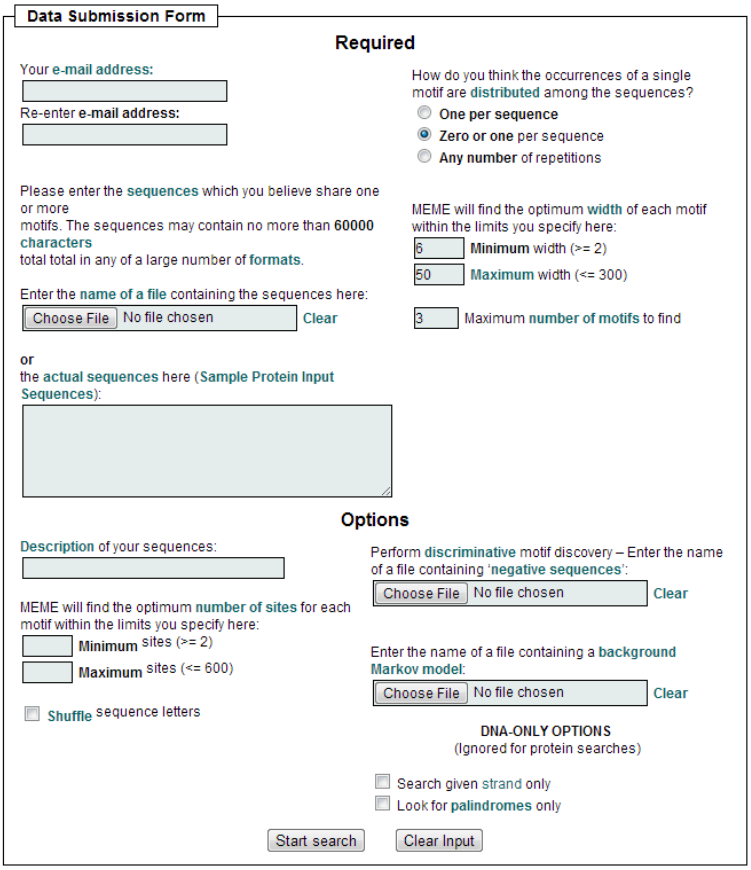

Figure 2: An example MEME submission form [6].

MEME output

The MEME Suite web service output is comprehensive and useful. All discovered motifs are displayed with an option to submit motifs (either individually or as a group) for further analysis (see figure 3, below). Motifs are represented as position-dependant scoring matrices describing the score of each letter (A, C, T, and G) at that position.

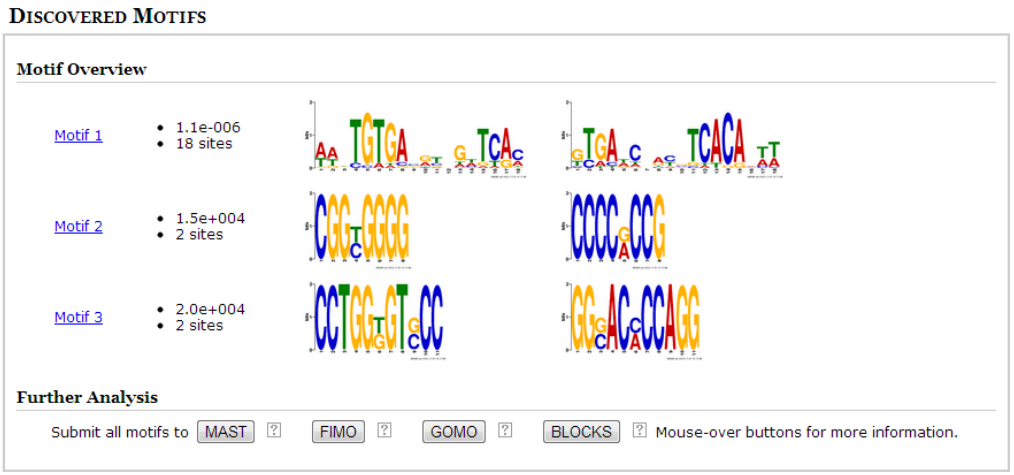

Figure 3: Example output of MEME showing three discovered motfs [6].

A regular expression is generated describing the motif. This regular expression can be used by other tools for searching for this discovered motif in other sequences.

Each motif has a corresponding summary table (see figure 4 below). This shows the E-value, the width of each motif, the number of sites needed in constructing this motif, the Log Likelihood Ratio (essentially a measure of likelihood considering noise and background models), Information Content (a measure of entropy with respect to the background model), and relative entropy [2].

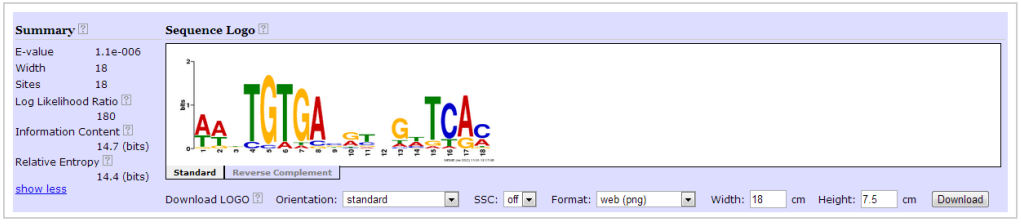

Figure 4: Example output of a single motif with its corresponding summary table [6].

Further visual representation of the motifs is given in the forms of block diagrams and a visualisation of all the occurrences of motifs in the input sequences in which they occur. See figure 5 and 6 below for an example of this.

Sites

Click on any row to highlight sequence in all motifs.

Name	Strand	Start	*p*-value	Sites
ara	-	58	2.51e-07	TGGCATAGCA AAGTGTGACGCCGTGCAA ATAATCAATG
lac	+	8	5.35e-07	AACGCAAT TAATGTGAGTTAGCTCAC TCATTAGGCA
malt	+	40	8.61e-07	AAAGATTTGG AATTGTGACACAGTGCAA ATTCAGACAC
ilv	-	42	1.69e-06	GCAAAGGGAA AATTGAGGGGTTGATCAC GTTTTGTACT
pbr322	-	56	2.85e-06	CTCCTTACGC ATCTGTGCGGTATTTCAC ACCGCATATG
deop2	+	59	2.85e-06	AGATTTCCTT AATTGTGATGTGTATCGA AGTGTGTTGC
uxu1	+	16	5.17e-06	AGAGTGAAAT TGTTGTGATGTGGTTAAC CCAATTAGAA
trn9cat	+	83	5.69e-06	CTTTTGGCGA AAATGAGACGTTGATCGG CACG
ce1cg	-	64	7.54e-06	GGACTTCCAT TTTTGTGAAAACGATCAA AAAAACAGTC
ompa	+	47	9.04e-06	TTTTTTTCAT ATGCCTGACGGAGTTCAC ACTTGTAAGT
crp	-	66	9.89e-06	TACTGCACGG TAATGTGACGTCCTTTGC ATACATGCAG
male	+	13	1.29e-05	TTACCGCCAA TTCTGTAACAGAGATCAC ACAAAGCGAC
galE	-	45	1.41e-05	AAGATGCGAA AAGTGTGACATGGAATAA ATTAGTGGAA
tdc	-	81	1.53e-05	AACAGG ATATGTGCGACCACTCAC AAATTAACTT
malk	+	60	1.67e-05	ATGTAAGGAA TTTCGTGATGTTGCTTGC AAAAATCGTG
cya	+	49	1.81e-05	TCAATCAGCA AGGTGTTAAATTGATCAC GTTTTAGACC
tnaa	+	70	2.73e-05	CTCCCCGAAC GATTGTGATTCGATTCAC ATTTAAACAA
bglr1	+	75	8.30e-05	CAAAGTTAAT AACTGTGAGCATGGTCAT ATTTTTATCA

Figure 5: Example output of the sites of a single motif. The first column is the name of the sequence. If the sequence strand is denoted '+' it means the motif is found as is in the training set; a sequence denoted '-' shows that the reverse compliment is found. The sites are arranged in order of increasing p-value. The p-value is a measure of statistical significance [6].

Block Diagrams

The height of the motif "block" is proportional to -log(p-value), truncated at the height for a motif with a p-value of 1e-10.
Click on any row to highlight sequence in all motifs. Mouse over the center of the motif blocks to see more information.

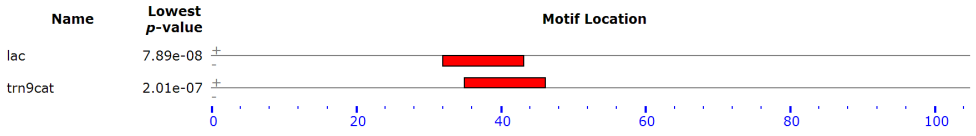

Figure 6: Example output of a block diagram. These are arranged in the same order of the input [6].

At this point in the users experience of the MEME Suite web service the next obvious step would be to take these discovered motifs and perform further analysis on them. The HTML output provides buttons to submit the output MEME data to MAST, FIMO, TOMTOM, GOMO, and BLOCKS.

2.2 MAST

MAST (Motif Alignment and Search Tool) takes as input motif(s) represented as position-dependant scoring matrices along with a specified database or databases for comparison. The comparison database(s) are generally relatively short. These input motifs can originate from databases or directly from the user but more usefully can be direct output from MEME. Input must be ungapped (suitable for output from MEME). MAST proceeds to produce output detailing the top scoring motifs [8]. A visual output is shown in figure 7 below.

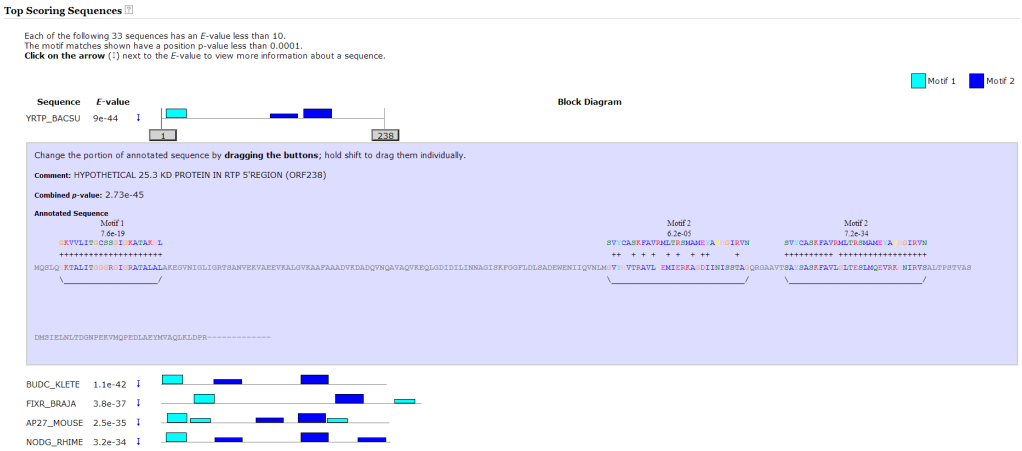

Figure 7: Example output for MAST. The position of each block gives an indication of where the match was found. The width of each block shows the width of the motif relative to the sequence considered. The height gives an indication of the significance of the find [6].

2.3 FIMO

FIMO (Find Individual Motif Occurrences) is another tool for scanning sequences for the occurrence of one or more motifs. FIMO can scan DNA or protein sequence databases. Although it is not the first of its kind, it outdoes similar tools in many respects. For example, RSAT [31] fails with regard to scanning proteins.

Input into FIMO consists of one or more motifs (thus seamlessly integrating with MEME) and either user-supplied sequences or a database to search. FIMO proceeds

to search for statistically significant occurrences of each motif. It does this by scoring each position in the searched sequence with a log-likelihood ratio. It also makes use of dynamic false discovery rates. Output from FIMO consists of statistically ranked motif occurrences. Output is provided in formats available as a starting poing for input into the UCSC Genome Browser [15].

2.4 DREME

A problem facing most motif discovery tools is the difficulty of searching large datasets. DREME is a novel motif search tool tailored for ChIP-seq data (chromatin immunoprecipitation followed by high throughput sequencing) experiments as these yield an extremely large number of predictions for TFBS's. What makes the DREME algorithm unique is its linear scalability with regard to large datasets (large sets of short sequences), allowing DREME to discover primary and cofactor ('helper') motifs otherwise overlooked by other tools (which selectively only use some of the data yielded by ChIP-seq experiments).

DREME is limited to finding motifs of length up to eight base pairs wide. This means it may miss information rich wider motifs. Although this is a downfall, the motivation for this choice is rooted in DREME not being a replacement tool for motif discovery but rather a complimentary tool. It is able to quickly discover overlooked motifs.

DREME takes as its input two sets of sequences (a positive and negative set - if no negative set is given the positive set is shuffled to provide this contrasting set) and a significance threshold which is used in the algorithm. Output consists of the discovered motifs, their logos, their reverse compliment logos, significant statistical data (a major good point for DREME; allowing for biologists to distinguish between statistical artefacts and actual functional motifs), an option to download, and an option to submit the output for further analysis. See figure 8 below for an example of this [5, 20] .

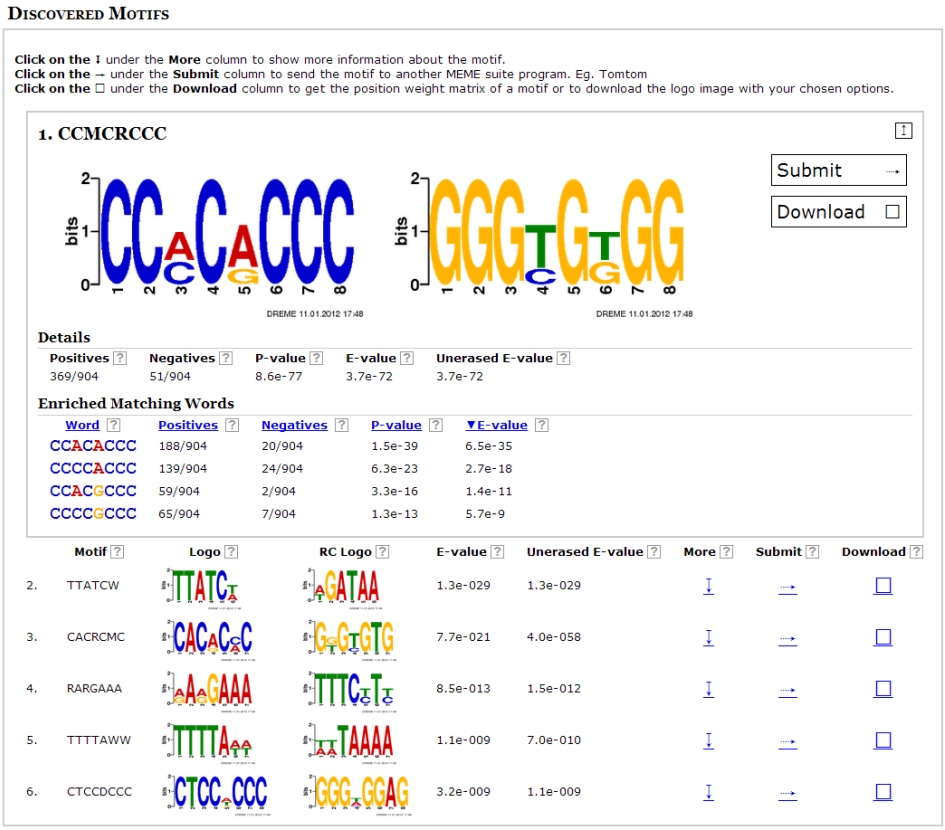

Figure 8: Example output for DREME. The first found motif is viewed with detail by clicking the down arrow in the 'More' column [6].

2.5 Installation and use procedure

The MEME Suite is a collection of command line tools with the option of a web service. Any user is able to install this web service. Anybody with some knowledge of programming can use the MEME tools in a pipeline fashion. Galaxy (discussed in section 3.1) uses a visual workflow environment to allow for web service users to do this in a high level manner and would be a suitable direction for the MEME Suite to head in in order to achieve better usability. A number of complications relating to the tying together of the MEME tools arise at this point. A number of command line parameters must be provided for the tools to function optimally (see below for an example thereof).

```
meme crp0.s -dna -mod zoops -nmotifs 3 -revcomp
```

The web service has an overlaying interface for this (see figure 2) but for complete flexibility of use a version must be installed locally. The MEME Suite has extensive documentation regarding this procedure (found at http://meme.nbcr.net/meme/doc/meme-install.html) with options for parellel installs, customized installations, and the installation of the web server.

2.6 Additional tools

There exist many additional tools for sequence analysis in the MEME Suite. These range in uses for visualisation to analysis on gapped motifs (and their visualisation) to motif enrichment analysis (SpaMo and CentriMo). Within the MEME Suite tools the input often consists (in part, at least) of sequence data and/or motifs. This sequence data has to be in the FASTA format (see section 4 for details).

The improvement of tools for use in bioinformatics is not one of competition for the love of money. The open source nature of most tools show the intent of those who make use of them. This is an environment for research and advance in technology for good. A good look at other tools and web services is warranted in this scope - their novel approaches, advantages and downfalls shall be considered for improvement and advance.

3 Other bioinformatics tool suites

3.1 Galaxy

Galaxy takes a different approach to bridging the gap between computer scientists and biologists. Galaxy places focus on accessibility, reproducibility and (the often underestimated element of) transparency in the context of software development. Many genomic experiemnts face the issue of reproducibility. Experimental reproducibility is essential for valid scientific research. The complex nature of experiments (with many involved steps, detailed and specific paramaters and chaining of computational tools) makes them difficult to reproduce without a framework for doing this. Galaxy proposes a Reproducible Research System which begs questions of input, output and representation but provides promise for easier and more widespread use of the accessible bioinformatics tools.

Galaxy is a open web based platform for use by those without programming skill (most biologists, for example). This platform provides a mechanism for command line parameter input, tool chaining and visualisation. Galaxy also allows for the installation of other tools - these are required to run on a command line; developers specify documentation on input and output parameters with galaxy creating a workbench environment for their use. This trades flexibility for user friendliness.

User friendliness is not the crux of the Galaxy's implementation. Analysis and methods involved are arguabily of more importance than simple output figures. Galaxy pro-

vides means of output designed for detailed, multi-layered analysis of experiments performed. This output takes the form of Galaxy pages with dynamic workflows and history analysis. See figure 9 for an example of these.

Galaxy makes use of metadata capture on datasets, tools and parameters in an automatic fashion. Although this metadata can be used to accurately reproduce and reuse an experiment, intent gets lost without the experimenter's knowledge. User descriptions of steps in the history of the experimentation provide a means to annotate and give reasoning for method and parameter choice (amongst other things). Another utility implemented by galaxy is the use of tags. These tags can be defined by Galaxy users and serve as a system for searching experiments, histories, and workflows.

The workflow editor within Galaxy provides a dynamic means of chaining tools and keeping track of this for later use and reproduction. The choice of workflows works but can tend to be obscure in some cases.

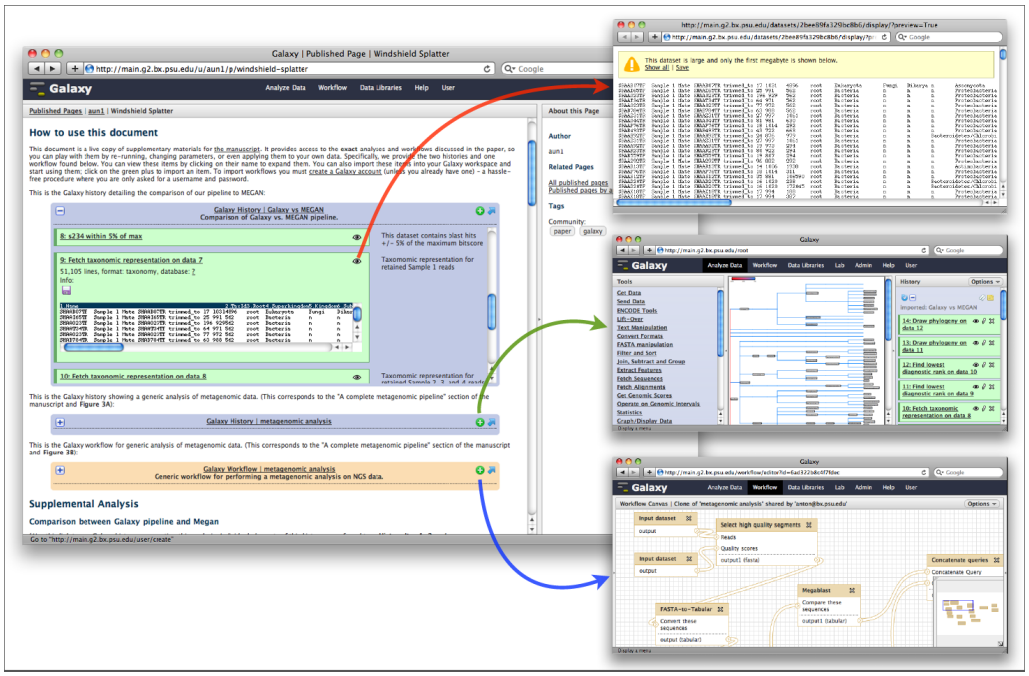

Figure 9: A Galaxy page example. Top right shows input taxonomy data. Middle right shows a history panel. Bottom right shows a workflow [14].

Galaxy tools can be split into three categories: query operations on datasets, sequence analysis tools, and output display tools. The core of galaxy does the job of binding all of these together in the attempt to fulfil the goals of accessibility, reproducibility and transparency. With approximately five thousand jobs processed daily, Galaxy is making headway [14].

3.2 RSAT

Regulatory Sequence Analysis Tools (RSAT) is a cluster of sequence discovery and search tools aimed at *cis*-regulatory modules of DNA. *Cis*-regulatory modules are sequences of DNA known to include a number of TFBS's related to gene expression and are particularly relevant due to their functional component.

RSAT has a series of tools for both pattern discovery and matching. An earlier edition only had the ability to do string matching but the current edition now uses position-specific scoring matrices (similar to MEME) along with a background model for genomic noise. This background model is very important for the correct functioning of the algorithm. RSAT supports over 600 genomes and has pre-computed background models for these.

Without a good background model much of the RSAT may fail. Making use of these background models, however, allows for random control tests to assess the presence of false positives. RSAT also has drawing facilities and a web service implementation.

The web service allows for users to submit input and receive output using a tool installed on the web server but has no interface for chaining tools together. For this the user needs to have some basic programming skill. A future effort to provide a web interface for tool chaining is needed. As with most bioinformatics tools (implementing a web service worth considering) documentation and demos are available for use of the tools in the suite.

Figure 10 shows an example of the multitude of tools existing in RSAT for searching and discovery. This stands in contrast with many other bioinformatics web services (with focus on a single pattern matching algorithm) but begs the question of the level of refinement for each tool in this suite. Future efforts to improve RSAT would include increased flexibility with other tools and databases and a means to rope tools together efficiently [31].

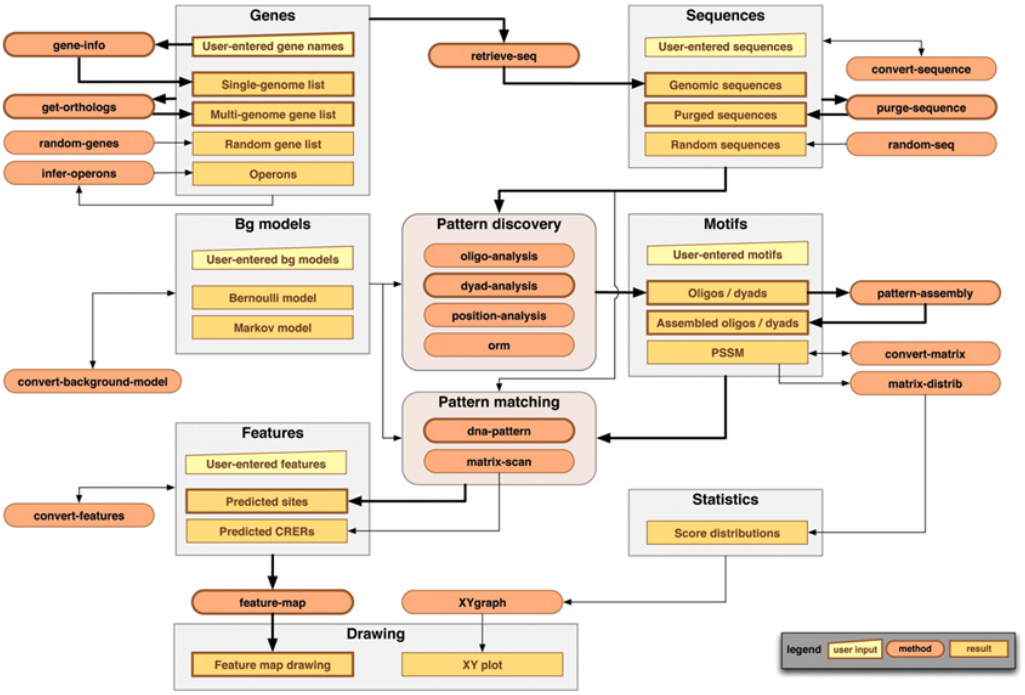

Figure 10: RSAT analysis tools flowchart. Round boxes represent programs; rectangular boxes represent data; trapezoids represent user input [31].

From a computer science perspective the lower level workings of these tools becomes a significant vantage point to stage an efficient model of abstraction. In the context of a new science such as bioinformatics (where sensitivity is to be afforded to biologists), a suitable abstraction away from the programmatic and lower level elements is necessary. For this to be realized all aspects of data, its representation, the tools associated and the linking of these tools, and the collation of all concepts involved are worthy, although sometimes somewhat subjectively defined, areas of thought. A natural starting point for this is the input data for a chosen tool suite. In our case this is the MEME Suite and the input format will be defined below .

4 Input - FASTA format

The FASTA format was first developed for a software package called FASTP [17]. The format starts with a header line starting with a '>' character followed by a unique name (truncated to twenty four characters, if necessary) without spaces. An optional comment/description can be given after a space. This line is then followed by one or more sequence lines (recommended to be under eighty characters in length). Spaces, blanks, and case are ignored in the sequence lines.

MEME web services accept input in multiple other formats (outlined in the web service documentation) and suggests the use of Readseq (found at http://www.ebi.ac.uk/Tools/sfc/readseq/) for conversion of formats. MEME has the ability to read in weight attributes for sequences. The purpose of this is for sequences of great similarity. Weight attribute headers can be anywhere in the file. These headers begin with a '>' followed by an all-caps 'WEIGHTS' and then followed by one or more weightings greater than zero and less than or equal to one. An example FASTA format file followed by one exemplifying the use of the weight attribute is given in figure 11[2]. The accepted alphabets are dependent on which type of analysis is being performed (DNA or protein, for example).

```
>crab_bovin ALPHA
ACTGGGGTCGGCTAGGCTCGAGATATATATTTCGCGATCTCT
CTATAGGGGCTCTAGAGCTCTCGAGAGAGAGAGCTCTCGAG
>crab_anapl BETA
ATTTGCTGATATAGCTCGCTCGATCGCTATATAGGCTCTAGA

>WEIGHTS 0.5 1.0
>WEIGHTS 0.3
>crab_bovin ALPHA
ACTGGGGTCGGCTAGGCTCGAGATATATATTTCGCGATCTCT
CTATAGGGGCTCTAGAGCTCTCGAGAGAGAGAGCTCTCGAG
>crab_anapl BETA
ATTTGCTGATATAGCTCGCTCGATCGCTATATAGGCTCTAGA
```

Figure 11: Examples of the FASTA format.

The purpose to considering this aspect of use of the MEME Suite is to move toward a consistantly more fluent and refined correlation of tools therein.

Other representations

In the case of research being published online, the format in which this is represented is often far from what is directly usable for the reproduction or continuation of the research. Galaxy makes use of its Reproducible Research System for remedying this problem. This still requires the data to be input in a specific format. Online papers, for example, may choose to represent the sequence data or, more commonly, the genomic coordinates in Microsoft Excel or Word documents, pdf's, plain text or tsv files. Each of these has various levels of difficulty associated with the extraction of useful, reliable data.

5 Human Computer Interaction

Human Computer Interaction (HCI) is a wide area of research encompassing sociology, cognitive science, computer science, and many others fields. It concerns itself with a pardaigm of transparency in the interaction between computers and humans. In a less general context this is exemplified in a person's use of an application. The MEME web service has within its arsenal many successful tools for positive research. Auxiliary tools can be developed for a higher level of abstraction affording greater efficiency for users without the need to descend into lower level details.

An area of research into HCI looks at Fitt's law as a model used to define the cost of movement to get to a target area (depending on the distance to and size of the target area). This is used in both ergonomics and HCI. Research in HCI is behind application developement. The desire for humans to use computers results in intuitive and natural solutions to the problem of HCI from an almost purely application development perspective. Consideration of and sensitivity to research done into HCI along with openness to good application solutions centered on users is important in improving functionality and usability within the MEME Suite. This includes well structured and intuitive user interfaces, modularity, and fast response time (considering these tools are designed to perform an assisitive task) [21, 11, 25].

6 Cloud Computing

The National Institute of Standards [23] and Technology defines cloud computing as:

> "...a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Three service models are proposed: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These models cater for the business sector in a highly scalable and efficient manner but are merely proposed solutions.

Since the idea of cloud computing has been made popular misconceptions of what it really is have become commonplace. The main concept underpinning cloud computing is that of provider side hardware and software (whether this is in a private, public or hybrid fashion) with applications and software available as a service rather than an asset.

Consider the case of a prospective business wanting to deploy a possibly lucrative idea on the web. Instead of having to commit resources to big, static hardware, this business can utilize public cloud infrastructure on a 'pay-as-you-go' basis. No upfront commitment is required. This characteristic of cloud computing allows for a balance of costs based on

high traffic and low traffic times - using one server for 1000 hours costs just as much as using 1000 servers for one hour.

With the speedy advance of cloud computing as a service, the question of cloud computing for scientific research becomes a necessary one to ask. It is not uncommon for researchers to require high performace computing for digital experimentation and modelling. The solution of supercomputing and high performance computing is not available to most researchers. The prospect of scientific computing on cloud resources has great potential along with many problems to overcome before this becomes a widely accepted solution.

Ostermann et al. analyse the Amazon EC2 (the most widely used public cloud) for use in the realm of scientific reserach. Due to the nature of specific solutions in scientific computing, the time cost of resource virtualisation for these specific environments becomes an issue.

Cloud computing is already considered a viable solution for smaller scale scientific computing (although it still does not stand up to established means of high performance computing). Specific changes to the cloud infrastructure (parallel storage and high speed networks, for example) have the potential to provide for a new paradigm of cloud computing - High Performance Computing as a Service [26, 16].

This potential scalability and computing power may serve as a valuable resource to run bioinformatics tool suites as the demand for computation for sequence analysis becomes greater and the cost of sequencing decreases (see figure 12). An ideal situation would be to move the MEME Suite to the public cloud space (such as the Amazon EC2 cloud) with only the need for payment credentials to use. Galaxy does offer an expanded use of the Amazon EC2 cloud but requires complex configuration - any attempt to implement this for the MEME Suite would require a sense of transparency and seemless transition.

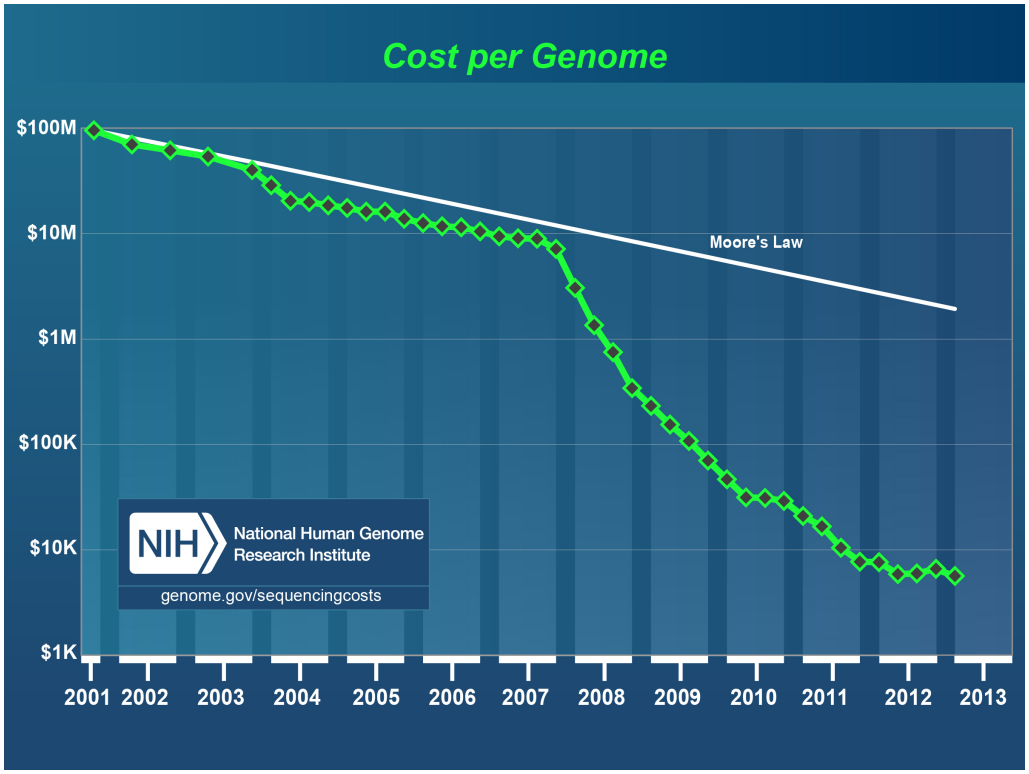

Figure 12: Cost per genome sequencing versus Moore's law [1].

7 Distributed computing

The usability of software is an important consideration to take into account when developing web applications. The processing time associated with web scraping (usually associated with wait time in the associated field of web crawling) and the wait time associated with MEME Suite jobs deserve attention to diminish. From parallel computing to General-Purpose Computation on GPUs [19] to distributed computing, there are architectures to suite the realm of the problem at hand.

Thain et al. are pioneers of the condor project, exploring problems faced by cooperative computing. HTCondor is their system designed as a solution for distributed computing. The focus is on high throughput, data intensive computing with a motto of "leave the owner in control, whatever the cost". This does come at a potentially great cost in terms of reliability.

The condor project faces this reliability issue with a failure consequence approach. If, for whatever reason, a resource cannot be used for computation anymore then the job at

hand is not lost - the progress is kept track of and can continue on a different available system. This exemplifies the desire to make use of any available computing power in a non-invasive, opportunistic manner.

Condor has been around for many years with the kernel remaining unchanged since 1988. This kernel consists of agents, resources and a matchmaker. Agents are responsible for keeping track of jobs and storing them and to find resources to run them. A user of the system can be either or both of these. The matchmaker decides which jobs to assign to which resources. A 'shadow' is responsible at the agent end for deciding what resource is needed for a job. Resources have a sandbox - this is responsible for ensuring a secure execution environment for the execution of the job. See figure 13 for a visual representation of this [30].

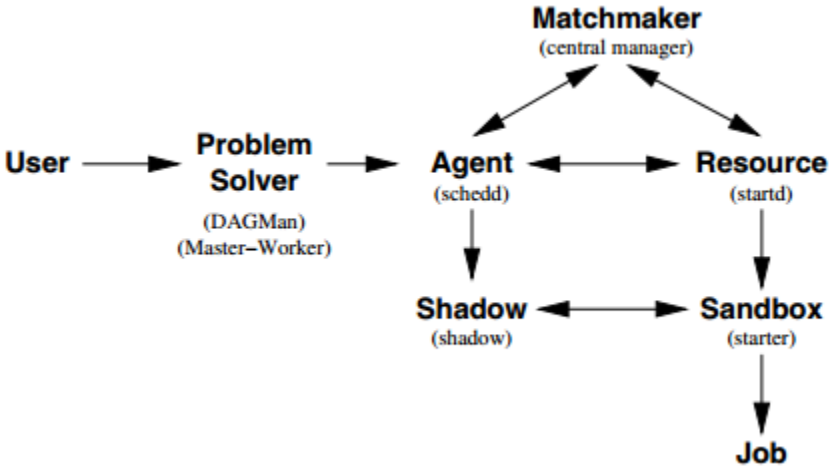

Figure 13: The Condor kernel [30].

This approach to utilising distributive computing for research is novel and well suited for a multitude of individual jobs. Whether this is a good solution for jobs associated with the MEME Suite is up for debate and is to be tested.

8 Web scraping

Web scraping is the act of retrieving information from within a web site or web page and providing a structured output. Web scraping employs a multitude of techniques. Simple techniques employ a copy and paste methodology on the web page or site to be scraped. An extension to this is to make use of regular expressions. These techniques fail extensively with the advance of Web 2.0 and its data driven, dynamic nature and better techniques must be developed.

Web scraping is closely linked with data mining and is not simply an intelligent parse of static web content. Instead, attention must be given to dynamic content such as JavaScript, secure access (HTTPS and sites with required login credentials), and databases dependant on user input. This is often accomplished by use of a wrapper program.

The technique of using a wrapper requires training examples and develops rules for extraction. In this way a wrapper is able to provide a solid solution for information extraction. The WARGO system (proposed and built by Pan et al.) employs a semi-automatic wrapper generation algorithm with specialised parser and extraction languages. The system allows users to provide a navigation tree by simply visually navigating the page intended for scraping. Input such as usernames and passwords can also be provided as the supervised extracting tool develops a XML output [27].

An XML file formed from the HTML of a web page is a good way of 'fixing' broken HTML (since HTML is a very forgiving language) and a necessary layer of abstraction for the web scraping process of gathering information.

A standalone web scraping service needs scalability, modularity, and flexibility. It also needs to employ good error recovery routines since the information being extracted is not designed for this use and tends to be incomplete [24].

Using web scraping techniques to integrate a web scraper in an auxiliary tool for the MEME Suite is a more viable option than building a full web scraping framework. Another option would be to use an open source web scraping application framework such as Scrapy (found at http://scrapy.org/) as a starting point and tweaking the implementation for gathering data from research papers in their non-uniform representations.

These representations range from excel data sheets containing genomic coordinates, pdf documents containing the data as either addendums or as separate entites, actual text representations in one of the many accepted available formats, or even in the paper itself. Often the data desired for reuse is not in a particularly uniform format with crucial information missing (whether the coordinates are 0- or 1- based, for example). In our context, dealing with these complications serve a greater purpose than building a near perfect web scraper.

9 An example

Genomic data is stored in online databases. The UCSC genomic browser is an example of this. Coordinates for browsing sequences in the genome on the UCSC browser are in the following (simplified) format:

chr1:123456-123590

This is a 1- based coordinate system. The BED format is another format used to

represent genomic coordinates. This is very similar to the above example but is rather 0-based. This provides yet another complication to extracting data from online research.

Two pieces of information is necessary in order to gather FASTA sequences from online research papers: which genome the genomic coordinates refer to and the genomic coordinates themselves. Using an external tool such as getFastaFromBed in the BEDTools suite [28], the FASTA sequences can be extracted. See figure 14 below for an example of supplementary data from an online research paper.

G87 f_x

	A	B	C	D	E	F
1	Supplementary Table 1. VDR binding intervals defined following calcitriol stimulation					
2						
3	Chromosome	Start	End	Peak value	Average value	
4	chr1	1700188	1700701	71	49	
5	chr1	1701512	1702022	92	55	
6	chr1	2480734	2481127	168	101	
7	chr1	3583145	3583780	115	67	
8	chr1	3806764	3807410	145	97	
9	chr1	8379671	8380465	99	67	
10	chr1	9408885	9409400	148	81	
11	chr1	9925706	9926330	77	48	
12	chr1	10914805	10915186	96	69	
13	chr1	11890423	11891603	252	129	
14	chr1	11892204	11892807	139	96	

Figure 14: Genomic coordinates in a excel spreadsheet [29].

References

[1] DNA Sequencing Costs. Online, April 2013. Available from: `http://www.genome.gov/sequencingcosts/`.

[2] The MEME Suite: Motif-based sequence analysis tools. Online, May 2013. Available from: `http://meme.nbcr.net/meme/`.

[3] Bailey, T. L. *Discovering motifs in DNA and protein sequences: The approximate common substring problem.* PhD thesis, University of California at San Diego, 1995.

[4] Bailey, T. L. Current Protocols in Bioinformatics: Discovering Novel Sequence Motifs with MEME. Online, 2002. Available from: `http://www.sdsc.edu/~tbailey/MEME-protocol-draft2/protocols.html`.

[5] Bailey, T. L. DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics 27*, 12 (Jun 2011), 1653–1659.

[6] Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., Ren, J., Li, W. W., and Noble, W. S. Meme suite: tools for motif discovery and searching. *Nucleic Acids Research 37*, suppl 2 (2009), W202–W208.

[7] Bailey, T. L., and Elkan, C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol 2* (1994), 28–36.

[8] Bailey, T. L., and Gribskov, M. Combining evidence using p-values: application to sequence homology searches. *Bioinformatics 14*, 1 (1998), 48–54.

[9] Bailey, T. L., Williams, N., Misleh, C., and Li, W. W. MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res 34*, Web Server issue (Jul 2006), W369–W373.

[10] Bayat, A. Science, medicine, and the future: Bioinformatics. *BMJ: British Medical Journal 324*, 7344 (2002), pp. 1018–1022.

[11] Bellamy, R., Bødker, S., Christiansen, E., Engeström, Y., VirginiaEscalante, Holland, D., Kaptelinin, V., Kuutti, K., Nardi, B. A., Raeithel, A., JamesReeves, Velichkovksy, B., and Zinchenko., V. P. *Context and Consciousness: Activity Theory and Human Computer Interaction*. MIT Press, 1996.

[12] Benoït, G. Bioinformatics. ann. rev. info. sci. tech. *Annual Review of Information Science and Technology 39* (2005), 176–218.

[13] Crick, F. H. C., and Watson, J. D. The complementary structure of deoxyribonucleic acid. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 223*, 1152 (1954), 80–96.

[14] Goecks, J., Nekrutenko, A., Taylor, J., and Team, G. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol 11*, 8 (2010), R86.

[15] Grant, C. E., Bailey, T. L., and Noble, W. S. Fimo: scanning for occurrences of a given motif. *Bioinformatics 27*, 7 (2011), 1017–1018.

[16] Juve, G., Deelman, E., Vahi, K., Mehta, G., Berriman, B., Berman, B. P., and Maechling, P. Scientific workflow applications on amazon ec2. In *E-Science Workshops, 2009 5th IEEE International Conference on* (2009), IEEE, pp. 59–66.

[17] Lipman, D., and Pearson, W. Rapid and sensitive protein similarity searches. *Science 227*, 4693 (1985), 1435–1441.

[18] Lowther, J. What is Computer Science? Online. Available from: `http://www.cs.mtu.edu/~john/whatiscs.html`.

[19] Luebke, D., Harris, M., Govindaraju, N., Lefohn, A., Houston, M., Owens, J., Segal, M., Papakipos, M., and Buck, I. Gpgpu: general-purpose computation on graphics hardware. In *Proceedings of the 2006 ACM/IEEE conference on Supercomputing* (New York, NY, USA, 2006), SC '06, ACM.

[20] Machanick, P., and Bailey, T. L. Meme-chip: motif analysis of large dna datasets. *Bioinformatics 27*, 12 (2011), 1696–1697.

[21] MacKenzie, I. S. Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact. 7*, 1 (Mar. 1992), 91–139.

[22] Mandal, A. Online. Available from: `http://www.news-medical.net/health/What-is-DNA.aspx`.

[23] Mell, P., and Grance, T. The nist definition of cloud computing (draft). *NIST special publication 800* (2011), 145. Definition of cloud computing.

[24] Myllymaki, J. Effective web data extraction with standard xml technologies. In *Proceedings of the 10th international conference on World Wide Web* (New York, NY, USA, 2001), WWW '01, ACM, pp. 689–696.

[25] Olson, G. M., and Olson, J. S. Human-computer interaction: Psychological aspects of the human use of computing. *Annual Review of Psychology 54*, 1 (2003), 491–516. PMID: 12209025.

[26] Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., and Epema, D. A performance analysis of ec2 cloud computing services for scientific computing. In *Cloud Computing*. Springer, 2010, pp. 115–131.

[27] Pan, A., Raposo, J., Álvarez, M., Hidalgo, J., and Viña, Á. Semi-automatic wrapper generation for commercial web sources. *Proceedings of the IFIP TC8/WG8 1* (2002), 265–283.

[28] Quinlan, A. R., and Hall, I. M. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics 26*, 6 (2010), 841–842.

[29] Ramagopalan, S. V., Heger, A., Berlanga, A. J., Maugeri, N. J., Lincoln, M. R., Burrell, A., Handunnetthi, L., Handel, A. E., Disanto, G., Orton, S.-M., Watson, C. T., Morahan, J. M., Giovannoni, G., Ponting, C. P., Ebers, G. C., and Knight, J. C. A chip-seq defined genome-wide map of vitamin d receptor binding: Associations with disease and evolution. *Genome Research 20*, 10 (2010), 1352–1360.

[30] Thain, D., Tannenbaum, T., and Livny, M. Distributed computing in practice: the condor experience. *Concurrency and Computation: Practice and Experience 17*, 2-4 (2005), 323–356.

[31] Thomas-Chollier, M., Sand, O., Turatsinze, J.-V., Janky, R., Defrance, M., Vervisch, E., Brohée, S., and van Helden, J. Rsat: regulatory sequence analysis tools. *Nucleic Acids Research 36* (April 2008), W119–W127.

[32] Wing, J. M. Computational Thinking and CS@CMU. Online, 2006. President's Professor and Head Computer Science Department Carnegie Mellon University. Available from: `http://www-cgi.cs.cmu.edu/afs/cs/usr/wing/www/CT_at_CMU.pdf`.