Cartwheel: Tools for Regulatory Genomics

September 18, 2007

1 Abstract

Understanding gene regulation in detail is a goal of both developmental biology and microbial physiology, yet computational tools for investigating the function of non-coding sequence remain relatively immature. Simple, intuitive, and powerful tools that allow individual bench biologists to explore regulatory function in genomic sequence can dramatically increase experimental productivity by both generating and eliminating hypotheses. However, there are very few computational tools that let biologists generate and test hypotheses about regulatory function, even though the opportunities to integrate genomic evidence with experimental design are growing rapidly with the increasing number of sequenced genomes. In particular, there is a growing body of techniques used to predict regulatory DNA and transcription factor/DNA interactions that is essentially unusable by individual biology investigators.

The Cartwheel Project, an existing suite of integrated tools for regulatory genomics, was designed in the Davidson Lab at Caltech in order to find conserved non-coding sequences in localized genomic regions; it is now a standard tool in many labs, because Cartwheel fills the desperate need for easy-to-use genomic analysis tools. However, Cartwheel was designed before many whole genomes were available. We need to enhance and extend Cartwheel to better interact with genome databases and annotations. We also plan to provide accurate and fast multi-genome motif searching and conservation analysis, to facilitate the use of multiple pieces of evidence in hypothesis generation. Finally, we need to continue maintaining the existing tools, and we also need to develop more educational materials on regulatory genomics in order to better support informed research with our tools.

Because transcriptional and post-transcriptional gene regulation is part of almost every biological process, tools to help biologists better study gene regulation will benefit the study of development, physiology, and disease.

2 Specific Aims

Regulatory genomics and regulatory element dissection increasingly depends on *de novo* computational analysis of non-coding DNA sequence. **Cartwheel is currently the only graphical tool built to visualize and cross-compare several different kinds of comparative genomic sequence analyses**. It was developed in direct response to the needs of experimental biologists in the Davidson Lab at Caltech. Developmental biologists need to work with poorly assembled genomic sequence, search sequence for elements conserved between several different species, display and adjust parameters for several different kinds of analyses, and perform exploratory searches for binding sites that occur singly or together. These needs are met by the existing Cartwheel tools. In addition, the Cartwheel Web server lets users maintain a personal sequence collection, analyze sequences with BLAST matches, display the results of gene finding algorithms, and annotate sequences with other features. Users can also run, display, and compare several different kinds of comparative sequence analyses on pairs of sequence. **There is no other platform for regulatory genomics that allows users this range of options.**

Built in 2002, Cartwheel is starting to show its age. The explosion of genomic sequences means that it is now possible to take advantage of conservation within and across multiple genomes to refine hypotheses, and the needs of the biological community for genomic sequence analysis tools have expanded in response. The increase in sophistication of genome annotations, conservation searching tools, and binding site discovery tools means that biologists need better interfaces for interacting with these annotations and search tools. **We propose to enhance and extend Cartwheel to handle these new needs.**

The large increase in transcriptional and post-transcriptional binding site information over the last decade means that we desperately need tools to search for and display predicted binding sites. In particular, there is no easy way to search multiple whole genomes for user-supplied binding sites or compare sequence conservation with binding sites. This approach is critical both for generating and for falsifying regulatory hypotheses. We propose to provide both infrastructure and interfaces for this kind of search.

The flexibility and ease of use of the Cartwheel system also makes it a particularly good choice for biologists with no prior genomics expertise. Cartwheel has a simple Web server interface for establishing analyses and a straightforward and intuitive graphical interface for interacting with analysis results and extracting sequences of interest. Because research programs are often based on the results of computational software, good user interfaces, tutorials, and discussions are needed to help ensure the proper use of computational tools. Moreover, the ability to visualize and compare results from multiple tools is important for cross-validation of computational techniques. We propose to develop interfaces, educational materials, and video demos to help biologists use computational tools effectively and correctly.

As Cartwheel becomes more heavily used, we expect the burden of maintaining the Cartwheel software and servers to increase; currently there is no grant support for this purpose. We plan to continue maintaining the existing Cartwheel installations, enhance installation and usability documentation to enable additional installations, and update our tools and techniques to make maintaining the software easier.

Specific Aim 1: Improve existing Cartwheel functionality and interfaces

Specific Aim 2: Provide a platform for exploring transcription factor binding on genomic scales.

Specific Aim 3: Develop tutorials, video demos, and educational guides for regulatory genomics.

Specific Aim 4: Maintain and support the Cartwheel software and Cartwheel server installations.

3 Background and Significance

3.1 Gene regulation in animal development and microbial physiology

Gene regulation is central to the study of biology. From microbial physiology to animal development, spatial and temporal control of gene expression in response to the cell cycle, intra- and inter-cellular signalling, and environmental information is a universal part of the processes of growth, differentiation, and adaptation [4, 20].

The underlying mechanisms of gene regulation are known in outline. Transcriptional regulation is controlled by the recruitment and activation of polymerase to specific gene start sites, usually regulated by sequence-specific DNA binding proteins. In eukaryotes, RNA splicing and export are controlled by polyprotein/RNA complexes that recognize sequence specific features on nuclear RNA and direct normal and alternative splicing. Finally, there are a variety of common post-export regulatory steps, including message sequestration, preferential stabilization and degradation, and translation inhibition, that are performed both by miRNA and RNA-binding proteins.

Despite this general knowledge, we are still in the process of understanding the detailed regulation of specific genes. While the broad outlines of several developmental gene networks are clarifying, the number of developmental genes for which all upstream regulatory factors are known can be counted on the fingers of one hand. The details of physiological gene regulation in microbes are poorly understood, even in *E. coli*. Tissue-specific splice enhancers and alternative splicing control mechanisms in animal and plant development are a burgeoning field of study, but both the scope and details of regulated splicing are only now emerging. And our view of the post-transcriptional control of messages is growing more complicated every day.

Currently it is thought that somewhere between 10% and 30% of all genes in vertebrate genomes are regulated post-transcriptionally. From gene expression studies one can find many cases where there is inconsistency between the levels of RNA and protein expression derived from the same gene. This difference could be due to transcript stability, transcript processing, sub-cellular localization of the message, or rate of translation. The regulatory mechanisms include RNA binding proteins and regulatory RNAs [12].

We cannot say that we understand the role of a gene in development or physiology until we understand how and why that gene's expression is controlled.

Our study of such processes profits dramatically from the advent of whole-genome sequencing. We can now devise assays that query targets of DNA- and RNA-binding proteins on the scale of an entire genome or a whole transcriptome, and we can link the results of those assays back to genomic sequence. We can search within and across genomes for binding sites and conservation patterns and finally understand, on a global genomic scale, the downstream targets of regulatory proteins.

However, the availability of entire genomes and entire transcriptomes has also led to an increasing divide between the computationally able and the biologically able, who rarely overlap in a single person. It is clear that we will not understand gene regulation in detail without synthesizing results from both computational genomics and experimental biology. This is a major goal of the ENCODE consortium [8].

3.2 Computer-assisted investigation of regulatory information

While gene regulation is of central importance in both physiology and development, it can be difficult to study. Transcriptional and post-transcriptional regulatory elements are difficult to find and have no obvious de novo statistical signature. Transcriptional regulatory regions, or *cis*-regulatory elements, can lie within tens or hundreds of kilobases of the genes they regulate, and there are no experimental techniques capable of finding them on a whole-genome scale; transcription factor binding sites within regulatory regions can be investigated using chromatin IP, which is dependent on good antibodies, but also has an unknown false positive rate; and in vitro assays for both transcriptional regulatory function and protein/DNA binding can

be misleading. Post-transcriptional regulatory elements should in theory be simpler to investigate because they involve less overall sequence – UTRs are much shorter than non-coding regions – but very little work has been done in this area until recently [19].

Over the last decade, a number of computational approaches have been successfully applied to the study of transcriptional regulation. As more genomic sequence becomes available, comparative sequence analysis has been increasingly effective in helping to find and dissect *cis*-regulatory elements. Whole-genome binding site searches have become been effectively utilized in microbes and some of the smaller animal genomes, although search of larger genomes still yields unacceptable false positive rates. A number of large-scale motif discovery algorithms have also been developed to help extract binding site predictions from gene sets. Finally, several high-throughput experimental assays like ChIP-chip, ChIP-seq, and RIB-chip produce data that cannot be understood without recourse to computational tools.

The problem of understanding gene regulation can be divided into three areas: finding *cis*-regulatory modules, dissecting them into binding sites, and correlating whole-genome predictions with data from large-scale experimental assays.

3.2.1 Finding *cis*-regulatory modules

Two computational approaches have been very successful in locating *cis*-regulatory modules in animal genomes. Comparative sequence analysis of non-coding sequence is essentially the *de facto* method for predicting the location of *cis*-regulatory modules in most animal genomes, while binding site cluster analysis has been very effective in *Drosophila* and *C. elegans*.

Comparative sequence analysis relies on the signature of selection left by functional regions in non-coding DNA. Sequences near to the gene of interest in multiple organisms are compared and those that are alike above a certain threshold of similarity are singled out as conserved. These conserved elements are then tested for function in vitro or in vitro, usually by being cloned into a reporter construct and introduced into an amenable organism. Comparative sequence analysis has worked quite well in sea urchins, nematodes, chordates, and many vertebrate genomes [3, 14].

Binding site cluster analysis relies on the observation that regulatory elements tend to contain multiple binding sites for one or more transcription factors. Using a simple motif search, the whole genome is scanned for binding sites, and clusters are picked out from the genome and tested. This approach has been effective in the *Drosophila* genome (180 Mb) and the *C. elegans* genome (100 Mb) [1, 11]. The primary drawback of this approach is that requires a priori knowledge of transcription factor binding sites, as well as some idea of what transcription factors work together.

3.2.2 Finding and predicting binding sites

Finding and predicting binding sites is distinct from locating *cis*-regulatory elements, because binding sites almost always work in tandem with other binding sites.

Typically, finding binding sites is a matter of having some a priori knowledge about the sequence-specific binding preferences of the transcriptional regulators of interest. The binding preferences can be represented in a number of ways: the two most common are IUPAC motifs, e.g. "WGATAR", that reflect motif degeneracy by inclusion of all known binding sequences, and position-weight matrices (PWMs) that represent motif degeneracy and/or binding strength by weighting the contribution of individual nucleotides in specific positions. These techniques have been extremely effective in microbes, although they have a fairly high false positive rate; this high false positive rate makes it difficult to rely on binding site search in larger genomes, which have substantially more non-coding sequence. While several backgrounding algorithms have been developed to reduce false positive rates, there seem to be a substantial number of binding sites that reside in a "twilight zone", close to the discovery background [10].

The main problem with binding site search, then, comes down to high false positive rates. Two approaches have been developed for reducing the false positive rate. The first approach uses the biological fact that binding sites tend to cluster in order to reduce the background prediction rate, and has been effective at finding functional regulatory modules (see above). The second approach analyzes conservation of binding sites, either in isolation or in tandem with other sites, as an independent source of evidence for function; in addition to the strict sequence conservation approach used to find regulatory modules, binding site location near to orthologous genes can be considered positive evidence for function.

Binding site discovery is an even more challenging problem than binding site search. Most binding site discovery algorithms scan putatively co-regulated sequences for common subsequences, from which they build position-weight matrices or other discriminators of binding. This has worked very well in a number of cases, including in analysis of microarray data, ChIP-chip data, and ChIP-seq data. Nonetheless, binding site discovery seems to best be done in conjunction with a limiting set of biological sequences rather than on a whole-genome scale, and a survey of many binding site discovery algorithms demonstrates that they each have different specificities and sensitivities [21]. This suggests that the best use of binding site discovery programs lies in running several of them and comparing the results.

3.2.3 Large-scale experimental assays of protein/nucleotide binding

The increasing use of large-scale experimental assays has led to an interesting problem: because assays like ChIP-chip, ChIP-seq, and RiP-chip assays produce so much data, the results cannot be understood without recourse to computation. Moreover, these large scale assays yield both false positives and false negatives, so the computation necessary to understand these results is not trivial. Several recent publications have devoted significant computational resources to these problems [2, 9, 13, 15]. Computational frameworks supporting the detailed mapping and querying of large-scale regulatory assays have yet to emerge, although it is clear that they will be needed in the future.

3.2.4 Post-transcriptional regulation

Eukaryotic cells frequently employ post-transcriptional mechanisms that involve RNA-binding proteins (RBP) to coordinate the expression of their genes. RBPs regulate RNA splicing, transport, translation and stability, and in doing so, form networks that orchestrates the fate of the transcriptome. Investigating post-transcriptional regulation in perspective can be aided by computational approaches. Both *in silico* and *in vivo* methods are necessary to map regulatory elements, to determine the mRNAs that are regulated by a given RBP and to establish a logical relationship amongst their encoded protein products.

3.2.5 Effectiveness of computational approaches to regulatory genomics

It is difficult to overstate the usefulness of computational approaches to experimental biology, or the need for both easy-to-use and more powerful tools in regulatory genomics. We have three personal anecdotes to offer: First, comparative sequence analysis, as used in the Davidson Lab and elsewhere, has changed the search for transcriptional regulatory regions from a 2 year search by an experienced researcher into a 3 month summer student project, significantly aiding the creation of one of the best developmental regulatory network models in existence [22, 5]. Second, binding site analysis in microbes has guided experiments in *S. oneidensis* such that we now have an understanding of the differences between aerobic and anaerobic metabolism between *S. oneidensis* and *E. coli*; such a survey would have been unthinkable without computational support [7]. And, third, the combined use of computational tools and ChIP-seq combined has led to an unprecedented whole-genome understanding of the binding of NRSF in the mouse genome [9, 15].

3.3 Building functional, usable software for biologists as end user

The large number of sequenced genomes presents both new problems and new opportunities for biologists. As we discuss above, it is now possible to investigate the function of the huge swathes of noncoding sequence present in most genomes, and computational tools are critical for this. Most biologists, however, have little computational expertise, and so must rely on others' tools.

There is a rich tradition of tool development in bioinformatics, computational biology, and regulatory genomics. Every issue of Genome Research, Bioinformatics, and other journals contains papers about tools that are relevant to gene regulation, genome-wide searches, and genome-scale annotation. The tools are not lacking.

Many computational tools only work through the command line, however, and those that do not are usually accessible only via an isolated Web site that makes it difficult to cross-compare, correlate, and build on results from other programs. This means that biologists wishing to use these tools must become expert in largely irrelevant activities like UNIX command line program execution and data parsing/transformation.

The solution to these problems is clear: we need to build usable graphical interfaces that let biologists run many different kinds of analyses, build integrated views of analysis results and sequence data, and import and export data in rich formats. So why don't more programs provide such functionality?

The central reason, we believe, is that writing graphical interfaces and easy-to-use software is boring, difficult, expensive, and time-consuming. It is also difficult to fit into a traditional research program, both because funding opportunities for such efforts are lacking and because biology graduate students are encouraged to do experimental biology, while computer science and other theoretical graduate students generally develop new approaches rather than making old approaches easier to use. While one might expect industry to step in, the biotech industry has been relatively ineffective in developing good interfaces to research tools, perhaps in part because of licensing issues (see below).

Genome and sequence databases are a bright spot in this dim assessment. Databases such as NCBI, ENSEMBL, UCSC, FlyBase, and WormBase have developed easy-to-use and information rich graphical interfaces to genome-scale datasets out of necessity. However, such sites have strong mission statements, serve a very large user base, and have finite resources; thus, they cannot generally integrate research software that does not fit their mission statement. Moreover, the computational resources of these databases are focused on providing views and searches of static annotations, so they may not be interested in or capable of running computationally expensive tools on their datasets.

3.3.1 Building interfaces as part of a research program in regulatory genomics

The ultimate goal of many types of computational predictions – and certainly most computational predictions in regulatory genomics – is to be validated *in vivo* or *in vitro*. If software is difficult to use, however, experimental biologists are loathe to devote the time to learn the software in order to apply it to their own problems. Providing easy-to-use interfaces and data output formats is a productive way to get biologists to test our computational predictions. These tests in turn provide immediate feedback on the specificity and sensitivity of computational predictions.

Frameworks that let computational researchers provide useful interfaces to their tools to biologists are therefore invaluable. There are several such frameworks being developed, such as Taverna and Galaxy, but they are still young and tend to be very broadly focused [6, 16]. In contrast, tools that are tightly focused on one specific area of investigation can provide better integration of tools and more relevant displays, at the expense of generality.

3.3.2 Licensing and software availability issues

Software licensing is a big problem standing in the way of integrating multiple computational tools. There are two general kinds of licensing issues.

One issue is the license for using and modifying packages. While there are no systematic studies of software licensing in bioinformatics, many useful packages are freely available only to academic researchers, and most are not available in source form. If software can only be used in academia, then anyone making use of that software in an integrated framework must similarly restrict their functionality, thereby adding an additional significant burden for framework developers. Software that is not available in source form is difficult to use as part of a robust framework, because it is almost always tightly bound to a specific platform by compiled-in version dependencies.

The other issue is the license for redistributing packages. Only packages adhering to an OSI-certified license (www.opensource.org) can be modified and redistributed with those modifications, or integrated tightly into a larger framework. Even when the source code to a package is available, the package may not be open source. This burdens framework developers because they must maintain separate ”forks” of package source trees. Another problem is that even some open source licenses are incompatible: for example, source code under ”copyleft” licenses like the GNU Public License cannot necessarily be combined with source code from ”copy-free” licenses like the BSD license. These are issues to which there are no good solutions yet, although they are all issues being tackled by the open source community.

One possible solution is to make software available under dual-license, and to make sure that as much software as is possible is released under open source licenses. Another solution is to rely on a small set of core packages and make arrangements for re-licensing of these packages for framework use, with citation and co-authorship credit.

3.3.3 Software reliability

A problem faced by anyone who would put long-term effort into developing research software is the need to build maintainable and reliable software. As anyone who has used a computer is aware, software is often unreliable. Research software is no exception, and in fact may be more susceptible to errors and other unreliability than other software, because it is usually written to solve problems with no obviously correct answer, so errors can go undetected indefinitely. In addition, research software often has relatively few users and is written by inexperienced graduate students. However, research programs are often based on the results of such software, leading to the potential for serious problems; for example, recently a number of protein structure predictions have been overturned because of a single sign error in structure prediction software [17]. This is a problem by no means limited to biology: Press et al. [18] estimates that a substantial number of physics and chemistry papers are incorrect because of improper use of random number generators.

However, the growing dependence of biology research on bioinformatics software implies that this is a growing problem for biology. Gene predictions and data integration pipelines dominate our biological research in genomics; protein homology predictions generate hypotheses throughout the study of physiology, development, and disease; literature search and citation systems guide our reading; and binding site and conservation analyses influence our experimental design. Because of this increasing dependence on software, errors in software development can hugely influence our science.

While there are no easy solutions here, one potential solution to these problems is to provide interfaces and data comparison software that can cross-compare results from multiple programs run with multiple parameters. Another solution is to adhere to good software design and development principles. A third solution is to develop regression test frameworks capable of determining whether or not computational predictions have changed erroneously during software development. And, finally, the ultimate solution must be to develop research software that accurately and reliably makes predictions of verifiable biological significance.

3.3.4 Software and user education in genomics

Another problem with bioinformatics software development and genomics in general is the need to educate users with respect to the computational techniques – the algorithms, data integration, and visualization techniques being used. While properly a component of research, the increasing reliance on ”black box” computational predictions in biology means that users are often unaware of the prior assumptions upon which the software depends, the full content and biological source of data sets, and the meaning of the output restrictions of computational programs. BLAST is an excellent case in point: how many people really understand BLAST? In our personal experience, people use BLAST in a variety of inappropriate ways, including searching morpholino and primer sequences against whole genomes, searching incomplete genome assemblies as a way to verify absence of genes in particular genomes, and using too-stringent or too-lax expectation thresholds in homology searches.

This is a problem that requires multiple overlapping solutions, but good user interfaces, good tutorials, and good discussion of algorithms are all critically important to informing biological users of what tools can and cannot do.

3.3.5 Computational support for hypothesis driven regulatory genomics

Support for hypothesis driven investigation of regulatory genomics is still in its infancy. Bioinformaticians tend to focus on whole-genome questions, while many experimental biologists are more often interested in the regulation of a small set of relevant genes. This leads to an uncomfortable accommodation: biologists often like to adjust search parameters over a wide range, perhaps allowing for a higher false positive rate in the hopes of feature identification close to a gene of interest, or looking for fewer false positives or false negatives. Computational scientists tend to focus on optimizing specificity and sensitivity against ”known good” data sets, which can lead to more false negatives or more false positives. We believe that user interfaces that support parameter adjustment and fast searching enable the kind of iterative analyses that experimentalists need in order to investigate the parameters relevant to the genes they are interested in. This moreover may have the effect of increasing the willingness of an experimentalist to trust the computational results and test them biologically.

References

[1] HR Bigelow, AS Wenick, A Wong, and O Hobert. Cisortho: a program pipeline for genome-wide identification of transcription factor target genes using phylogenetic footprinting. *BMC Bioinformatics*, 5:27, 2004.

[2] AR Borneman, ZD Zhang, J Rozowsky, MR Seringhaus, M Gerstein, and M Snyder. Transcription factor binding site identification in yeast: a comparison of high-density oligonucleotide and pcr-based microarray platforms. *Funct Integr Genomics*, 7(4):335–45, 2007.

[3] CT Brown, Y Xie, EH Davidson, and RA Cameron. Paircomp, familyrelationsii and cartwheel: tools for interspecific sequence comparison. *BMC Bioinformatics*, 6:70, 2005.

[4] EH Davidson. *The Regulatory Genome*. Academic Press, 2006.

[5] EH Davidson, JP Rast, P Oliveri, A Ransick, C Calestani, CH Yuh, T Minokawa, G Amore, V Hinman, C Arenas-Mena, O Otim, CT Brown, CB Livi, PY Lee, R Revilla, AG Rust, Z Pan, MJ Schilstra, PJ Clarke, MI Arnone, L Rowen, RA Cameron, DR McClay, L Hood, and H Bolouri. A genomic regulatory network for development. *Science*, 295(5560):1669–78, 2002.

[6] B Giardine, C Riemer, RC Hardison, R Burhans, L Elnitski, P Shah, Y Zhang, D Blankenberg, I Albert, J Taylor, W Miller, WJ Kent, and A Nekrutenko. Galaxy: a platform for interactive large-scale genome analysis. *Genome Res*, 15(10):1451–5, 2005.

[7] JA Gralnick, CT Brown, and DK Newman. Anaerobic regulation by an atypical arc system in shewanella oneidensis. *Mol Microbiol*, 56(5):1347–57, 2005.

[8] R Guigo, P Flicek, JF Abril, A Reymond, J Lagarde, F Denoeud, S Antonarakis, M Ashburner, VB Bajic, E Birney, R Castelo, E Eyras, C Ucla, TR Gingeras, J Harrow, T Hubbard, SE Lewis, and MG Reese. Egasp: the human encode genome annotation assessment project. *Genome Biol*, 7 Suppl 1:S2.1–31, 2006.

[9] DS Johnson, A Mortazavi, RM Myers, and B Wold. Genome-wide mapping of in vivo protein-dna interactions. *Science*, 316(5830):1497–502, 2007.

[10] U Keich and PA Pevzner. Subtle motifs: defining the limits of motif finding algorithms. *Bioinformatics*, 18(10):1382–90, 2002.

[11] M Markstein, P Markstein, V Markstein, and MS Levine. Genome-wide analysis of clustered dorsal binding sites identifies putative target genes in the drosophila embryo. *Proc Natl Acad Sci U S A*, 99(2):763–8, 2002.

[12] J Mata, S Marguerat, and J Bahler. Post-transcriptional control of gene expression: a genome-wide perspective. *Trends Biochem Sci*, 30(9):506–14, 2005.

[13] RP McCord, MF Berger, AA Philippakis, and ML Bulyk. Inferring condition-specific transcription factor function from dna binding and gene expression data. *Mol Syst Biol*, 3:100, 2007.

[14] W Miller, KD Makova, A Nekrutenko, and RC Hardison. Comparative genomics. *Annu Rev Genomics Hum Genet*, 5:15–56, 2004.

[15] A Mortazavi, EC Leeper Thompson, ST Garcia, RM Myers, and B Wold. Comparative genomics modeling of the nrsf/rest repressor network: from single conserved sites to genome-wide repertoire. *Genome Res*, 16(10):1208–21, 2006.

[16] T Oinn, M Addis, J Ferris, D Marvin, M Senger, M Greenwood, T Carver, K Glover, MR Pocock, A Wipat, and P Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–54, 2004.

[17] GA Petsko. And the second shall be first. *Genome Biol*, 8(2):103, 2007.

[18] WH Press, SA Teukolsky, WT Vetterling, and BP Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[19] P Sanchez-Diaz and LO Penalva. Post-transcription meets post-genomic: the saga of rna binding proteins in a new era. *RNA Biol*, 3(3):101–9, 2006.

[20] AS Seshasayee, P Bertone, GM Fraser, and NM Luscombe. Transcriptional regulatory networks in bacteria: from input signals to output responses. *Curr Opin Microbiol*, 9(5):511–9, 2006.

[21] M Tompa, N Li, TL Bailey, GM Church, B De Moor, E Eskin, AV Favorov, MC Frith, Y Fu, WJ Kent, VJ Makeev, AA Mironov, WS Noble, G Pavesi, G Pesole, M Regnier, N Simonis, S Sinha, G Thijs, J van Helden, M Vandenbogaert, Z Weng, C Workman, C Ye, and Z Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol*, 23(1):137–44, 2005.

[22] CH Yuh, CT Brown, CB Livi, L Rowen, PJ Clarke, and EH Davidson. Patchy interspecific sequence similarities efficiently identify positive cis-regulatory elements in the sea urchin. *Dev Biol*, 246(1):148–61, 2002.