Nucleic Acids Research, 2004, Vol. 32, Web Server issue ***W235–W241***
DOI: 10.1093/nar/gkh483

SeqVISTA: a new module of integrated computational tools for studying transcriptional regulation

Zhenjun Hu[1], Yutao Fu[1], Anason S. Halees[1], Szymon M. Kielbasa[3] and Zhiping Weng[1,2,*]

[1]Bioinformatics Program and [2]Department of Biomedical Engineering, Boston University, 44 Cummington Street, Boston, MA 02215 and [3]Institute for Theoretical Biology, Humboldt University Berlin, Invalidenstraße 43, D-10115 Berlin, Germany

Received February 16, 2004; Revised April 21, 2004; Accepted May 4, 2004

ABSTRACT

Transcriptional regulation is one of the most basic regulatory mechanisms in the cell. The accumulation of multiple metazoan genome sequences and the advent of high-throughput experimental techniques have motivated the development of a large number of bioinformatics methods for the detection of regulatory motifs. The regulatory process is extremely complex and individual computational algorithms typically have very limited success in genome-scale studies. Here, we argue the importance of integrating multiple computational algorithms and present an infrastructure that integrates eight web services covering key areas of transcriptional regulation. We have adopted the client-side integration technology and built a consistent input and output environment with a versatile visualization tool named SeqVISTA. The infrastructure will allow for easy integration of gene regulation analysis software that is scattered over the Internet. It will also enable bench biologists to perform an arsenal of analysis using cutting-edge methods in a familiar environment and bioinformatics researchers to focus on developing new algorithms without the need to invest substantial effort on complex pre- or post-processors. SeqVISTA is freely available to academic users and can be launched online at http://zlab.bu.edu/SeqVISTA/web.jnlp, provided that Java Web Start has been installed. In addition, a stand-alone version of the program can be downloaded and run locally. It can be obtained at http://zlab.bu.edu/SeqVISTA.

INTRODUCTION

Transcriptional regulation is an extremely important mechanism in controlling the spatial and temporal production of mRNA molecules, which impacts the subsequent production of protein molecules. Thus, from a finite number of genes, an almost infinite variety of protein forms (produced from alternative splicing and post-translational modifications) can be created in a highly regulated fashion in response to physiological and environmental stimuli. Transcriptional regulation is highly complex, especially in multicellular eukaryotic organisms such as humans. Despite the extraordinary strides that have been made in genome sequencing and computational algorithm development, our ability to recognize functional regulatory elements in multicellular eukaryotic genomic sequences remains limited. Currently available computational algorithms, if applied individually, are unable to reliably detect *cis*-elements that are functional *in vivo*. This is largely due to the complexity of eukaryotic genomes. For example, the chromatin around a site may not be open for the DNA to bind to transcription factors (TFs), and even if a site were accessible, the gene may be expressed in a cell type in which the TF is not transcribed. Furthermore, the prediction of the proximal promoter may be in error because of the alternative usage of another first exon (and thus the associated proximal promoter) at a distant chromosomal location. The computational methods will obviously continue to improve as more experimental data become available. However, we argue that biological systems are so complex that we are extremely far away from producing a generic computational model that is sophisticated enough to capture the regulatory mechanisms of all genes.

Therefore, in order for current computational methods to significantly impact our biological understanding, we must (i) integrate all methods and databases so that the user can take advantage of the strengths of different methods as well as annotations describing previous experimental results on the same gene, and (ii) make the system extremely user-friendly to bench biologists, so that they can incorporate their own expert knowledge and experimental results and perform computation–experiment iterations to maximize the impact on their results.

There are a large number of sequence databases of regulatory regions and programs for analyzing these sequence

*To whom correspondence should be addressed. Tel: +1 617 353 3509; Fax: +1 617 353 6766; Email: zhiping@bu.edu

regions. A long list is available at our lab website: http://zlab.bu.edu/zlab/gene.shtml. Some programs have a web interface and others can be downloaded to run locally. Many of the online resources provide remarkable user interfaces and rich connections between diverse types of data. However, as with all rapidly developing fields, there has been little cooperation among bioinformatics development efforts. Each author exerts maximal creativity in order to achieve the best user experience when it comes to designing both data organization and user interface. Although individually highly functional, the lack of interoperability across multiple tools poses a severe inconvenience for an experimental biologist user in several aspects: (i) since most sensible biological questions require the integration of various types of data, the user frequently needs to use multiple tools to retrieve and analyze these data; (ii) as programs with similar goals can have distinctly different user interfaces, the user has to spend substantial time adjusting to each program; (iii) there is no easy way to exchange data between the programs and the user often needs to submit the same input to multiple programs one by one and attempt to piece together and compare the output of these programs, which are formatted rather differently. The lack of interoperability is disastrous for bioinformaticians, who must develop computer programs to communicate in multiple formats. Numerous parsers need to be implemented to load data with different formats or data from different sources with distinct interfaces. If an interface or data format changes, the parser will need to be modified to accommodate the change. This situation is described in L. Stein's excellent reviews (1,2).

To date, we are aware of three programs that are working toward the goal of integration. However, there is still ample room for further development. Regulatory Sequence Analysis Tools (RSAT) is a compilation of multiple tools in one website (3). It has extensive documentation, and is easy to use. However, the user still needs to paste the input sequence into each tool and perform the analysis individually. Furthermore, the resource has been designed for analyzing prokaryote and yeast sequences. Toucan is a stand-alone program with functionality for phylogenetic footprinting, overrepresented motif search and *ab initio* motif discovery (4). However, it has a rather limited visualization front-end. INCLUSive is a web compilation of microarray analysis and *cis*-element motif analysis tools (5), using Toucan as the visualization module. Similar to RSAT, tools in INCLUSive are only loosely coupled and the user must run them individually.

In this paper, we present an integration effort that includes eight web services (described in the next section), which cover most key approaches for finding regulatory motifs in higher eukaryotes. We have augmented a versatile sequence and annotation visualization program SeqVISTA (6) with a new Motif module to facilitate the integration. SeqVISTA supplies a consistent input and output interface as well as the ability to recognize rich annotations in input sequences. Its new Motif module contains functions to directly query the web services and retrieve results. The module can automatically recognize the logic of the sequential application of multiple programs and the relationships among their results and provides a coherent and versatile visualization of the output files. It is technically straightforward to integrate additional web services, which makes SeqVISTA a general integration infrastructure for biological sequence analysis.

INCLUDED WEB SERVICES

Table 1 lists the addresses for the eight web services that are now integrated in SeqVISTA. Below we briefly describe the method of each of them as follows. Among the five tools that have been developed in our lab, Glam and Clover were downloadable, while the others were downloadable also supported web-based user interfaces. None of them could communicate via the Simple Object Access Protocol (SOAP; http://www.w3.org/TR/2003/REC-soap12-part0-20030624/), which is the current approach for stable integration across multiple software platforms. We have developed SOAP-based web services for these five programs.

(i) *PromoSer:* PromoSer is a web service that facilitates extraction of user-specified regions around the transcription start sites of all annotated human, mouse and rat genes. The transcription start sites (TSSs) have been identified computationally by considering alignments of a large number of partial and full-length

Table 1. Included web services

Program	Web interface	SOAP WSDL	References
Retrieve Promoter Sequences			
PromoSer	http://biowulf.bu.edu/zlab/PromoSer/	http://biowulf.bu.edu/zlab/promoser/promoser.wsdl	(7)
Repeat masking			
Repeat masker	http://repeatmasker.genome.washington.edu/cgi-bin/RepeatMasker	Not available	A.F.A. Smit and P. Green, personal communication
Simple matrix-based motif detection			
Possum	http://zlab.bu.edu/~mfrith/possum/	http://biowulf.bu.edu/clover/possumwsdl	
Screening a motif library for detecting over-represented motifs			
Clover	http://zlab.bu.edu/clover/	http://biowulf.bu.edu/clover/wsdl	(8)
MotifScanner	http://www.esat.kuleuven.ac.be/~thijs/Work/MotifScanner.html	http://www.esat.kuleuven.ac.be/inclusive/MotifScanner.wsdl	(4,5)
Ab initio motif detection with Gibbs sampling			
Glam	http://zlab.bu.edu/glam/	http://biowulf.bu.edu/clover/glamwsdl	(13)
MotifSampler	http://www.esat.kuleuven.ac.be/~thijs/Work/MotifSampler.html	http://www.esat.kuleuven.ac.be/inclusive/MotifSampler.wsdl	(12,15,16)
Detection of motif clusters			
Cluster-Buster	http://zlab.bu.edu/cluster-buster/	http://biowulf.bu.edu/clover/cbwsdl	(30)

mRNA sequences to genomic DNA while allowing for alternative promoters. The service has been described previously (7). PromoSer relies on a periodically updated, pre-computed database to quickly and efficiently process batch queries for promoters. The input to PromoSer is a set of transcribed sequences (mRNA or EST) or GenBank identifiers of such sequences and the output is the corresponding promoter sequences and a table of TSS locations. Full-length cDNA sequences are not compared online but are incorporated into the database during updates. Sequence based queries are mapped to the genome to find associated clusters of transcripts and their associated TSSs and promoters.

(ii) *RepeatMasker:* RepeatMasker screens DNA sequences in FASTA format against a library of repetitive elements and returns a masked query sequence ready for database searches as well as a table annotating the masked regions (A. F. A. Smit and P. Green, personal communication). The repetitive nature of such regions may lead to artificially strong motif hits, but in some cases they may contain genuine regulatory elements.

(iii) *Possum:* The most basic way of detecting regulatory motifs is to scan the position-specific scoring matrix representing the motif along the sequence of interest and report all the matches that score above a user specified threshold. Possum does precisely this.

(iv) *Clover:* One way of detecting *cis*-elements in the promoters of co-regulated genes is to determine whether the promoters are enriched with any known motifs. We have developed a computer program called Clover that screens a set of DNA sequences against a library of position-specific scoring matrices and reports any motifs that are overrepresented in the sequence set. We further establish the statistical significance of the results by random shuffling of input sequences and matrices and/or applying the program to background sequences (8).

(v) *MotifScanner:* MotifScanner is a program that can be used to screen DNA sequences with precompiled motif models (4,5). It has a similar goal to Clover as described above (8). MotifScanner is based on a probabilistic sequence model in which motifs are assumed to be hidden in a noisy background sequence modeled with higher-order Markov models.

(vi) *Glam:* Many methods have been developed to detect motifs that are overrepresented in the regulatory regions of co-regulated genes. Among them, Gibbs sampling has several advantages, among which it is rooted in an elegant Markov Chain Monte Carlo formulation and has the ability to jump out of local optima. Several groups have developed Gibbs sampling based methods for detecting transcription factor binding sites (9–12). We have developed an enhanced Gibbs sampling algorithm Glam that addresses two limitations of previous methods (13): it incorporates automatic optimization of the alignment width and it estimates statistical significance using BLAST-like formulas (14).

(vii) *MotifSampler:* MotifSampler is a Gibbs sampling based motif finding algorithm (12,15,16). It has a similar goal to Glam as described above (13). The strength of MotifSampler is the use of higher-order background models.

(viii) *Cluster-Buster:* A number of methods have been proposed to detect clusters of sites in an input sequence that are preferentially bound by several transcription factors (17–29). We have taken a modeling approach, searching for regions of the sequence that resemble a statistical model of a motif cluster more than they resemble a model of 'background DNA'. We have developed three methods to tackle this problem: Cister (24), Comet (25) and Cluster-Buster (30). Cluster-Buster is our newest software; it tackles the problem head-on, employing a linear-time heuristic that attempts to return the same cluster predictions as the full quadratic-time algorithm (30).

In order to help the user learn these web services, as well as the batch function described below, we have prepared several tutorials. These tutorials can be accessed at http://zlab.bu.edu/SeqVISTA/tutorials/. Phylogenetic footprinting is a commonly used technique to discover functionally conserved sequence regions across species, which may have a higher chance of harboring regulatory elements. We plan to incorporate a phylogenetic footprinting service into SeqVISTA in the near future. Presently, the user can upload a sequence with non-conserved regions in lower-case letters and they will be skipped over in the motif-finding services in SeqVISTA.

INTEGRATION VIA SeqVISTA

SeqVISTA: a user-friendly sequence viewer

Previously, we developed a Java-based sequence visualization program named SeqVISTA (6). It presents a holistic, interactive graphical view of sequence records with supporting annotation data. It has most functions for sequence manipulation, such as load, copy, paste, locate and pattern search. We have tried to make SeqVISTA as user-friendly as possible. Specifically, we have developed several functions in the sequence panel of SeqVISTA to render sub-sequence selection effortless. In addition, the user can launch the program while browsing a sequence record using Internet Explorer by clicking the 'SeqVISTA' button, which is added to the browser during the installation of SeqVISTA. SeqVISTA can communicate with external analysis programs and displays their outputs along with the GenBank annotations of the sequence in an integrated fashion. SeqVISTA runs on all computer platforms that support Java.

Method of integration

Our goal is to develop the essential features of an integrated infrastructure for computational studies of gene regulation. The infrastructure contains three components: (i) *Preprocessing*: focused on data collection and formatting; (ii) *Computational Core*: focused on data analysis, e.g. repeat masking and motif discovery; (iii) *Post-Processing*: focused on output visualization and integration. SeqVISTA covers the basic components of pre- and post-processing and the web services described in the previous section constitute the computational core. Since the original publication of SeqVISTA (6), we have significantly improved it to provide a reliable open architecture for loading sequences from different data sources and for

performing versatile analyses on these sequences. The new features that are essential for the integration are described as follows:

(i) We have augmented the data model in SeqVISTA to organize input data required by various analysis tools. Some methods require one input sequence at a time, e.g. Possum and Cluster-Buster in Table 1. Others operate on multiple sequences (e.g. Clover, Glam, MotifScanner and MotifSampler). Except for RepeatMasker, PromoSer, MotifSampler and Glam, the remaining programs in Table 1 also require the position-specific scoring matrices of one or more motifs.

(ii) We have implemented constraints within the data model so that multiple sequences can be 'linked'. The output of an analysis program is treated as an entirely new sequence record, since new features have been added by the analysis. However, it would be useful to compare this new record with the original record, as well as with the outputs from other analysis programs. We have implemented the concept of parent/child sequence records and developed the necessary functions to merge their annotations. The 'parent' record is defined as the original sequence loaded by the user while the results from various analysis programs are called 'child' records and are linked to the 'parent' record. This way, we keep the results from different analyses independent while the integrity of each sequence and its annotations remains.

(iii) A published web interface (web service) coupled with the SOAP technology provides a reliable solution for integration. We have developed SeqVISTA functions to directly query any tool that has a SOAP-based web service and retrieve results.

(iv) We have developed a computational pipeline for performing multiple analyses in the background with the option of executing several programs in a user specified order within each analysis. This function takes advantage of the easy data exchange capability provided by the integration. Some of the algorithms in the computational core require intensive computing and it is essential to be able to formulate an automated pipeline so that multiple analyses can be performed in the appropriate sequential order such that the output of one analysis can become the input of the analysis in the next step of the pipeline, without further user intervention.

Input

The user can load an input sequence locally from a file or remotely at a web address (which corresponds to the text file of the sequence record). The latter facilitates easy exchange of sequence records among different labs through the Internet. SeqVISTA also allows the user to supply a GenBank Identification (GI) number or an accession number and it will retrieve the sequence record directly from the NCBI server. For local files, different data formats are identified by the filename extension and assigned to a corresponding parser through an XML-based configuration file (SeqVISTA.xml, located at the SeqVISTA install directory). Similarly, a file over the web can be identified by its base Universal Resource Locator (URL) and assigned to the corresponding parser. For example, the URL http://zlab.bu.edu/muscle_mouse.fasta will trigger SeqVISTA to interpret the data using the FASTA parser. For computational efficiency, parsers are loaded into SeqVISTA at run time.

For each web service integrated within SeqVISTA, we have developed the corresponding service adaptor. A service adaptor allows the user to enter various parameters to modify the behavior of the corresponding web service. The service adaptors, together with the default parameters of the corresponding web service, are also configurable in the configuration file SeqVISTA.xml mentioned above. The parameters of a service adaptor are categorized into three groups: required, optional and advanced.

For the eight services integrated in SeqVISTA, the required categories of adaptor parameters in general include two items: sequence and motif. There are three options for the input sequence, allowing the user to perform analysis on the active sequence (at any point in time there is one active sequence), selected sequences, or all the sequences. A sequence can be selected by mouse-clicking the check box in front of the sequence in the tree panel of SeqVISTA (Figure 1; left panel). The sequence selection functions can also be invoked by right-mouse-clicking in the tree panel or by selecting from the Edit menu tab. SeqVISTA allows the user to select one or multiple motifs from a list provided by the corresponding service. For example, Possum, Clover and Cluster-Buster allow a flexible selection of motifs from a tree-structured list organized by motif families in the JASPAR database (31). The user can also upload motif matrices.

The optional category is typically service specific. For the services that have not had corresponding web interfaces before (Clover and Glam), we provide instant tool-tips when the user passes the mouse over the corresponding field. A detailed explanation of these parameters can be found in SeqVISTA's user manual.

The advanced category provides parameters that are only useful for advanced users in special cases. For example, the user can redirect SeqVISTA to run the RepeatMasker service at another server that is faster by changing the service address.

Output

We continue to use the three-panel format for SeqVISTA's display window, as described in detail previously (6). Figure 1 is a screen shot of the SeqVISTA window displaying four original input mouse sequences and the results of MotifSampler, Clover, Possum, MotifScanner and Glam on these sequences. The left panel (tree panel) is a tree structure of all sequences and their features (each feature in a different color). Predicted motif sites by the various programs are represented as features. The top-right panel (graphics panel) contains graphical depictions of the features indicated by colored boxes, with features on the + strand drawn above the line representing the sequence and features on the − strand below the line. The location and width of each colored box represent the location of the feature in the sequence and the number of bases to which the feature corresponds. The lower-right panel (sequence panel) contains the nucleotide sequence for the active sequence record. The three panels are dynamically linked; if the user selects a feature in one panel by mouse

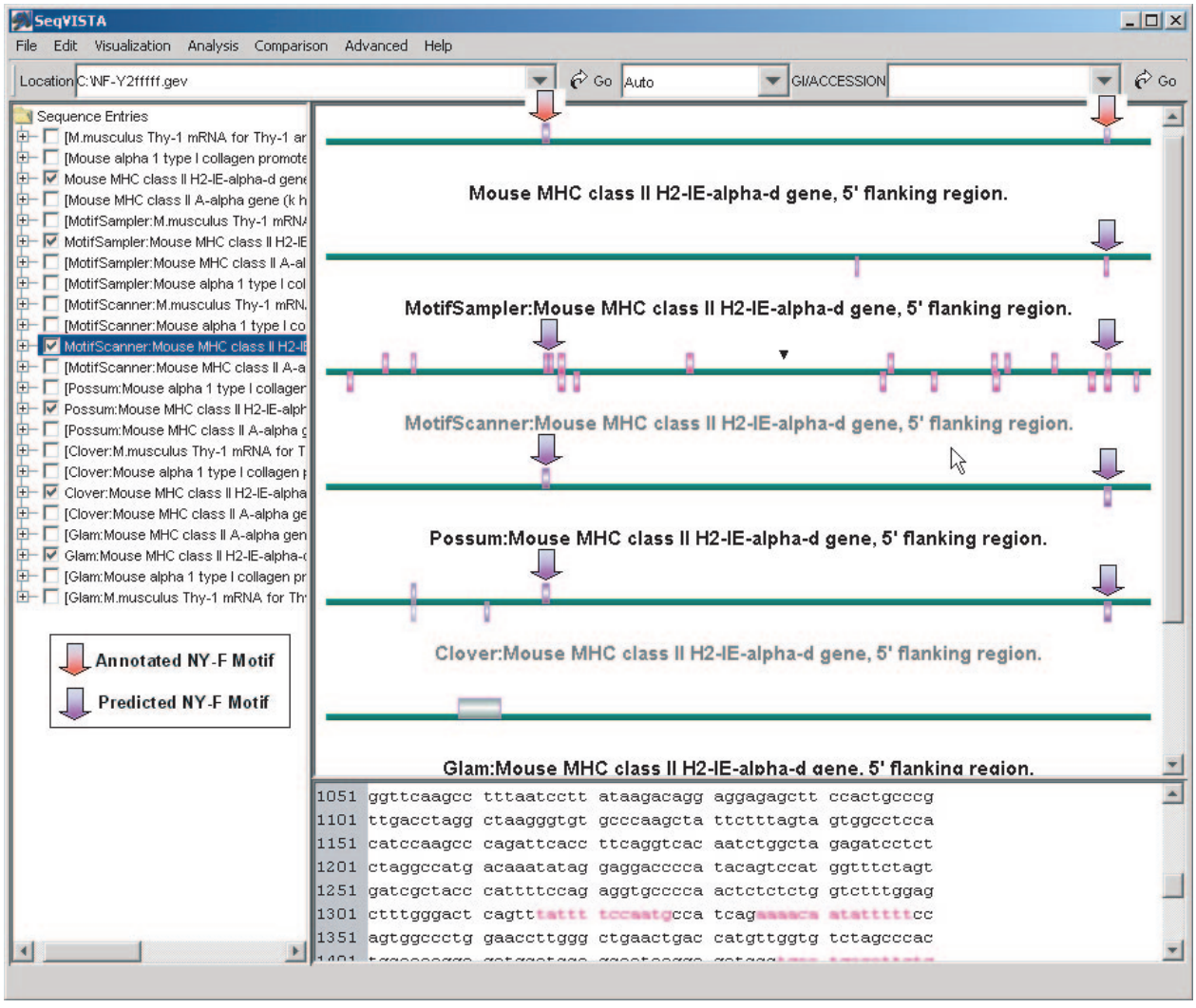

Figure 1. Screenshot of SeqVISTA with multiple analysis results. The top sequence in the upper-right graphical panel corresponds to the original input sequence of mouse Thy-1. There are two binding sites for transcription factor NF-Y, as indicated by two arrows above this sequence. The entire set of vertebrate motifs in TRANSFAC was used for MotifScanner and the entire JASPAR set of motifs was used for Clover. Only the NF-Y motif was used for Possum. In this example, the highest scoring motif identified by Glam does not correspond to either of the annotated sites. For clarity, some sequences are hidden, indicated by the square brackets around their names in the left tree panel.

clicking, the corresponding feature and sub-sequence will be displayed and highlighted accordingly. The user can hide selected sequences or features to improve clarity. One sequence may contain one or more plots in the graphics panel to represent the results of analysis programs. The user can show, hide or delete individual plots.

An analysis produces another sequence record with features corresponding to the results. Such a record is saved in SeqVISTA as a new record, but linked to the original sequence record that was used as the input for the analysis. The name of the analysis program is appended to the beginning of the input sequence name to produce the output sequence name, which allows the user to clearly identify the parent/child relationship between these two sequences as described above. Figure 1 compares the outputs of five programs on a set of four input sequences. For clarity, the results for only one sequence is shown, along with the original input sequence with experimentally annotated motif binding sites indicated with an arrow. It is apparent that the results of multiple programs are easily contrasted in such a visualization setting. The user can obtain the details of an individual prediction by mousing over the corresponding colored box in the graphics panel. A tool tip will appear with the details of the prediction (e.g. from and to positions and score).

A BATCH FUNCTION FOR PIPELINING MULTIPLE ANALYSES

The unified data model and the integration of abundant analysis programs provided by our infrastructure enable the flexibility of performing batch data analysis with multiple programs on a selected set of sequences. In addition, a pipelined batch process can be carried out sequentially such that the result of one analysis can serve as the input for another analysis. The analyses are done in the background and the user can perform visualization in the meantime. We have developed the corresponding software components to control the

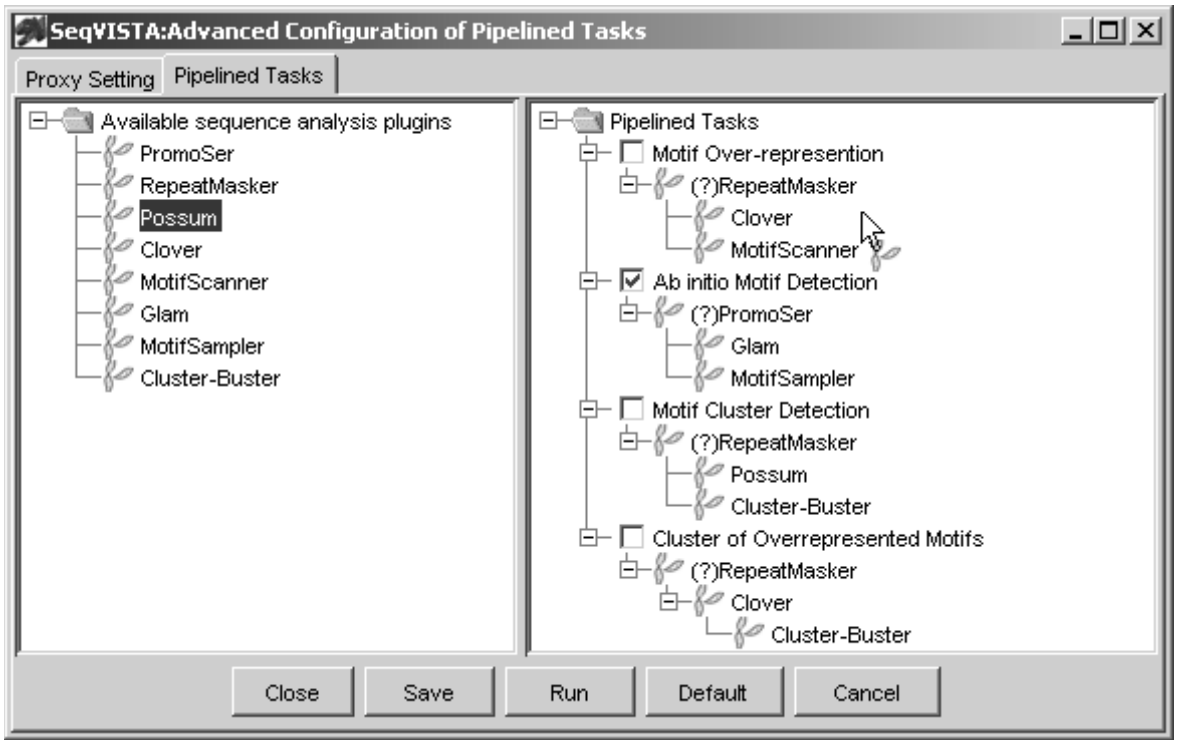

Figure 2. Screenshot of the configuration window for constructing pipelined tasks. The left panel lists available web services. These components can be dragged to the right panel to construct new pipelines. We have listed three predefined pipelines in the right panel and new pipelines can be created through the pop-up menu in the right panel combined with mouse drag-and-drop.

batch process in a separate thread. These components provide a graphical user interface (GUI) to let the user configure the batch process (see Figure 2 and explanation below). The user may also change the default parameters (such as cutoff values) for different analysis services using this GUI. SeqVISTA will check whether the analysis program takes multiple sequences as input. If not, the sequences can be fed to the analysis program one by one without intervention from the user.

Figure 2 illustrates the GUI used to facilitate the construction and management of pipelined tasks. The left panel lists all the available services that can be used to construct a pipeline while the right panel lists the created pipelines. A pipeline is defined as a set of tasks that will be carried out with specified web services either in parallel or sequentially. Each pipeline node is marked with a checkbox, which allows a subset of pipelines to be carried out. The right panel of Figure 2 shows four example pipelines. The user can drag a web service from the left panel and drop it into the right panel and configure it with the pop-up menu in the right panel to create new pipelines. The task nodes within a pipeline are maintained using a hierarchical structure in which the tasks at the same level will be carried out in parallel while sub-tasks will be carried out sequentially. For example, the second pipeline shown in Figure 2 (*Ab initio* Motif Detection) indicates that PromoSer will be first run to obtain the promoter sequences, which are then used as input to Glam and MotifSampler, which will be run in parallel. Each task node can be configured to change the default parameters of the service it represents. SeqVISTA recognizes the dependencies of the tasks within a pipeline and will require a specific order of configuration whenever necessary. For services that always require to be configured (e.g. PromoSer requires the user to enter a list of GenBank accession numbers), their task nodes are indicated with a '?' as shown in Figure 2. The '?' will disappear once the node is configured.

The created pipeline can be saved and reused next time, with different input sequences and reconfiguration of individual services. By default the running pipeline is represented as an animated button at the task queue panel. The progress of the pipelined tasks can be examined by clicking on the button. The corresponding results of the pipelined analysis, however, will be only added to SeqVISTA when all tasks have been finished.

ACKNOWLEDGEMENTS

We thank Brian Pierce and Heather Burden for thoroughly proofreading the manuscript. This work has been supported in part by NSF grants DBI-0078194 and MRI DBI-0116574 and NIH grants 1P20GM066401-01, 1R01HG03110-01 and A08-POGM66401A.

REFERENCES

1. Stein,L. (2002) Creating a bioinformatics nation. *Nature*, **417**, 119–120.
2. Stein,L.D. (2003) Integrating biological databases. *Nature Rev. Genet.*, **4**, 337–345.
3. van Helden,J. (2003) Regulatory sequence analysis tools. *Nucleic Acids Res.*, **31**, 3593–3596.
4. Aerts,S., Thijs,G., Coessens,B., Staes,M., Moreau,Y. and Moor,B.D. (2003) Toucan: deciphering the *cis*-regulatory logic of coregulated genes. *Nucleic Acids Res.*, **31**, 1753–1764.

5. Coessens,B., Thijs,G., Aerts,S., Marchal,K., De Smet,F., Engelen,K., Glenisson,P., Moreau,Y., Mathys,J. and De Moor,B. (2003) INCLUSive: A web portal and service registry for microarray and regulatory sequence analysis. *Nucleic Acids Res.*, **31**, 3468–3470.
6. Hu,Z., Frith,M., Niu,T. and Weng,Z. (2003) SeqVISTA: a graphical tool for sequence feature visualization and comparison. *BMC Bioinformatics*, **4**, 1.
7. Halees,A.S., Leyfer,D. and Weng,Z. (2003) PromoSer: A large-scale mammalian promoter and transcription start site identification service. *Nucleic Acids Res.*, **31**, 3554–3559.
8. Frith,M.C., Fu,Y., Yu,L., Chen,J.F., Hansen,U. and Weng,Z. (2004) Detection of functional DNA motifs via statistical over-representation. *Nucleic Acids Res.*, **32**, 1372–1381.
9. Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.F. and Wootton,J.C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, **262**, 208–214.
10. Liu,X., Brutlag,D.L. and Liu,J.S. (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.*, 127–138.
11. Roth,F.P., Hughes,J.D., Estep,P.W. and Church,G.M. (1998) Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.*, **16**, 939–945.
12. Thijs,G., Marchal,K., Lescot,M., Rombauts,S., De Moor,B., Rouze,P. and Moreau,Y. (2002) A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J. Comput. Biol.*, **9**, 447–464.
13. Frith,M.C., Hansen,U., Spouge,J.L. and Weng,Z. (2004) Finding functional sequence elements by multiple local alignment. *Nucleic Acids Res.*, **32**, 189–200.
14. Karlin,S. and Altschul,S.F. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl Acad. Sci. USA*, **87**, 2264–2268.
15. Thijs,G., Lescot,M., Marchal,K., Rombauts,S., De Moor,B., Rouze,P. and Moreau,Y. (2001) A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics*, **17**, 1113–1122.
16. Marchal,K., Thijs,G., De Keersmaecker,S., Monsieurs,P., De Moor,B. and Vanderleyden,J. (2003) Genome-specific higher-order background models to improve motif detection. *Trends Microbiol.*, **11**, 61–66.
17. Prestridge,D.S. (1995) Predicting Pol II promoter sequences using transcription factor binding sites. *J. Mol. Biol.*, **249**, 923–932.
18. Kondrakhin,Y.V., Kel,A.E., Kolchanov,N.A., Romashchenko,A.G. and Milanesi,L. (1995) Eukaryotic promoter recognition by binding sites for transcription factors. *Comput. Appl. Biosci.*, **11**, 477–488.
19. Crowley,E.M., Roeder,K. and Bina,M. (1997) A statistical model for locating regulatory regions in genomic DNA. *J. Mol. Biol.*, **268**, 8–14.
20. Frech,K., Danescu-Mayer,J. and Werner,T. (1997) A novel method to develop highly specific models for regulatory units detects a new LTR in GenBank which contains a functional promoter. *J. Mol. Biol.*, **270**, 674–687.
21. Wasserman,W.W. and Fickett,J.W. (1998) Identification of regulatory regions which confer muscle-specific gene expression. *J. Mol. Biol.*, **278**, 167–181.
22. Krivan,W. and Wasserman,W.W. (2001) A predictive model for regulatory sequences directing liver-specific transcription. *Genome Res.*, **11**, 1559–1566.
23. Wagner,A. (1999) Genes regulated cooperatively by one or more transcription factors and their identification in whole eukaryotic genomes. *Bioinformatics*, **15**, 776–784.
24. Frith,M.C., Hansen,U. and Weng,Z. (2001) Detection of *cis*-element clusters in higher eukaryotic DNA. *Bioinformatics*, **17**, 878–889.
25. Frith,M.C., Spouge,J.L., Hansen,U. and Weng,Z. (2002) Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucleic Acids Res.*, **30**, 3214–3224.
26. Berman,B.P., Nibu,Y., Pfeiffer,B.D., Tomancak,P., Celniker,S.E., Levine,M., Rubin,G.M. and Eisen,M.B. (2002) Exploiting transcription factor binding site clustering to identify *cis*- regulatory modules involved in pattern formation in the *Drosophila* genome. *Proc. Natl Acad. Sci. USA*, **99**, 757–762.
27. Markstein,M., Markstein,P., Markstein,V. and Levine,M.S. (2002) Genome-wide analysis of clustered dorsal binding sites identifies putative target genes in the *Drosophila* embryo. *Proc. Natl Acad. Sci. USA*, **99**, 763–768.
28. Rebeiz,M., Reeves,N.L. and Posakony,J.W. (2002) SCORE: a computational approach to the identification of *cis*-regulatory modules and target genes in whole-genome sequence data. Site clustering over random expectation. *Proc. Natl Acad. Sci. USA*, **99**, 9888–9893.
29. Rajewsky,N., Vergassola,M., Gaul,U. and Siggia,E.D. (2002) Computational detection of genomic cis-regulatory modules applied to body patterning in the early *Drosophila* embryo. *BMC Bioinformatics*, **3**, 30.
30. Frith,M.C., Li,M.C. and Weng,Z. (2003) Cluster-Buster: Finding dense clusters of motifs in DNA sequences. *Nucleic Acids Res.*, **31**, 3666–3668.
31. Sandelin,A., Alkema,W., Engstrom,P., Wasserman,W.W. and Lenhard,B. (2004) JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.*, **32**, D91–D94.