# STORMSeq: A Method for Ranking Regulatory Sequences by Integrating Experimental Datasets with Diverse Computational Predictions

Jim C. Huang and Brendan J. Frey

## 1 Introduction

The problem of sequence search, such as discovering transcription factor (TF) binding sites, microRNA targets and structural genetic variants, remains a significant challenge in genomics. Several *de novo* computational methods have been developed with the aim of searching for overrepresented sequences using sequence data [2, 12]. Due to the degeneracy of such sequences, such methods often require the use of sequence conservation in order to minimize false positive rates. To address this, computational methods have recently begun to account for additional features such as the accessibility of target sequences due to RNA secondary structure [15], contextual features [7] or other types of quantitative profiling data [4, 8, 19, 20]. As newer methods for discovering sequences and new profiling technologies continue to emerge, the issue of how to update existing sequence search methods to account for multiple types of data remains a significant challenge. In addition to accounting for several types of data, incorporating the large number of computational predictions already available will also be desirable.

Jim C. Huang
Microsoft Research
One Microsoft Way, Redmond, WA, 98052, USA
e-mail: jimhua@microsoft.com

Brendan J. Frey
Probabilistic and Statistical Inference Group, University of Toronto
10 King's College Road, Toronto, ON, M5S 3G4, Canada
e-mail: frey@psi.toronto.edu

### 1.1 Previous work

In recent years, many different methods have been proposed to address the problem of integrating together large heterogeneous datasets in the context of sequence search. For example, probabilistic generative models have been proposed in which sequence search consists of inference and learning [8, 19] given sequence and expression data. Although such methods explicitly model the impact of sequences on gene expression while accounting for uncertainty, a major challenge is to account for newer datasets as well as new sources of regulatory variability. Each additional dataset to be analyzed is likely to introduce a significant number of additional parameters and hidden variables, dramatically increasing the cost and complexity of inference and learning under the generative framework. Thus, as the number of types and sizes of data continue to increase, it is likely that both model misspecification and prohibitive computational complexity will hamper the practicality of probabilistic models with latent variables for discovering sequences. Owing to the difficulty in developing purely sequence-based models of regulatory sequences, a major challenge is then to incorporate additional data types under a unified tractable and principled framework.

### 1.2 Sequence search as a problem of learning to rank

A strategic approach to the above problem can be obtained by noting that the problem of sequence search is inherently a problem of learning to rank, whereby we are given a large number of possible sequences and only some relatively small number are of biological significance. Furthermore, there is often a well-defined notion of preference between sequences. An example of this arises when searching for transcription factor sites, whereby some sites are more strongly bound than others by certain transcription factors. Thus when discovering sequences, it is desirable to explicitly model the fact that sequences do not fall into two distinct categories of positives and negatives but instead have different degrees of significance attached to them, so that a plausible model should assign a higher score for sequences with higher importance.

Some methods have in fact formulated the problem of sequence search as one of ranking, so that they assign a score to each sequence with the implicit assumption that high-scoring sequences are more likely to be *bona fide* than low-scoring ones. The idea of discovering sequences using an explicit ranking formulation has been explored previously by [3, 4, 20] in the context of using the orderings obtained from microarray intensities to learn position-specific scoring matrices (PSSMs) for transcription factor binding sites (TFBS). This was shown to significantly improve predictive accuracy with respect to other model-based methods, as no assumptions on the functional relationship between measured intensities and sequences needed to be made in order to learn to rank sequences. The improved accuracy of such ranking-based methods with respect to model-based methods then suggests that a

good method for discovering sequences would be one specifically tailored to the problem of learning to rank.

Given the above methods for ranking sequences, our goal here is to expand on previous work along three directions. First, we address the presence of statistical dependence relationships between variables in the problem of ranking, since the rank of one sequence can only be determined given the ranks of all sequences. Second, the scoring function used by the previous methods of [3, 4, 20] was parameterized by a PSSM and so only accounted for sequence inputs. Here we will allow for ranking functions which can account for rich feature spaces obtained from quantitative measurements such as expression profiling data. Lastly, by formulating the sequence search problem as one of ranking, we can leverage information across several experimental datasets and diverse prediction methods via the orderings over sequences that each provides. Under the framework of learning to rank, orderings provided by diverse computational methods and those provided by experimental data are all comparable and readily accounted for, even if measured/predicted values between different prediction methods may be difficult to compare. Thus, given that we observe many different partial orderings provided by diverse datasets and prediction methods, our aim will be to predict orderings over sequences so that sequences which are often highly ranked across different experiments and prediction methods should also be highly ranked by our method. The proposed framework of ranking then offers three significant advantages over previous approaches for sequence search. First, the framework makes minimal assumptions about the relationships between sequences and measured/predicted labels for the sequences and so largely avoids the issue of model misspecification. Second, it allows us to leverage orderings provided by heterogeneous datasets and prediction methods which may have little overlap with one another in the sequences they contain, but are nevertheless informative when combined together under a single model. Lastly, predictive accuracy is improved by explicitly modelling the dependencies involved in learning to rank.

To model the statistical dependencies in learning to rank, we can take advantage of the structured ranking learning framework which was recently proposed in [10]. This probabilistic framework for learning to rank is based on a novel class of probabilistic graphical models called cumulative distribution networks [9, 11], or CDNs. In learning to rank in a structured setting where we account for dependence relationships between model variables, or *structured ranking learning*, the goal is to learn a ranking function under a structured loss functional which accounts for the statistical dependence relationships involved in predicting pairwise preferences between sequences, as misranking one sequence affects how we rank other sequences. In the context of discovering sequences, we can then interpret a set of prediction methods and a set of experimental measurements as observations which convey partial orderings over some subset of the sequences of interest. Thus we present STORMSeq, a method formulated which scores sequences given a set of features and a set of orderings over subsets of the sequences to be ranked. Our method generalizes the RankMotif++ method of [4] to a structured learning setting where we can A) account for the dependencies in the problem of ranking, B) incorporate rich feature

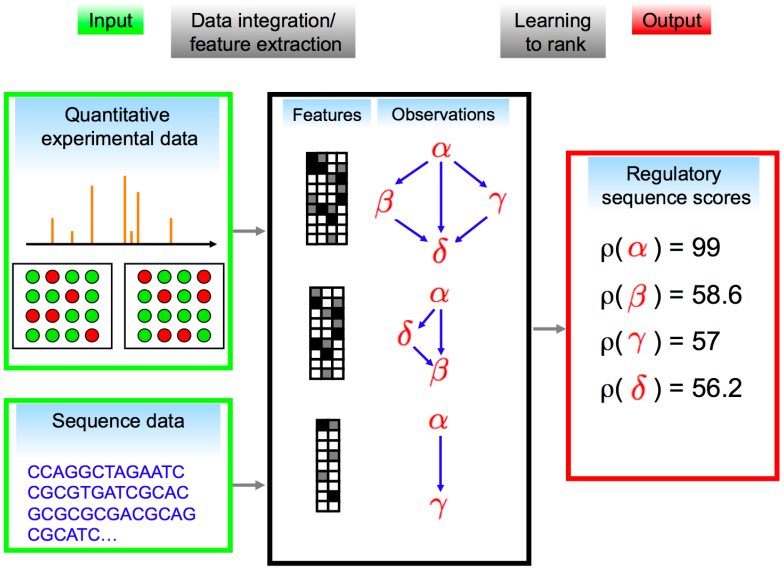



**Fig. 1** The STructured ranking of Regulatory Motifs and Sequences (STORMSeq) method. Given multiple independent observations conveying various orderings over sequences and given the observed sequences and input features extracted for each observation (e.g.: mRNA, microRNA and protein measurements, sequence context features), STORMSeq learns a ranking function such that the probability of generating the observed orderings is maximized.

spaces such as quantitative measurements of mRNA and protein expression in addition to sequence data, and C) account for diverse computational prediction methods as additional data. The outline of the method is illustrated in Figure 1. We will apply the proposed framework to the problems of scoring transcription factor binding sites and microRNA targets, although the framework is general enough to be applied to a wide variety of bioinformatics problems, such as ranking therapeutic drug targets, finding genetic associations or scoring protein-protein interactions.

## 2 STORMSeq: STructured ranking of Regulatory Motifs and Sequences

We will begin by describing the problem of structured ranking learning for discovering sequences using the framework of [10]. Suppose we wish to score sequences in the set $\mathscr{S}$. Let $s_\alpha$ be a particular sequence in $\mathscr{S}$ which is indexed by $\alpha$. Here, a sequence is any segment of nucleotides or amino acids for which one can extract features. For example, in the case where we wish to discover microRNA targets, a 'sequence' may correspond to the entire 3' untranslated region (3'UTR) for a particular gene, so that one has access to the sequence of the 3'UTR, as well as other

features for the 3'UTR sequence. These can include its level of expression across many tissues/cell types, the abundance of proteins which are translated from the sequence preceding the 3'UTR and the expression of a microRNA which putatively targets a site in the 3'UTR sequence. This is illustrated in Figure 2(a): for each node $\alpha$, we are provided with a corresponding sequence $s_\alpha$ and a set of features $\mathbf{x}_\alpha$ which will aid in learning to rank the sequences.

Suppose now that we are given a set of $N$ observations $\mathscr{D} = \{D_1, \cdots, D_N\}$, where each observation $D_n$ provides an ordering of the sequences in some subset $\mathscr{S}_n \subseteq \mathscr{S}$. Here, an observation contains a partial ordering of the sequences to be ranked. For example, in the context of scoring microRNA targets, orderings might be provided by gene expression values in microRNA overexpression experiments [8] or they can be provided by scores output by computational prediction methods [7, 13, 17]. The orderings over sequences in an observation can then be viewed as a set of pairwise preference relationships between sequences, which we will denote using $\alpha \succ \beta$. For a given observation, we can then represent the ordering between sequences as a directed graph in which a directed edge $e = (\alpha \to \beta)$ is drawn between two nodes $\alpha, \beta$ if sequence $s_\alpha$ was *preferred* to sequence $s_\beta$ within observation $D_n$. We will denote this directed graph as the order graph $G_n = (V_n, E_n)$ for observation $D_n$, where $E_n$ is the set of all edges in the order graph and each node $\alpha \in V_n$ corresponds to a unique sequence $s_\alpha \in \mathscr{S}_n$. An example of such an order graph is shown in Figure 2(b). Thus the $n^{\text{th}}$ observation consists of the set $D_n = \{G_n, \{s_\alpha, \mathbf{x}_\alpha\}_{\alpha \in V_n}\}$, so that our data consists of a collection of independent observations $\mathscr{D} = \{D_1, \cdots, D_N\}$. One immediate advantage of the proposed framework is that orderings over sequences can be compared between observations despite the fact that measured/predicted values between observations may not be comparable. Furthermore, the orderings conveyed by different observations can be partial and can be defined over different subsets of sequences.

To combine the different orderings together, we now define a *ranking function* $\rho(\alpha) : V_n \to \mathbf{R}$ which assigns real-valued scores to sequences. If we model the stochastic score $\sigma_\alpha$ of a given node $\alpha$ as

$$\sigma_\alpha = \rho(\alpha) + \pi_\alpha, \tag{1}$$

where $\pi_\alpha$ is a random variable specific to node $\alpha$, then we can define the preference event $\alpha \succ \beta$ as being equivalent to the following:

$$\alpha \succ \beta \Leftrightarrow \pi_{\alpha\beta} \equiv \pi_\beta - \pi_\alpha \leq \rho(\alpha) - \rho(\beta). \tag{2}$$

Here, $\pi_{\alpha\beta}$ is a *preference variable* between $\alpha, \beta$. Thus for each edge $(\alpha, \beta)$ in the order graph $G_n$, we assign a corresponding continuous-valued preference variable $\pi_{\alpha\beta}$ which should satisfy the above inequality in order for the preference relation $\alpha \succ \beta$ to be observed. Now we can define the quantity $r(e; \rho, D_n) = \rho(\alpha) - \rho(\beta)$ and collect these into a vector $r \equiv r(D_n; \rho) \in \mathbf{R}^{|E_n|}$ of pairwise differences, where $|E_n|$ is the number of edges in the order graph. Similarly, let $\pi_e \equiv \pi_{\alpha\beta}$ be the preference variable defined along edge $e$ in the order graph $G_n$. Having defined the preference variables, we must now select an appropriate loss measure for learning the ranking

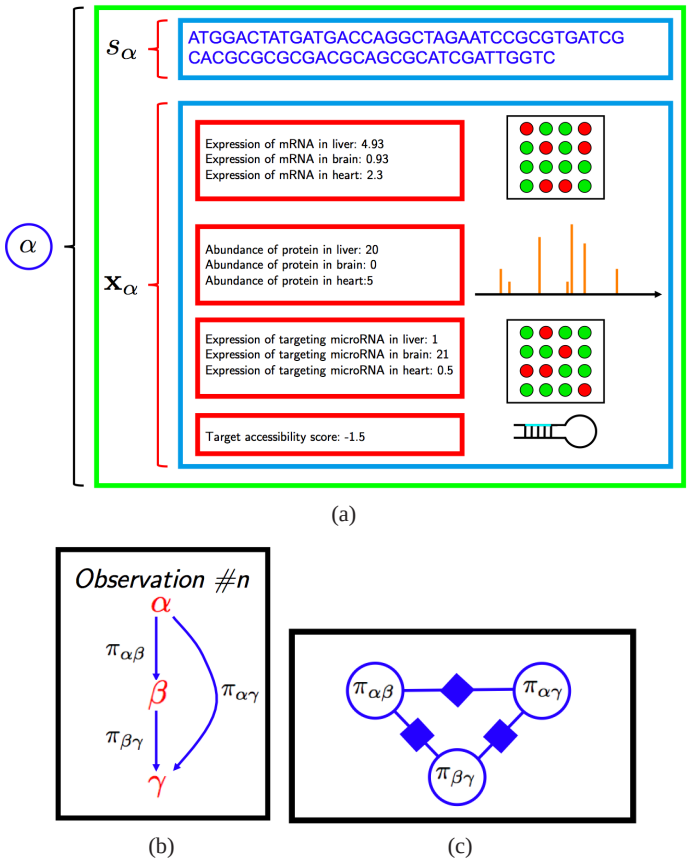



**Fig. 2** The STructured ranking of Regulatory Motifs and Sequences (STORMSeq) method. a) Feature extraction. For each sequence $s_\alpha$ to be ranked, we assign a corresponding node $\alpha$ and a set of corresponding features which are relevant to ranking the sequence. For the example shown, the sequence $s_\alpha$ may correspond to the sequence for the entire 3' untranslated region (3'UTR) of a gene, so that the feature vector $\mathbf{x}_\alpha$ include the expression of the gene carrying the sequence, the abundance of protein produced from the coding region for the gene carrying the sequence and the expression of a putative microRNA which targets the sequence; b) An observation consisting of an order graph over three nodes where each node $\alpha, \beta, \gamma$ in the order graph corresponds to a unique sequence $s_\alpha, s_\beta, s_\gamma$ to be ranked, and each directed edge expresses a preference relationship between two nodes. An order graph can be readily established from log p-value scores, expression ratios or other available statistics which provide an indication of the relevance or importance of a given sequence. In this example the order graph corresponds to the ordering $\alpha \succ \beta \succ \gamma$. Each edge in the order graph then corresponds to preference variables $\pi_{\alpha\beta}, \pi_{\beta\gamma}, \pi_{\alpha\gamma}$; c) The corresponding cumulative distribution network (CDN) defined over the preference variables specified by the observation of b). The CDN models the joint CDF over the preference variables and allows us to compactly specify dependencies between preferences so we can perform structured ranking learning [10]

function. For a given observation $D_n$, we will choose the loss measure to be the neg-

ative log-probability of observing the preference relationships between sequences in order graph $G_n$. From Equation (2), this will take the form of a probability measure over events of the type $\pi_e \leq r(e;\rho,D_n)$ so that we obtain

$$\mathbb{P}[E_n|V_n,\rho] = \mathbb{P}\left[\bigcap_{e\in E_n}[\pi_e \leq r(e;\rho,D_n)]\right] = F_\pi\big(r(D_n;\rho)\big), \tag{3}$$

where $F_\pi$ is the joint CDF over the preference variables $\pi_e$. Thus, for a given observation $D_n$, *any* probability over the set of preference events $\pi_{\alpha\beta} \leq r(e;\rho,D_n)$ will take on the form of a joint CDF $F_\pi\big(r\big)$ over the preference variables $\pi \equiv \{\pi_{\alpha\beta}\}_{(\alpha,\beta)\in E_n}$, where the CDF $F_\pi$ is evaluated at $r(D_n;\rho)$.

Given multiple independent observations $\mathscr{D} = \{D_1,\cdots,D_N\}$, we can then define a *structured loss functional* $\mathscr{L}(\mathscr{D};\rho,F_\pi)$ as the log-probability of independently generating the observed orderings in $\mathscr{D}$, so that

$$\mathscr{L}(\mathscr{D};\rho,F_\pi) \equiv -\sum_{n=1}^{N}\log F_\pi\big(r\big) \tag{4}$$

where each term in the loss functional is the log of a joint CDF. Whilst each of these log-CDF terms is defined over many preference variables with a high degree of dependence amongst variables, we can nevertheless represent each term compactly as a cumulative distribution network (CDN) [9, 11], which is a graphical model representing the joint CDF of several random variables (see Appendix). An example of a possible CDN representing a joint CDF over three pairwise preferences is shown in Figure 2(c).

Having defined the structured loss functional $\mathscr{L}(\mathscr{D};\rho,F_\pi)$, the problem of learning to rank sequences from observations $D_1,\cdots,D_N$ will then consist of minimizing the loss functional with respect to the ranking function $\rho$ and the CDF $F_\pi$. Let $\theta$ denote the vector of parameters which parameterize both the ranking function $\rho$ and the joint CDF $F_\pi$, so that we can write the structured loss as a function of $\theta$, or

$$\mathscr{L}(\mathscr{D};\theta) \equiv \mathscr{L}(\mathscr{D};\rho,F_\pi) = \sum_{n=1}^{N}\mathscr{L}(D_n;\theta) = -\sum_{n=1}^{N}\log F_\pi\big(r(D_n;\theta)\big). \tag{5}$$

In order to optimize $\mathscr{L}(\mathscr{D};\theta)$ with respect to $\theta$, we will assume that we can compute the gradient $\nabla_\theta\mathscr{L}(D_n;\theta)$ for each observation $D_n$. Given the gradient, we can then proceed to optimize the structured loss functional using a stochastic gradients descent (SGD) algorithm whereby for each observation $D_n$, we construct a CDN for order graph $G_n$ and we update the parameters of the model according to the rule $\theta \leftarrow \theta - \mu\nabla_\theta\mathscr{L}(D_n;\theta)$, where $\mu$ is a learning rate parameter for the SGD algorithm. This leads to an efficient method for learning to rank, as we only need to store the CDN for a single observation for the purpose of computing a gradient and updating the model parameters: this is illustrated graphically in Figure 3.

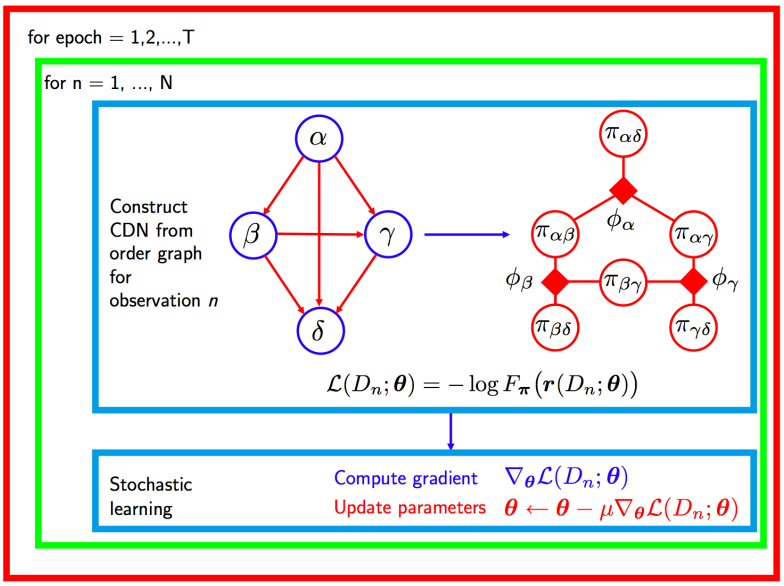



**Fig. 3** Illustration of the STORMSeq framework. For each observation $D_n$, we construct a CDN defined over preference variables corresponding to edges in the order graph $G_n$ (top). For this example, we have an order graph defined over four nodes and six preference variables. The CDN then models the joint CDF over the six preference variables as a product of functions: here the model consists of a product of three functions so that $F_{\pi}\Big(\mathbf{r}(D_n;\theta)\Big) = \phi_{\alpha}(r_{\alpha,\delta}, r_{\alpha,\beta}, r_{\alpha,\gamma})\phi_{\beta}(r_{\alpha,\beta}, r_{\beta,\gamma}, r_{\beta,\delta})\phi_{\gamma}(r_{\alpha,\gamma}, r_{\beta,\gamma}, r_{\gamma,\delta})$. Once the CDN has been constructed, we can perform stochastic learning of parameters by computing the gradient of the log-CDF modeled by the CDN and then updating the vector of parameters $\theta$ (bottom). We can then repeat this process for each observation and for a number $T$ of epochs, or passes through the training set.

## *2.1 Ranking using sequence and quantitative features*

In order to adapt the above framework to the problem of ranking sequences, we will use a ranking function $\rho(\alpha)$ which has the general form

$$\rho(\alpha) = \rho_{seq}(s_{\alpha};\mathbf{M}) + \rho_{quant}(\mathbf{x}_{\alpha};\mathbf{w}) \tag{6}$$

where $\rho_{seq}, \rho_{quant}$ are functions which assign scores to the sequence $s_{\alpha}$ and its corresponding feature vector $\mathbf{x}_{\alpha}$. Here, it is possible to specify different parametric forms for $\rho(\alpha)$ which assign scores to sequences under various assumptions. In order to score any given node $\alpha$ based on sequence $s_{\alpha}$ alone, we will consider the sum of contributions of subsequences of $s_{\alpha}$ under the assumption that each subsequence contributes independently to the overall score for $s_{\alpha}$ (see Appendix). We will choose $\rho_{quant}$ to be a linear function of the quantitative features, so that $\rho_{quant}(\mathbf{x}_{\alpha};\mathbf{w}) = \mathbf{w}^T\mathbf{x}_{\alpha}$. Given these parameterizations, a sequence $s_{\alpha}$ will have a higher score if both $\rho_{seq}$ and $\rho_{quant}$ assign high scores to $s_{\alpha}$.

The proposed framework of learning the ranking function from observations is then summarized as follows: given a training set of observations consisting of sequences to be ranked, associated quantitative features and a partial ordering over the sequences, we wish to learn a ranking function $\rho(\alpha)$ which maximizes the probability of generating the observed orderings by assigning higher scores to those sequences which are most consistently highly ranked in the observations $\{D_1,\cdots,D_n\}$. In order to learn $\rho(\alpha)$, we can compute the gradients $\nabla_{\mathbf{M}}\rho_{seq}(s_\alpha;\mathbf{M}),\nabla_{\mathbf{w}}\rho_{quant}(\mathbf{x}_\alpha;\mathbf{w})$ (see Appendix) in order to perform gradient-based learning. The ranking function is such that we can account for sequence data in addition to other quantitative features such as expression measurements. The use of CDNs to represent the structured loss functional for learning to rank then allows us to account for the fact that learning to rank is inherently a problem in which one must account for the presence of statistical dependence relationships between model variables.

We emphasize at this juncture that STORMSeq has been formulated in a general way so that it is applicable to many different problems in which we wish to learn a ranking function using multiple instances of orderings, sequence data and other quantitative features. To illustrate how STORMSeq might be used in practice, we will apply it to two problems of sequence search. In the first of these problems, we will score sequences bound by transcription factors using the protein binding microarray data of [3]. In the second, we will score targets of the let-7b microRNA in human retinoblastomas using both microRNA overexpression data [8] and other quantitative features such as protein abundance and mRNA expression levels of targets. Before we proceed, it will be instructive to study the relation between STORMSeq and a previous method for learning to rank sequences from orderings over sequences obtained from microarray measurements.

## *2.2 The RankMotif++ model as a cumulative distribution network*

It is worth noting that in the RankMotif++ model of [4], the objective being minimized corresponds to the log-CDF over preferences under the assumption that preference variables are mutually independent. More precisely, in RankMotif++ the loss function is given by $\mathscr{L}(\theta)=\log F_\pi(\mathbf{r}(D_n))$, where the probability over all pairwise preferences $\alpha\succ\beta$ is represented by a product over logistic functions of $r_{\alpha\beta}=\rho(\alpha)-\rho(\beta)$ so that

$$F_\pi(\mathbf{r}(D_n))\equiv\mathbb{P}\big[\pi\leq\mathbf{r}(D_n)\big]=\prod_s\frac{1}{1+\exp(-\nu r_s)}=\prod_{\alpha\succ\beta}\frac{1}{1+\exp\big(-\nu\big(\rho(\alpha)-\rho(\beta)\big)\big)} \tag{7}$$

with $\rho(\alpha)=\rho_{seq}(s_\alpha)$ and $\nu>0$. Thus the above loss function can be represented using a disconnected CDN model where each function node corresponds to the CDN function $\phi_s(r_s)=\big(1+\exp(-\nu r_s)\big)^{-1}$ and all pairwise object preferences are modeled as being independent of one another.

# 3 Results

## *3.1 Discovering transcription factor binding profiles*

We will first apply the proposed structured ranking learning framework to the problem of ranking sequences using measurements from a protein binding microarray (PBM) experiment. We obtained PBM data from the Supplementary Material section of [3], which consisted of measured intensities of 35-mer probes bound by five different transcription factors Cbf1, Ceh-22, Oct-1, Rap1, Zif268 across two experimental replicate arrays *Array 1* and *Array 2*. The PBM data consisted of intensity measurements $y_\alpha$ for a set of sequences $\{s_\alpha \in \mathscr{S}\}$, where each probe on the array is indexed by $\alpha$ and $s_\alpha$ denotes the nucleotide sequence of a given probe on the array. We used the array labeled *Array 1* as our training data and the probe measurements from *Array 2* as test data. The goal here is to then learn a ranking function which assigns scores to probe sequences under the assumption that higher scores should indicate an increased probability of a TF binding to a sequence.

We applied the STORMSeq method and evaluated the resulting ranking function on the test set. In order to compare STORMSeq to similar methods, we also ran the MatrixREDUCE [6], MDScan [18], Prego [20] and RankMotif++ methods on the same training data and evaluated these on the same test data using the settings specified by [4] (see Appendix for details). Here we applied STORMSeq without using additional quantitative features to provide a fair comparison to the other methods which rank sequences using only sequence data. The performance of all five methods for the above five TFs are summarized in Figures 4(a) and 4(b) using precision versus recall curves, as well as Normalized Discounted Cumulative Gain [14] curves which account for how well a method ranks high-intensity sequences (see Appendix). The use of the NDCG metric here is well-suited to the problem at hand, as the truncation level *n* can be interpreted as the number of sequences to be further validated or analyzed, so that a higher NDCG value is obtained if the most significant sequences appear at the top of the list in their correct order of significance. Here, the significance of a sequence is determined by the strength with which a transcription factor binds to it, so that the highest score should be assigned to the most strongly bound sequence. Figures 4(a) and 4(b) demonstrate that by ranking in a structured learning setting and by making no particular assumption about the relationship between sequence s and measured PBM intensities, we increase predictive accuracy as measured by precision, recall and NDCG compared to the other unstructured prediction methods such as RankMotif++. In particular, according to the NDCG metric, our method of ranking also has increased accuracy in terms of the ranking itself, so that sequences with higher intensities are more likely to be ranked higher by STORMSeq than by the other models.

The corresponding PSSMs found by each of the above methods are shown in Figure 5. As can be seen, the PSSMs learned by STORMSeq are consistent with those found by the other methods as well as with PSSMs previously reported for this dataset [3, 4]. It is worth noting here that the consensus sequence for RAP1

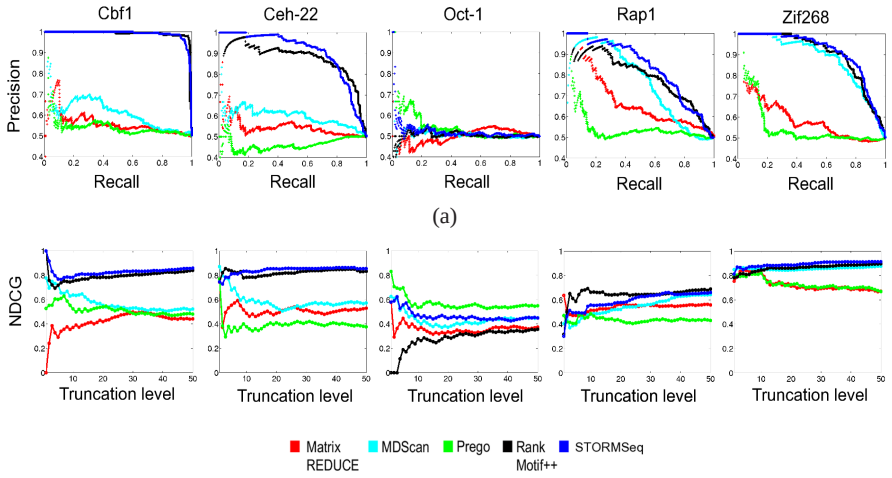



**Fig. 4** a) Precision versus recall using five different methods for the Cbf1, Ceh-22, Oct-1, Rap1, Zif268 transcription factors studied in [3, 4]. The methods shown are MatrixREDUCE (red), MDScan (cyan), Prego (green), RankMotif++ (black) and STORMSeq (blue); b) The corresponding curves showing Normalized Discounted Cumulative Gains (NDCG) versus the truncation level, or the number of top-ranking sequences. Both a) and b) show that by ranking in a structured learning setting using STORMSeq, we generally improve predictive accuracy, in terms of precision, recall and NDCG, with respect to the other unstructured learning methods shown here.

found by our method, as well as the consensus reported by the Prego and MDScan methods agree with the 1st 6 base positions of the widely published motif *ACACCC* [21]. Also, observe that while the PSSMs obtained by STORMSeq can be degenerate at many positions for various TFs, the improved performance of STORMSeq over these methods suggests that these methods are likely to underestimate the degeneracy of the motifs to be discovered as a consequence of model misspecification.

One reviewer has pointed out that the particular sequence ranking function used above is not designed to allow for gaps in motifs [5]. One advantage of the structured ranking learning framework is that the user can choose from many ranking functions for any given problem, so that the user can specify a ranking function which accounts for the presence of gaps, or other specific features of the motifs to be found. In the case where we wish to learn a PSSM for gapped motifs, we can constrain the degenerate positions in the PSSM by constraining the entropy of the nucleotide frequency at these positions: we provide an example of this in the Appendix.

Having applied the structured ranking learning framework to the problem of discovering transcription factor binding sites, we will also demonstrate the usefulness of STORMSeq for discovering microRNA targets, which also consist of short nucleotide sequences which regulate the activity of genes.

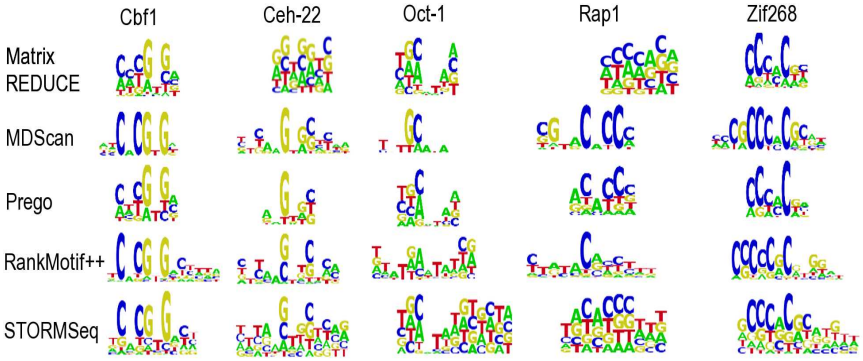



**Fig. 5** Motifs found by the MatrixREDUCE, MDScan, Prego, RankMotif++ and STORMSeq methods (rows) for each of the TFs.

## *3.2 Discovering microRNA targets*

In addition to learning to rank transcription factor binding sites, we will also demonstrate the usefulness of STORMSeq for ranking microRNA targets. MicroRNAs consist of molecules of 22-25 nucleotides which target mRNA transcripts through complementary base-pairing to short target sites, in a fashion analogous to the operation of transcription factors. However, unlike transcription factors, microRNAs are generally inhibitory in their activity, so that microRNA activity generally represses the activity of their target genes either by reducing the abundance of their target mRNA transcripts or by repressing translational activity of their target mRNAs [1, 8]. There is substantial evidence that microRNAs are an important component of the cellular regulatory network, providing a post-transcriptional means to control the amounts of mRNA transcripts and their protein products [1, 7, 8, 15]. As a consequence of their important role in gene regulation, many previous methods have been proposed for performing genome-wide discovery of targets of microRNAs [7, 8, 13, 17].

We will focus here on the let-7b microRNA and a dataset profiling the expression of human mRNAs in WERI-Rb1 retinoblastoma samples after the transfection of a synthetic RNA duplex of the mature let-7b hairpin [8]. Under the assumption that microRNA regulation is causes reduced mRNA expression, pairwise preference relationships between sequences were asserted using the same criteria as in [4], but using negative log-expression-ratios of expression from the let-7b transfections. Thus, the score of a sequence should correspond to the amount of down-regulation by let-7b. We constructed our dataset in a fashion similar to that used in the previous example for transcription factor binding sites (see Appendix). In contrast to the previous problem which had relatively few sources of data variability, here we are provided with *in vivo* expression measurements of genes which may have several different regulators, some of which may themselves be regulated by let-7b. The problem of

scoring microRNA targets is therefore representative of the type of problem more commonly encountered in genomics, where the goal is to discover sequences in the presence of many sources of *in vivo* regulatory variability. The hypothesis here is that we can leverage additional information in the form of independent quantitative measurements and computational predictions in order to better account for the variability in orderings over sequences.

To learn to rank microRNA targets, we used human 3'UTR sequence data, mouse mRNA expression, mouse let-7b expression and mouse protein abundance data [1, 22, 16] across brain, heart, liver, lung and placenta tissue pools, whereby the mouse mRNAs were selected as homologs of the human mRNAs in the above WERI-Rb1 assay. Furthermore, the expression for the let-7b microRNA in the above tissue pools corresponds to that of mouse homolog for let-7b (see Appendix). Here we selected sequences which have associated mouse mRNA and protein measurements.

In addition to expression features, we would also like to account for other contextual sequence features, such as microRNA site accessibility. To this end, we ran the PITA [15] algorithm for computing an accessibility score for each 3'UTR sequence given the mature let-7b sequence. This score, which we will here denote as $\Delta\Delta G$, is a function of the accessibility of a target site given the most likely secondary structure of the target mRNA. Combined with the above mRNA, microRNA and protein abundance features, this yielded a total of 16 quantitative features for each sequence to be scored. Thus for this problem, each 3'UTR sequence corresponds to a putative let-7b-target interaction so that let-7b putatively targets at least one target site in the 3'UTR sequence. The above 16 features thus form the feature vector $\mathbf{x}_\alpha$ which we will use for learning to rank microRNA targets.

### 3.2.1 Incorporating diverse computational predictions

In addition to the above features, we would like to also incorporate computational target predictions for let-7b from the PicTar [17], TargetScan [7] and RNA22 [13] sequence-based target prediction methods. In order to assign scores to candidate microRNA targets, each of these methods makes use of various criteria such as conservation and contextual sequence features. The scores output by these prediction methods can be then used to generate an order graph over sequences, so that each method provides a partial ordering over some subset of microRNA-target interactions (see Appendix).

Given all of the above, we applied STORMSeq under three settings, where 1) we only used sequence data for learning to rank targets, B) we only used quantitative features (mRNA and microRNA expression, protein abundance and $\Delta\Delta G$), and C) we also used information provided by diverse computational prediction methods in addition to both sequence and quantitative features (see Appendix). To assess the out-of-sample predictive performance of our method, we selected a random sample of 250 positive sequences for our training data and the remainder for the test data. Similarly, we selected 250 sequences from the negative group for our training set

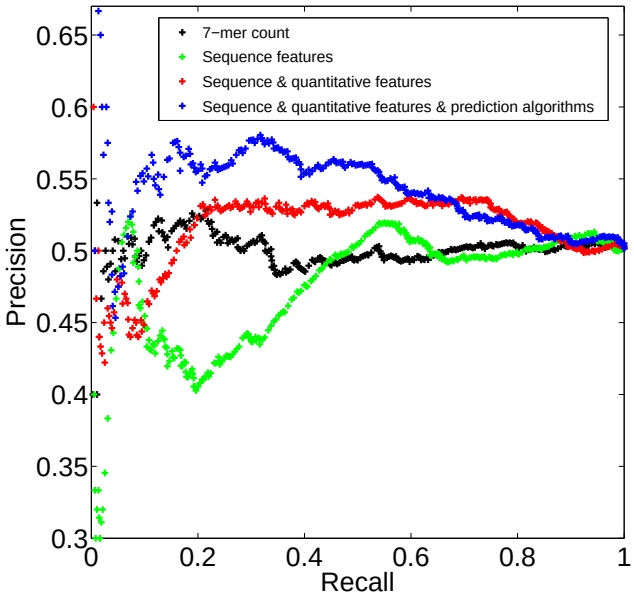
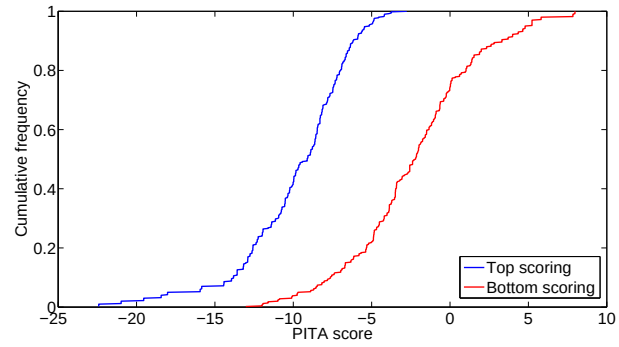


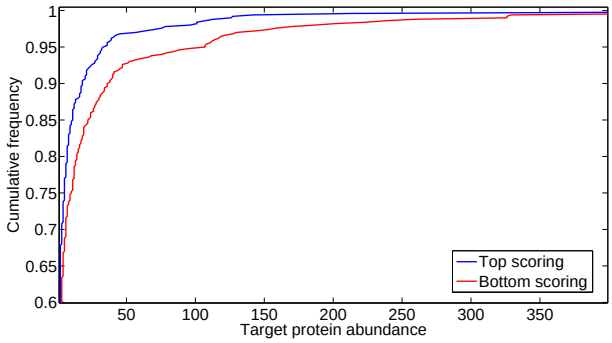

**Fig. 6** Precision versus recall for different STORMSeq learning configurations using expression data for mRNAs in response to let-7b transfection [8]. By incorporating additional sources of sequence information, sequence context and quantitative profiling features, STORMSeq achieves higher accuracy (blue) than using 7-mer counts to predict downregulation (black), using sequence data alone (green) or sequence data combined with quantitative features without computational predictions as additional data (red).

and the rest for the test data. We thus formed five independent training/test splits in this fashion (see Appendix). For each of the five train/test datasets, we computed precision and recall for each of these experimental settings. The resulting precision and recall curves, averaged over the five test sets, are shown in Figure 6. As can be seen, incorporating sequence data, quantitative features and computational predictions together under one model yields an improvement in predictive accuracy compared to using sequence alone or sequence in tandem with quantitative features. This indicates that by leveraging multiple sources of information about microRNA regulation, we can significantly increase the accuracy with which we discover microRNA targets.

For further validation, we show the cumulative distribution of $\Delta\Delta G$ scores for the top and bottom 100 targets ranked according to STORMSeq (Figure 7(a)). We expect *a priori* that sequences with lower $\Delta\Delta G$ score are more likely to be bound by a targeting microRNA than not. As can be seen, high-scoring targets have a significantly lower average $\Delta\Delta G$ value than low-scoring targets ($P < 10^{-20}$, Wilcoxon-Mann-Whitney test), demonstrating that the targets discovered by STORMSeq are likely to be genuinely targeted by let-7b. Furthermore, the protein abundances




(a)




(b)

**Fig. 7** a) Cumulative frequency plots of the $\Delta\Delta G$ scores on the top and bottom 100 targets as ranked by STORMSeq. High-scoring STORMSeq targets generally have higher target site accessibility and so have a lower $\Delta\Delta G$ value compared to low-scoring targets ($P < 10^{-20}$, Wilcoxon-Mann-Whitney); b) Cumulative frequency plots of protein abundances for top and bottom 100 targets as ranked by STORMSeq. High-scoring STORMSeq targets have significantly lower target protein abundance ($P = 7.73 \times 10^{-4}$) as a result of microRNA repressive activity.

for the top and bottom 100 targets differed significantly as well (Figure 7(b), $P = 7.73 \times 10^{-4}$), adding support for the hypothesis that the targets which receive a high score under STORMSeq are *bona fide*, as microRNA activity generally leads to lower protein abundance and mRNA transcript abundance [1, 7, 8, 15].

To assess the use of purely sequence-based methods for this problem, we also ran the MEME [2] and AlignACE [12] algorithms using default settings on the 250 positive sequences for each training set and examined the resulting PSSMs reported by both algorithms. The PSSMs obtained from these methods can then be used to rank sequences. We found that for all five training/test datasets, none of the PSSMs discovered by MEME and AlignACE led to any significant ability to rank let-7b targets (data not shown), suggesting that without additional information in the

form of sequence conservation or quantitative measurements, *de novo* approaches to scoring sequences are significantly more likely to find poor models by virtue of either using only sequence information or by virtue of model misspecification.

## 4 Discussion

We have presented the STORMSeq method for learning to rank regulatory sequences by combining heterogeneous datasets and diverse computational prediction methods. The explicit formulation of sequence search as a problem of ranking accounts for the fact that different sequences can have multiple levels of significance and any method for ranking should correctly order sequences by assigning a high score to biologically significant sequences. In particular, by accounting for the statistical dependence relationships which exist in learning to rank, STORMSeq improves predictive performance over other unstructured methods for learning to rank. In addition, STORMSeq largely avoids many of the issues of model misspecification and complex inference which may arise when modelling multiple heterogeneous datasets. As STORMSeq is formulated in fairly general terms, it can also be applied to other problems of sequence search such as ranking drug targets, discovering genetic associations or scoring protein-protein interactions, although we have not focused on such applications here.

In the case of ranking microRNA-target interactions we have shown that incorporating diverse computational predictions increases predictive accuracy as measured by precision and recall. It should be noted that one must exercise care in what additional sources of computational predictions are incorporated into the analysis. We found that by incorporating computational prediction methods which had inherently low accuracy, we could in fact decrease the predictive accuracy of our method (data not shown). In our case, particular computational prediction methods were included in our analysis on the basis of a previous study conducted in [8] which gauged the predictive accuracy of a variety of microRNA-target prediction methods according to a variety of metrics. We caution that in the case in which data is relatively limited in size, including computational predictions from methods which have low accuracy can adversely impact the accuracy of STORMSeq. A possible extension to the framework proposed here is to allow for outlier detection so that the model can discount the impact of outlier observations.

One reviewer pointed out that the optimization problem being solved is generally non-convex and may assign high probability to different orderings over sequences. Although the underlying ranking may not be unique for a given class of ranking functions and/or loss functionals, there may be a large number of —it partial orderings over sequences which are consistent with an underlying (and possibly unidentifiable) total ordering over sequences. Thus, although many orderings may be possible and STORMSeq may learn one of these, those which are most useful in practice are those orderings in which the *relevant* sequences are correctly ranked, while less of a penalty should be assigned whether we have correctly ranked the less relevant sequences. Thus the issue of whether the ranking of relevant sequences is

identifiable may be of concern, so that standard techniques for avoiding poor local minima must be used and the solutions obtained from multiple restarts should be compared with one another.

An important issue which arises often in practice concerns the tractability of the proposed framework. In a setting in which one is given a large number of sequences to be ranked for a single observation, the number of edges in an order graph may in the worst case reach $O(n^4)$, where $n$ is the number of objects in the observation. As storing and processing such a large observation may be intractable, we have made use of the mean absolute deviation (MAD) criterion for asserting preference relationships (see Appendix), which has the effect of reducing the number of pairwise preferences to be modeled. One can devise similar schemes to reduce the number of pairwise preferences to be modeled, as many of these will represent pairwise ordering constraints between very highly relevant sequences and irrelevant ones. We have also found that one can randomly break up an observation defined over many sequences into a set of multiple observations defined over smaller subsets of the sequences. Each of these observations could then be tractably modeled using the proposed method. In addition or as an alternative to the above, one can choose a CDN graph which is tractable and amenable to fast computations. An advantage of the proposed framework is that it is possible to use sparser CDN graphs which tradeoff the presence of dependencies between pairwise preferences for tractability and speedups in computation time.

We have applied STORMSeq to the problems of scoring sequences bound by transcription factors and scoring microRNA targets, whereby performing structured learning and combining different data types with computational predictions was shown to improve predictive accuracy. In the case of ranking microRNA targets, features relating to expression patterns in mouse proved to increase the ranking accuracy of scoring targets in human retinoblastomas. This suggests that STORMSeq may also be useful for problems in comparative genomics as a principled means for combining diverse datasets from different species. Other interesting extensions of the STORMSeq would include scaling the proposed framework to genome-wide detection of regulatory sequences as well as using richer representations for the ranking function which could account for direct interactions between the sequences to be ranked.

## Acknowledgements

BJF is a Fellow of the Canadian Institute for Advanced Research and holds a Canada Research Chair in Information Processing and Machine Learning. This research was partly funded by CIHR and OGI/Genome Canada.

## References


1. Babak, T., Zhang, W., Morris, Q.D., Blencowe, B.J. and Hughes, T.R. (2004) Probing microRNAs with microarrays: Tissue specificity and functional inference. RNA 10:1813–1819.
2. Bailey, T.L., Williams, N., Misleh, C. and Li, W.W. (2006) MEME: discovering and analyzing DNA and protein sequence motifs. Nucleic Acids Research 34:W369–W373.
3. Berger, M.F., Philippakis, A.A., Qureshi, A.M., He, F.S., Estep III, P.W. and Bulyk, M.L. (2006) Compact, universal DNA microarrays to comprehensively determine transcription factor binding specificities. Nature Biotechnology 24:1429–1435.
4. Chen, X., Hughes, T.R. and Morris, Q.D. (2007) RankMotif++: a motif-search algorithm that accounts for relative ranks of K-mers in binding transcription factors. Bioinformatics 23:i72–i79.
5. Chen, C.-Y., Tsai, H.-K., Hsu, C.-M., Chen, M.-J. M., Hung, H.-G., Huang, G. T.-W. and Li, W.-H. (2008) Discovering gapped binding sites of yeast transcription factors. Proceedings of the National Academy of Sciences 105:2527–2532.
6. Foat, B.C. , Morozov, A.V. and Bussemaker, H.J. (2006) Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. Bioinformatics 22:e141–e149.
7. Grimson, A., Farh, K.K.-H., Johnston, W.K., Garrett-Engele, P., Lim, L.P. and Bartel, D.P. (2007) MicroRNA targeting specificity in mammals: determinants beyond seed pairing. Molecular Cell 27:91–105.
8. Huang, J.C., Babak, T., Corson, T.W., Chua, G., Khan, S., Gallie, B. L., Hughes, T.R., Blencowe, B.J., Frey, B.J. and Morris, Q.D. (2007) Using expression profiling to identify human microRNA targets. Nature Methods 4:1045–1049.
9. Huang, J.C. and Frey, B.J. (2008) Cumulative distribution networks and the derivative-sum-product algorithm. In: Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence (UAI).
10. Huang, J.C. and Frey, B.J. (2009) Structured ranking learning using cumulative distribution networks. In: Advances in Neural Information Processing Systems (NIPS) 21.
11. Huang, J.C. (2009) Cumulative distribution networks: Inference, estimation and applications of graphical models for cumulative distribution functions. University of Toronto Ph.D. thesis.
12. Hughes, J.D., Estep III, P.W., Tavazoie, S. and Church, G.M. (2000) Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. Journal of Molecular Biology 296:1205–14.
13. Huynh, T., Miranda, K., Tay, Y., Ang, Y.-S., Tam, W.-L., Thomson, A. M., Lim, B. and Rigoutsos, I. (2006) A pattern-based method for the identification of microRNA-target sites and their corresponding RNA/RNA complexes. Cell 126:1203–1217.
14. Jarvelin, K. and Kekalainen, K. (2002) Cumulated evaluation of IR techniques. ACM Information Systems 20: 422–446.
15. Kertesz, M., Iovino, N., Unnerstall, U., Gaul, U. and Segal, E. (2007) The role of site accessibility in microRNA target recognition. Nature Genetics 39:1278–1284.
16. Kislinger, T., Cox, B., Kannan, A., Chung, C., Ignatchenko, A., Scott, M.S., Gramolini, A., Morris, Q.D., Hughes, T.R., Rossant, J. et al. (2006) Global survey of organ and organelle protein expression in mouse: combined proteomic and transcriptomic profiling. Cell 125:173–186.
17. Krek, A., Grün, D., Poy, M.N., Wolf, R., Rosenberg, L., Epstein, E.J., MacMenamin, P., da Piedade, I., Gunsalus, K.C., Stoffel, M. et al. (2005) Combinatorial microRNA target predictions. Nature Genetics 37:495–500.
18. Liu, X.S., Brutlag D.L. and Liu, J.S. (2002) An algorithm for finding proteinDNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. Nature Biotechnology 20:835–839.
19. Segal, E., Raveh-Sadka, T., Schroeder, M., Unnerstall, U. and Gaul, U. (2008) Predicting expression patterns from regulatory sequence in *Drosophila* segmentation. Nature 451:535–540.
20. Tanay, A. (2006) Extensive low-affinity transcriptional interactions in the yeast genome. Genome Research 16:962–972.

21. Wingender, E., Knüppel, R., Dietze, P. and Karas, H. (1995) TRANSFAC® database as tool for the recognition of regulatory genomic sequences. In: H.A. Lim and C.R. Cantor, (eds.) Bioinformatics & Genome Research, pp. 275-282, World Scientific Publishing Co., Inc., New Jersey.
22. Zhang, W., Morris, Q.D., Chang, R., Shai, O., Bakowski, M.A., Mitsakakis, N., Mohammad, N., Robinson, M.D., Zirngibl, R., Somogyi, E. et al. (2004) The functional landscape of mouse gene expression. Journal of Biology 3:21–43.

# Appendix

## *Cumulative distribution networks*

The CDN [9, 11] is an undirected bipartite graphical model in which the joint CDF $F(\mathbf{z})$ over a set of random variables is modeled as a product over functions defined over subsets of these variables. More formally, for variable set $\mathbf{Z}$, the joint CDF is given by

$$F(\mathbf{z}) = \prod_{s \in S} \phi_s(\mathbf{z}_s), \tag{8}$$

where $S$ is a set of function indices and for $s \in S$, $\phi_s(\mathbf{z}_s)$ is defined over some subset of the variables in $\mathbf{Z}$. For detailed derivations of the properties of CDNs, including marginal and conditional independence properties, we refer the reader to [9]. The CDN framework provides us with a means to compactly represent multivariate joint CDFs over many variables: in the next section we will formulate a loss functional for learning to rank which takes on such a form.

### A structured loss functional for learning to rank

Let the ranking function $\rho(\alpha) \equiv \rho(\alpha; \mathbf{a})$ be parameterized by the parameter vector $\mathbf{a}$ so that $r(D_n; \rho) \equiv r(D_n; \mathbf{a})$. For a given order graph $G_n$, the structured loss functional is then given by

$$\mathscr{L}(D_n; \theta) \equiv \mathscr{L}(D_n; \mathbf{a}, \nu) = -\log F_\pi\big(r(G_n; \mathbf{a})\big) = -\log \phi(\mathbf{r}(D_n; \mathbf{a})) \tag{9}$$

where $\theta = \big[\mathbf{a}\ \nu\big]$ is the set of parameters. Here we can choose from a wide variety of CDN topologies and functional forms for the CDN functions, such as the particular CDN used in [10]. We will represent the joint CDF using a single CDN function $\phi(\mathbf{r})$ set to a multivariate sigmoidal function so that

$$\phi(\mathbf{r}) = \frac{1}{1 + \sum_e \exp(-\nu r(e; \mathbf{a}, D_n))}, \quad \nu > 0. \tag{10}$$

For the given CDN and ranking functions, the learning problem for the current observation $D_n$ then becomes

$$\min_{\mathbf{a},\nu} \quad \sum_n \log\left(1+\sum_{e\in E_n}\exp\big(-\nu r(e;\mathbf{a},D_n)\big)\right) \quad \text{s.t.} \quad \nu>0. \tag{11}$$

In order to solve the above optimization problem, we will use a stochastic gradient descent algorithm which will require us to compute the gradient $\nabla_a \mathscr{L}(D_n;\theta)$ for each observation $D_n$. This is given by

$$\nabla_{\mathbf{a}}\mathscr{L}(D_n;\theta) = \nu\phi\big(\mathbf{r}(D_n;\mathbf{a})\big)\sum_{e\in E_n}\exp(-\nu r(e;\mathbf{a},D_n))\nabla_{\mathbf{a}} r(e;\mathbf{a},D_n),$$

with

$$\nabla_{\mathbf{a}} r(e;\mathbf{a},D_n) = \nabla_{\mathbf{a}}\rho(\alpha;\mathbf{a}) - \nabla_{\mathbf{a}}\rho(\beta;\mathbf{a}) \tag{12}$$

The derivative with respect to the CDN function weight $w$ is then given by

$$\partial_\nu\Big[\mathscr{L}(D_n;\theta)\Big] = -\sum_{e\in E_n} r(e;\mathbf{a},D_n)\exp\big(-\nu r(e;\mathbf{a},D_n)\big)\,\phi(\mathbf{r}) \tag{13}$$

With the above gradients, we can then proceed to construct a CDN for each observation $D_n$ and updating the parameters of the model according to the rule $\theta \leftarrow \theta - \mu\nabla_\theta\mathscr{L}(D_n;\theta)$, where $\mu$ is a learning rate parameter.

## *Ranking functions for sequence and quantitative features*

Suppose we are given a sequence $s_\alpha$ of length $L_\alpha$ which we would like to score. Let $s_\alpha^{k:k+K-1}$ be a subsequence of $s_\alpha$ of length $K$ starting at position $k$ and let $s_\alpha^j$ be the symbol observed at position $j$ in sequence $s_\alpha$. Given a PSSM $\mathbf{M}$ of length $K$ (where $M_{k,b}$ is equal to the probability of emitting symbol $b$ at position $k$ of the PSSM) we can define the score for sequence $s_\alpha$ as the probability that a transcription factor binds to at least one subsequence of length $K$ in $s_\alpha$ according to the PSSM, so that

$$\rho_{seq}(s_\alpha;\mathbf{M}) = \log\Big(1-\prod_{k=0}^{L_\alpha-K}(1-P(s_\alpha^{k+1:k+K}|\mathbf{M}))\Big) \tag{14}$$

where $P(s_\alpha^{k+1:k+K}|\mathbf{M}) = \prod_{j=k+1}^{k+K} M_{j,s_\alpha^j}$ is the probability of binding to subsequence $s_\alpha^{k+1:k+K}$ according to $\mathbf{M}$. The derivative of the ranking function $\rho_{seq}(s_\alpha;\mathbf{M})$ with respect to the parameter $M_{k,b}$ is equal to

$$\frac{\partial\rho_{seq}(s_\alpha;\mathbf{M})}{\partial M_{k,b}} = \frac{1-\exp\big(\rho_{seq}(s_\alpha;\mathbf{M})\big)}{\exp\big(\rho_{seq}(s_\alpha;\mathbf{M})\big)}\left(\sum_i \frac{P(s_\alpha^{i+1:i+K}|\mathbf{M})}{1-P(s_\alpha^{i+1:i+K}|\mathbf{M})}\big([s_\alpha^{i+k}=b]-P(b|\mathbf{M})\big)\right) \tag{15}$$

We can then collect these derivatives into a vector to form the gradient $\nabla_{\mathbf{M}}\rho_{seq}(s_\alpha;\mathbf{M})$.

In the case where we are provided with quantitative features in the form of a feature vector $\mathbf{x}_\alpha$, we can define the ranking function $\rho_{quant}(\mathbf{x}_\alpha;\mathbf{w})$ to be a linear function given by $\nabla_{\mathbf{w}}\rho_{quant}(\mathbf{x}_\alpha;\mathbf{w}) = \mathbf{x}_\alpha$. Once we have computed both gradients, we can evaluate

$$\nabla_{\mathbf{a}}\rho(\alpha;\mathbf{a}) = \begin{bmatrix} \nabla_{\mathbf{M}}\rho_{seq}(s_\alpha;\mathbf{M}) \\ \nabla_{\mathbf{w}}\rho_{quant}(\mathbf{x}_\alpha;\mathbf{w}) \end{bmatrix}. \tag{16}$$

## *Ranking functions for discovering gapped motifs*

In the case in which we wish to allow for gaps, we can posit a ranking function of the same form as in Equation (14), but with an additional constraint that for degenerate positions $j$ in the PSSM $\mathbf{M}$, we have $M_{j,a} = 0.25$ for $a \in \{A,C,G,T\}$. This constraint is equivalent to forcing certain positions to be contribute the same score to the total sequence score regardless of what nucleotides occur at these positions. Alternatively, we could regularize each degenerate position of the PSSM by adding some constant $C_j$ to each entry $M_{j,a}$, where $C_j$ is chosen so that for position $j$ is a distribution that is close to being uniform. For the former constraint, we would simply update the entries of the PSSM for only non-degenerate positions. For the latter constraint, we can regularize the appropriate entries of $\mathbf{M}$ during the learning process by simply adding $C_j$ after each update of the PSSM. An example of the PSSM for such a gapped motif is shown in Figure 8. It is worth noting that the length of the gap, or number of degenerate positions in the PWM, can either be specified by the user or it can be selected via cross-validation, as with the length of the PWM.


**Fig. 8** An example of a gapped motif.

## *The RankMotif++ method as a disconnected CDN*

For the RankMotif++ model of [4], the corresponding probability over all pairwise preferences $\alpha \succ \beta$ is modeled by a product over logistic functions of $\rho(\alpha) - \rho(\beta)$ so that $F_\pi(\mathbf{r}(D_n)) \equiv \mathbb{P}[\pi \leq \mathbf{r}(D_n)] = \prod_s \frac{1}{1+\exp(-\nu r_s)} = \prod_{\alpha\succ\beta} \frac{1}{1+\exp\big(-\nu\big(\rho(\alpha)-\rho(\beta)\big)\big)}$ with $\rho(\alpha)$ corresponding to the sequence ranking function $\rho_{seq}$ above. This can thus

be represented as a completely disconnected CDN where each function node corresponds to $\phi_s(r_s) = \frac{1}{1+\exp(-\nu r_s)}$ and all pairwise object preferences are modeled as being independent of one another. This is illustrated in Figure 9 for an example with four sequences $s_\alpha, s_\beta, s_\gamma, s_\delta$ to be ranked in which we represent the corresponding joint CDF using two different CDNs.

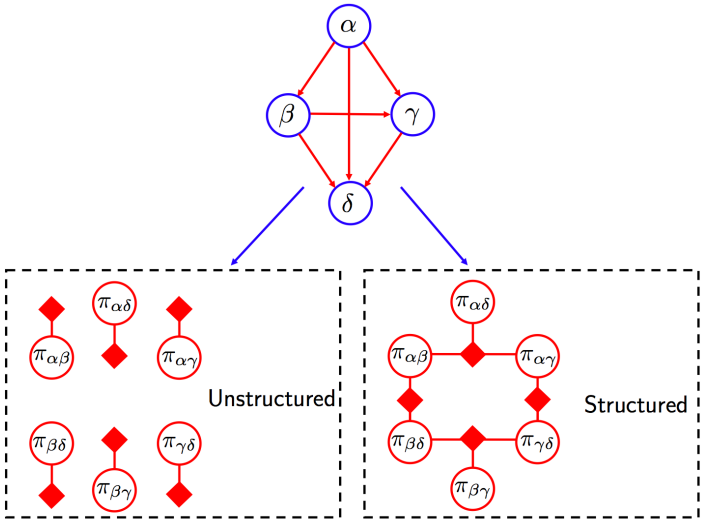



**Fig. 9** An example of an order graph over four nodes $\alpha, \beta, \gamma, \delta$ corresponding to the ordering $\alpha \succ \beta \succ \gamma \succ \delta$, with CDNs representing two different loss functions corresponding to different independence assumptions about pairwise preferences. Whereas the RankMotif++ method of [4] corresponds to an unstructured learning method which assumes independence of preference variables, STORMSeq models the dependencies between preferences by introducing connections between preference variables in the corresponding CDN.

## *Settings for STORMSeq*

We ran STORMSeq for 100 epochs, or passes through the training observations, using a stochastic gradients optimization method. The learning rate was set to $\mu = 0.1$ with a decay rate of $1/t$ at the end of each epoch $t$. In order to provide regularization on the CDN width parameter $\nu$, we set a constraint $\nu \leq 1$. In the case where we learn a PSSM $\mathbf{M}$, we enforce the constraints that $M_{k,b} > 0 \; \forall \; k, b$ and $\sum_b M_{k,b} = 1 \; \forall k = 1, \cdots, K$. In the case where we learn weights $\mathbf{w}$, we set an additional $L_1$-norm constraint of $\|\mathbf{w}\|_1 \leq 50$. All computational runs were performed in triplicate and the best optimum achieved on training data was selected for evaluation

on test data using criteria described in the sections below. Additional details on the learning method are provided in [10].

## *Methods for ranking sequences bound by transcription factors*

Data was downloaded from the Supplementary Material section of [3], which consisted of measured intensities $y_\alpha$ for a set of sequences $\{s_\alpha \in \mathscr{S}\}$. The dataset contained five experiments across two microarrays (*Array 1* and *Array 2*) profiling the binding of the transcription factors Cbf1, Ceh-22, Oct-1, Rap1, Zif268. We used the array labeled *Array 1* as the source of our training data, and the probe sequences from *Array 2* as the source of our test data. We normalized the microarray intensity data in both sets by first shifting microarray intensities such that the minimum intensity was equal to one, then applying a log-transformation, as in [4]. We labelled the 250 probe sequences which had the highest measured intensity as positives and the 250 sequences with the lowest normalized intensities as negatives. We then constructed the order graph over these 500 sequences based on preferences assessed using the criteria used by [4] where we compute the median absolute deviation $m$ of the 500 normalized intensities and asserted $\alpha \succ \beta$ if $y_\alpha > y_\beta + 3\sigma$ and at least one of $s_\alpha, s_\beta$ were labelled as positive sequences as described above, where $\sigma = m/0.6745$, where 0.6745 is the median absolute deviation of the standard normal.

Using the above sequence ranking function $\rho(s_\alpha; \mathbf{M})$ for a given PSSM length $K$, we ran STORMSeq and RankMotif++ using three random initializations each, whereby we selected the model which maximized the Spearman correlation with the training data, as per [4]. For each initialization, the PSSM $\mathbf{M}$ was initialized to a set of random positive values and then normalized so that $\sum_b M_{k,b} = 1 \; \forall \; k = 1, \cdots, K$. The MatrixREDUCE, MDScan and Prego methods were run on the training data as specified in [4], and the resulting PSSM models were selected using the same Spearman correlation metric as above. For all models, we varied $K$ from 7 to 13 and selected the value of $K$ which optimized the above Spearman correlation criteria.

## *Methods for ranking microRNA targets*

We focused on the human genes in the let-7b transfection experiment which A) had 3'UTR sequence data provided by Ensembl and B) were provided with both mRNA expression and protein abundance data in 3,636 paired mRNA-protein expression profiles obtained from cDNA microarray and mass-spectrometry across brain, heart, liver, lung and placenta tissue pools in mouse [16, 22]. This yielded a total of 799 human 3'UTR sequences to be scored. We then selected the 400 sequences with the lowest log-expression ratios as positives and labelled the other 399 genes as negatives. To assess the out-of-sample predictive performance of our method, we selected a random sample of 250 positive sequences for our training data and the

remainder for the test data. Similarly, we selected 250 sequences from the negative group for our training set and the rest for the test data. We thus formed five independent training/test splits in this fashion. Preferences were then assessed as described above for the PBM data. Once we obtained the training and test datasets, we ran STORMSeq with $K = 7$ on each of the training datasets and selected the best model out of 3 random restarts via the Spearman correlation between the learned ranking function scores and the rankings seen in the training data.

In conjunction with the above data, we used the expression let-7b across brain, heart, liver, lung and placenta tissue pools [1] with the mRNA/protein profiles mentioned above. Additionally, the $\Delta\Delta G$ accessibility score was computed by the PITA algorithm [15] using the mRNA sequences for each of the mouse mRNAs in the data from [22] and using the mature mouse let-7b sequence for the default algorithm settings provided in [15].

We downloaded microRNA target predictions for the let-7b microRNA from the Supplementary Data resources for the TargetScan [7], PicTar [17] and RNA22 [13] algorithms. The set of TargetScan predictions contains both conserved and non-conserved targets and the set of RNA22 targets contains both target predicted from 5'UTR and 3'UTR sequences. We mapped all predictions to the above mouse mRNA and microRNA labels. Pairwise preference relationships were established for a given 3'UTR sequence by summing over microRNA target site scores within the given 3'UTR sequence and sorting scores. For a given prediction method, the preference $\alpha \succ \beta$ was established between two 3'UTR's $s_\alpha, s_\beta$ if $s_\alpha$ had a higher score than $s_\beta$ and at least one of $s_\alpha, s_\beta$ were labelled as positive sequences as described above.

## *Assessing ranking performance*

To assess predictive performance of any given ranking method, we scored each node $\alpha$ using the ranking function $\rho(\alpha)$ learned by the method. Given the ordering obtained from $\rho$ and given positive/negative labels for the nodes being ranked, we can then compute Precision and Recall as

$$Precision = \frac{TP}{TP+FP}$$
$$Recall = \frac{TP}{TP+FN}$$

where $TP, FP, FN$ correspond to the number of true positives, false positives and false negatives respectively.

We also used the Normalized Discounted Cumulative Gains [14] metric, which is commonly in use in information retrieval research. The NDCG accounts for the fact that highly relevant sequences should be ranked higher by a given method, so more weight should be placed on correctly ranking highly relevant sequences than marginally relevant ones. The formula for computing the NDCG for truncation level

$n$, or the number of top-ranking sequences, is

$$NDCG(n) = Z_n \sum_{j=1}^{n} \frac{2^{r(j)} - 1}{c_j} \tag{17}$$

where $r(j)$ is an observed label indicating the level of importance of the sequence (e.g.: amount of downregulation from a microRNA) and $Z_n$ is a constant to ensure that $NDCG(n) = 1$ for the perfect ranking, so that higher NDCG indicates increased ability to predict the ordering of sequences. The weights $\{c_1, \cdots, c_n\}$ are an increasing sequence of real-valued positive numbers which allow us to penalize errors made in the top of the ranked list whilst discounting errors made for less relevant sequences. Here we chose $c_j = \log_2(1+j) \; \forall j = 1, \cdots, n$. The advantage of the NDCG metric is that it does not assume that sequences are to be classified as positive or negative and it accounts for both multiple label values and the fact that highly important sequences should be ranked first. This contrasts with the use of Area Under the ROC Curve, or AUC, which weighs misranking errors equally regardless of where they occur in a ranked list. The NDCG can be also seen as an approximation to the cost of experimentally validating or analyzing sequences at the top of the list which are not biologically relevant.

In the case of where we are scoring sequences bound by transcription factors, we set the labels $r(j)$ to be the normalized array intensities, shifted to be non-negative and scaled to obtain a maximum label of 1. For the purpose of evaluating on let-7b targets, we set the above relevance labels to be the negative log-expression-ratios of each putative target, shifted to be non-negative and scaled to obtain a maximum label of 1.