RESEARCH

Sort-Seq Tools: sequence-function relationship modeling for massively parallel assays

William T. Ireland[1] and Justin B. Kinney[2*]

Abstract

A variety of massively parallel assays for measuring high-resolution sequence-function relationships have been developed in recent years. However, software for learning quantitative models from these data is lacking. Here we describe Sort-Seq Tools, a software package that allows multiple types of quantitative models to be fit to massively parallel data in multiple different ways. We demonstrate Sort-Seq Tools on both simulated and published data from Sort-Seq studies, massively parallel reporter assays, and deep mutational scanning experiments. We observe that, as an inference method, information maximization generally outperforms both least squares optimization and enrichment ratio calculations.

Background

High throughput DNA sequencing technologies are being used to do far more than just sequence genomes [1]. One area of research that is rapidly expanding thanks to new sequencing technologies is the study of quantitative sequence-function relationships. In recent years, a variety of massively parallel assays capable of providing high-resolution measurements of sequence-function relationships have been described. These include Sort-Seq experiments on bacterial and yeast promoters [2, 3, 4], massively parallel reporter assays (MPRAs) of mammalian enhancers [5, 6], and deep mutational scanning (DMS) experiments on proteins [7, 8]. Fig. 1 provides an illustration of these three different assays.

Many massively parallel experiments, including those of [2, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18], share the common form illustrated in Fig. 2A.[1] One begins with a specific "wild type" sequence of interest. A library comprising variants of this wild type sequence (i.e., that have scattered substitution mutations) is then generated. These library sequences are used as input to an experimental procedure that measures a specific sequence-dependent activity and, as a result of this measurement, outputs sequences into one or more "bins." Finally, the number of occurrences of each variant sequence in each bin is assayed using high-throughput sequencing.

When analyzing data from massively parallel experiments, one often wishes to obtain activity measurements for individual sequences. Multiple software packages that address this need have been described [28, 29, 30]. In this paper we focus on a different task: how to use massively parallel data to infer quantitative models of sequence-function relationships. Specifically, given data of the form shown in Fig. 2B, we wish to learn the values of parameters in a mathematical model that describes the realtionship between sequence and activity. Such quantitative modeling is often motivated by the desire to predict the activities of sequences that have not been assayed in the experiment. Modeling can also provide a way to characterize biophysical mechanisms, e.g., measure *in vivo* protein-DNA and protein-protein interaction energies [2].

Multiple studies have fit quantitative models to massively parallel data (e.g., [2, 5, 7]), but in almost all cases this has been done using custom scripts. To our knowledge, the only published software package that provides such quantitative modeling capabilities of the type we seek is DMS Tools [30]. This package, however, facilitates only the simplest type of modeling: the inference of matrix models using enrichment ratios. This limitation places severe constraints on the

[*]Correspondence: jkinney@cshl.edu
[2]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, 11375, Cold Spring Harbor, NY
Full list of author information is available at the end of the article

[1] Not all massively parallel experiments have this form. SELEX-SEQ and related experiments often use library DNA that is completely random (e.g., [19, 20, 21, 22, 23, 24]), while some Sort-Seq and MPRA efforts have used libraries that contain specified arrangements of binding sites or large numbers of different genomic regions (e.g., [3, 25, 26, 27]).

types of questions that one might hope to answer using massively parallel data.

Here we introduce Sort-Seq Tools, a software package that enables sophisticated quantitative modeling of sequence-function relationships using data from a variety of massively parallel assays. Sort-Seq Tools provides simple command-line methods that enable the inference of multiple types of models (i.e., matrix models and neighbor models) using multiple inference methods (enrichment ratio calculations, least squares optimization, and mutual information maximization). Here we demonstrate these modeling capabilities on simulated data and on data from three prior studies: the Sort-Seq experiments of [2], the MPRA experiments of [5], and the DMS experiments of [7]. Importantly, we find that model inference using mutual information maximization almost always outperforms model inference using enrichment ratio calculations or least squares optimization.

Sort-Seq Tools also provides methods for simulating massively parallel data using a user-specified model, for computing useful summary statistics, and for evaluating quantitative models on arbitrary input sequences. These ancillary capabilities are elaborated in Supplemental Information (SI). All of this functionality is accessible via the command-line. Sort-Seq Tools is available on PyPI and can be installed with the command "`pip install sortseq_tools`". Source code and documentation is available on GitHub at `jbkinney/sortseq_tools`, and a snapshot of the software used to perform the calculations and to generate the figures featured in this paper is provided at [URL TO BE INCLUDED].

Methods

We formalize the problem of inferring quantitative models of sequence-function relationships as follows. We represent massively parallel data as a set of N sequence-measurement pairs, $\{S^n, M^n\}_{n=1}^N$, where each measurement M^n is a non-negative integer corresponding to the bin in which the n'th sequence sequence, S^n, was found. We assume that all sequences S have the same length L, and that at each of the L positions in each sequence there is one of C possible characters ($C = 4$ for DNA and RNA; $C = 20$ for protein). In what follows, each sequence S is represented as a binary $C \times L$ matrix having elements

$$S_{cl} = \left\{ \begin{array}{ll} 1 & \text{if character } c \text{ occurs at position } l \\ 0 & \text{otherwise} \end{array} \right. . \tag{1}$$

Here, $l = 1, 2, \ldots, L$ indexes positions within the sequence, while $c = 1, 2, \ldots, C$ indexes possible nucleotides or amino acids. Note that, in this representation, the same sequence S will typically be observed multiple times in each data set and will often fall into multiple different bins.

Our goal is to derive a function that can, given a sequence S, predict the activity R measured by the experiment. To do this we assume that the activity value R is given by a function $r(S, \theta)$ that depends on the sequence S and a set of parameters θ. Before we can infer the values of the parameters θ from data, we must first answer two distinct questions:

1 What functional form do we choose for $r(S, \theta)$?
2 How, specifically, do we use the data $\{S^n, M^n\}$ and the model predictions $r(S, \theta)$ to infer parameter values?

Sort-Seq Tools provides two different options for the function r: a "matrix" model, where each position in S contributes independently to the predicted activity, and a "neighbor" model, which accounts for potential epistatic interactions between neighboring positions. Sort-Seq Tools also provides three different methods for fitting the parameters θ to data: parameters values can be inferred by computing enrichment ratios (a method applicable only to matrix models), by performing least squares optimization, or by maximizing the mutual information between model predictions and measurements. These different model types and inference methods are elaborated below.

Matrix models and neighbor models

Matrix models have the form

$$r_{\text{mat}}(S, \theta) = \sum_{c=1}^{C} \sum_{l=1}^{L} \theta_{cl} S_{cl}. \tag{2}$$

In this context, θ is a $C \times L$ matrix where each element θ_{cl} represents the contribution of character c at position l to the overall sequence-dependent activity. For example, Fig. 3B shows the parameters of a matrix model that describes the sequence specificity of *Escherichia coli* RNA polymerase (RNAP). These parameters were inferred from the Sort-Seq data of [2] using Sort-Seq Tools.

Neighbor models have the form

$$r_{\text{nbr}}(S, \theta) = \sum_{c=1}^{C} \sum_{d=1}^{C} \sum_{l=1}^{L-1} \theta_{cdl} S_{cl} S_{d(l+1)}. \tag{3}$$

Such models comprise $C^2(L-1)$ parameters, denoted θ_{cdl}, that represent the contributions of all possible adjacent di-nucleotides or di-amino-acids within S. Fig. 3C illustrates one such model which, as above, describes RNAP and was inferred using Sort-Seq Tools operating on data from [2].

Both matrix and neighbor models, as written above, suffer from the presence of "gauge freedoms" – directions in parameter space that do not affect model predictions. These degrees of freedom must be eliminated if one wishes to interpret the values of individual parameters. Sort-Seq Tools accomplishes this by restricting θ to lie within the subspace of sequence variation. Moreover, the parameter values returned by Sort-Seq Tools are normalized so that the mean value of model predictions over all possible sequences is zero, and the variance in these values in equal to unity.

Enrichment ratio inference

The computation of enrichment ratios is the simplest way to infer quantitative sequence-function relationships from massively parallel data. The motivation for this inference method traces back to the seminal work of Berg and von Hippel [31, 32], and the resulting models can be thought of as the incarnation of position weight matrices [33] in the context of massively parallel experiments. This inference method is the one supported by dms_tools [30], and the calculation of such models is one of the primary types of analyses reported in the DMS literature [8].

Enrichment ratio inference, however, places strong restrictions on the types of models and data that one can use. Specifically, one is restricted to using matrix models only, and the data used to compute parameter values can consist of only two bins: a library bin ($M = 0$) and a selected bin ($M = 1$). Moreover, the validity of this inference procedure depends on assumptions that are often not satisfied by real-world experiments [34].

If these restrictions are met and one is willing to make the necessary assumptions, then model parameters θ^{ER} are computed using

$$\theta^{\mathrm{ER}}_{cl} = \log \frac{f^1_{cl}}{f^0_{cl}}, \tag{4}$$

where

$$f^M_{cl} = \frac{1}{N_M + C\lambda}\left(\sum_{n|M} S^n_{cl} + \lambda\right). \tag{5}$$

Here, f^M_{cl} denotes the fraction of sequences in bin M having character c at position l, N_M is the total number of sequences in bin M, λ is a nonnegative pseudocount (specified by the user), and the sum in Eq. 5 is restricted to sequences S^n that lie within bin M (i.e., for which $M^n = M$). Because enrichment ratio inference reduces to a simple counting problem, Sort-Seq Tools is able to perform this computation very rapidly.

Least squares inference

Least squares provides a computationally simple inference procedure that overcomes the most onerous restrictions of enrichment ratio calculations. It can be used to infer any type of linear model, including both matrix models and neighbor models. It can also be used on data that consists of more than two bins.

The idea behind the least squares approach is to choose parameters θ^{LS} that minimize a quadratic loss function. Specifically, we use

$$\theta^{\mathrm{LS}} = \mathrm{argmin}_\theta L(\theta), \tag{6}$$

where

$$L(\theta) = \sum_M \sum_{n|M} \frac{\left[r(S^n,\theta) - \mu_M\right]^2}{\sigma^2_M} + \alpha \sum_i \theta^2_i. \tag{7}$$

Here, μ_M is the presumed mean activity of sequences in bin M, σ^2_M is the presumed variance in the activities of such sequences, i indexes all parameters in the model, and α is a "ridge regression" regularization parameter [35]. By using the objective function $L(\theta)$, one can rapidly compute values of the optimal parameters θ using standard algorithms [36].

One downside to least squares inference is the need to assume specific values for μ_M and for σ^2_M for each bin M. Sort-Seq Tools allows the user to manually specify these values. There is a danger here, since assuming incorrect values for μ_M and σ^2_M will generally lead to bias in the inferred parameters θ^{LS} [37]. In practice, however, the default choice of $\mu_M = M$ and $\sigma^2_M = 1$ often works surprisingly well when bins are arranged from lowest to highest activity.

Another downside to least squares is the need to assume that experimental noise – specifically, $p(R|M)$ – is Gaussian. Only in such cases does least squares inference correspond to a meaningful maximum likelihood calculation. In massively parallel assays, however, noise is often strongly non-Gaussian. In such situations, least squares inference cannot be expected to yield correct model parameters for any choice of μ_M and σ^2_M [37].

Information maximization inference

An alternative inference procedure, one that does not suffer from the need to assume a specific form for experimental noise, is the maximization of mutual information. In the large data limit, information maximization is equivalent to performing maximum likelihood inference when the quantiative form of experimental noise is unknown [37, 38, 39]. This approach was originally proposed for receptive field inference in sensory neuroscience [40, 41, 42], but has since been applied in multiple molecular biology contexts [38, 43], including in the analysis of massively parallel experiments [2, 5].

In this approach, parameter values are chosen to maximize the mutual information between model predictions and measurements. Specifically, one chooses

$$\theta^{\mathrm{IM}} = \mathrm{argmax}_{\theta} I(\theta), \tag{8}$$

where

$$\begin{aligned} I(\theta) &= I[R; M] & (9) \\ &= \sum_{M} \int dR\, p(R, M) \log \frac{p(R, M)}{p(R)p(M)} & (10) \end{aligned}$$

is the mutual information between the bins M in which sequences are found and the corresponding model prediction R for those sequences. In what follows, $I(\theta)$ is referred to as the "predictive information" of the model. For each choice of θ, computing predictive information requires a regularized estimate of the joint probability distribution $p(R, M)$. Sort-Seq Tools currently uses standard kernel density methods [35] to estimate these distributions, although field-theoretic density estimation [44, 45] may ultimately prove superior in this context.

Following [2], Sort-Seq Tools identifies information-maximizing parameters using a Metropolis Monte Carlo algorithm in which each choice for θ has relative probability $\exp[NI(\theta)]$. Because this Monte Carlo procedure is computationally expensive, information maximization is much slower than enrichment ratio calculations or least squares inference. Running on a standard laptop computer, our current algorithm takes between 30 minutes and 2 hours for each of the information maximization tasks described below.

Results

To test the capabilities of Sort-Seq Tools, we analyzed data from previously published Sort-Seq [2], MPRA [5], and DMS [7] studies. Each of these studies generated multiple independent data sets, allowing us to test the inference capabilities of Sort-Seq Tools by training and testing models on separate data. We also analyzed simulated data in order to assess the ability of Sort-Seq Tools to accurately recover known parameter values.

Sort-Seq data

In their studies of the *E. coli lac* promoter, Kinney et al. [2] performed six independent Sort-Seq experiments, which they referred to as rnap-wt, crp-wt, full-wt, full-500, full-150, and full-0. All of these experiments assayed the transcriptional activity of variant sequences spanning a 75 bp region of the *lac* promoter (Fig. 3A). This assayed region is known to bind two proteins, RNAP and CRP, at two separate binding sites. In the original study [2], models for the sequence specificity of these two proteins were inferred by modeling how transcription depends on sequence variation within these two different binding sites.

For both RNAP and CRP, we used each of these six data sets to infer both matrix models and neighbor models.[2] Inference was performed using each of the three methods supported by Sort-Seq Tools: enrichment ratios (ER), least squares (LS), and information maximization (IM). The performance of each of these models on each of the available data sets was then quantified by estimating the predictive information $I[R; M]$.

Fig. 4A illustrates the performance of each inferred RNAP model (columns) on each of the published data sets (rows). Fig. 4B shows similar results for the inferred CRP models.[3] For both CRP and RNAP, the IM-inferred matrix models consistently outperformed the LS- and ER-inferred matrix models when evaluated on independent test data (Figs. 4C,4D,4E,4F). This finding lends support to the theory-based arguments of [39, 37] that information maximization has substantial advantages over other methods for inferring quantitative sequence-function relationships from massively parallel data.[4]

We also investigated whether neighbor models, which account for epistatic interactions between neighboring positions in a sequence, might provide better descriptions of RNAP and CRP than simple matrix models do. To our knowledge, the presence of such interactions in either of these well-studied proteins has yet to be definitively established (although see [46]). We therefore compared the predictive performance of

[2] Raw data from [2] is available on NCBI SRA, accession number SRA012345; processed data formatted for use with Sort-Seq Tools is provided on GitHub at jbkinney/sortseq_tools.

[3] RNAP models were not trained or tested on the crp-wt data set because the RNAP binding site was not mutagenized in that experiment. Similarly, CRP models were not trained or tested on the rnap-wt data set. CRP models were also not trained or tested on the full-0 data set because cAMP, a ligand that CRP requires in order to bind DNA, was absent in this experiment.

[4] We note that the ER matrix models computed by Sort-Seq Tools are essentially indistinguishable from the ER matrix models computed by dms_tools. This shows that the favorable performance of IM-based inference is not an artifact of how ER-based inference is implemented within Sort-Seq Tools. See SI for a direct comparision of the ER-based inference methods of Sort-Seq Tools and dms_tools.

matrix and neighbor models that were trained (using IM) on the same data sets (Figs. 4G, 4H).

Neighbor models did not always outperform matrix models in these tests. However, for both CRP and RNAP, neighbor models did perform better than their corresponding matrix models when the predictive information of the matrix model was high (Figs. 4E,4F). Such high matrix model predictive information values are expected to occur when the data used to train models is of high quality. We interpret this finding as evidence for epistatic interactions in the specificities of both CRP and RNAP. Indeed we expected *a priori* that crp-wt data would be the best data set for training models of CRP because the mutation rate used in this experiment was the highest (24%). This expectation is consistent with our finding that the CRP neighbor model inferred from crp-wt outperformed every other matrix model of CRP.

Simulated data

To further establish the ability of Sort-Seq Tools to properly infer quantitative models, we next analyzed simulated Sort-Seq data. Specifically, to generate simulated Sort-Seq data, we used the simulation capabilities of Sort-Seq Tools together with the nbr-IM models for RNAP and CRP inferred from the full-wt dataset of [2]. Eight data sets were simulated in total, four for RNAP and four for CRP. In each simulation, 10^6 cells were sorted into either 10 or 2 bins; see SI for simulation details. Half of these simulated data sets (labeled "train") were then used to infer matrix and neighbor models as described in the previous section. The other half (labeled "test") were used solely to evaluate model performance.

Fig. 5A shows results for the simulated RNAP data, while Fig. 5B shows corresponding results for simulated CRP data. The nbr-IM models performed best in every case tested, with virtually no apparent difference in performance between training and test data. In particular, all of the nbr-IM models performed substantially better than the mat-IM models, demonstrating the ability of Sort-Seq tools to learn correct epistatic interactions. Figs. 5C and 5D plot the values of parameters for inferred neighbor models against the corresponding parameter values of the neighbor models used to generate the data. We found very strong agreement, with a signal-to-noise ratio of 31 across the 528 parameters of the RNAP neighbor model, and a signal-to-noise ratio of 49 across the 336 parameters of the CRP neighbor model.

MPRA and DMS data

Sort-Seq tools is designed to facilitate the quantitative modeling of data from a variety of massively parallel assays, including MPRA and DMS experiments. To test the utility of Sort-Seq Tools in these contexts, we inferred matrix models using MPRA data from [5] and DMS data from [7].[5]

In [5], replicate MPRA experiments were performed on a synthetic cAMP responsive element (CRE). These experiments tested $\sim 2.7 \times 10^4$ microarray-synthesized CREs having randomly scattered substitution mutations (10% per nucleotide position) throughout an 87 bp region. Using Sort-Seq Tools, we inferred matrix models spanning this entire 87 bp region using IM, LS, and ER-based inference. We found that the IM-inferred models performed the best in cross-comparisons (Fig. 6A). Moreover, both of these IM-inferred models performed better on both data sets relative to the matrix model described in the original publication [5].

The DMS experiments of [7] assayed a variable region spanning 33 aa within a WW domain protein. Specifically, the gene sequences of this WW domain was mutagenized at ~2% per base. Multiple rounds of panning using a peptide ligand were then used to select WW-domain variants displayed on the surface of phage. The WW domain coding sequences present in the phage library after 0, 3, and 6 rounds of selection were then sequenced.

Using Sort-Seq Tools, we fit models to either the round 0 and round 3 libraries, or to the round 3 and round 6 libraries. When trained on round 0,3 data and tested on round 3,6 data, IM-inferred matrix models performed better than LS-inferred models and about the same as ER-inferred models (Fig. 6B). However, IM-inferred models fit to round 3,6 data actually performed worse than the corresponding ER-inferred models. This is the only situation we encountered where ER models outperformed IM models.

The poor performance of IM in this context is most likely due to the sparsity of data in the round 3,6 dataset. Specifically, in the round 3,6 dataset, we observed 8 amino-acid-position combinations with no representation in the data. Furthmore, 16 amino-acid-position combinations were represented by data from

[5] The preprocessed MPRA data of [5] was obtained from NCBI GEO, accession number GSE31982. The preprocessed DMS data of [8] was kindly provided by Douglas Fowler; raw data is available from NCBI SRA, accession number SRA020603. Processed data from both publications, formatted for use with Sort-Seq Tools, is provided on GitHub at `jbkinney/sortseq_tools`. The neighbor models fit to data from both of these studies performed poorly relative to matrix models. We therefore ignore these neighbor models in what follows.

only one sequence. By contrast, the round 0,3 dataset contained data on all amino-acid-position combinations, and for only 2 of these combinations did this data come from a single sequence. Our results therefore suggest that IM-based inference can perform at least as well on DMS data as ER-based inference, but only when datasets are sufficiently rich. More generally, the existence of 20 amino acids compared to 4 DNA/RNA bases places a significantly larger burden on the amount of data needed to obtain accurate models from DMS data relative to Sort-Seq or MPRA data. This is true regardless of the inference method one uses.

Discussion

Sort-Seq Tools provides routines for inferring quantitative models of sequence-function relationships from massively parallel data. Unlike existing bioinformatic software, Sort-Seq Tools allows the user to fit multiple types of models using multiple different inference methods. It also provides routines for simulating data, computing summary statistics, assessing model performance, and evaluating models on arbitrary input sequences. These capabilities fill a major gap in the current bioinformatics software repertoire.

Applying Sort-Seq Tools to previously published data sets, we observed that matrix models inferred using mutual information maximization consistently performed as well or better than matrix models inferred using either enrichment ratios or least squares optimization. The only exception to this finding occured in a situation where the training data covered less sequence space than the test data. Our findings thus validate previous theoretical work [37, 38, 39] arguing that the noisiness of massively parallel experiments makes information-based inference ideal in the large data regime. Sort-Seq Tools is the first software package to enable such information-based inference on massively parallel data.[6]

We also demonstrated the ability of Sort-Seq Tools to accurately learn neighbor models, which account for epistatic interactions between neighboring positions within a sequence. Not surprisingly, the accurate inference of neighbor models requires higher quality data than does the accurate inference of matrix models, and not all of the data sets analyzed here met this criterion. Still, when analyzing the data of [2] we did observe epistatic interactions in the sequence specificities of CRP and RNAP, a finding that was missed in the original publication.

There is still much to do to facilitate the quantitative modeling of sequence-function relationships. Inference with Sort-Seq Tools is currently limited to matrix models and neighbor models, yet there are a variety of other types of models that are likely to prove useful. Of particular interest are models with sparse all-versus-all pairwise interactions [46], models with interactions based on higher-order sequence features [47], deep neural network models [48], and nonlinear biophysics-based models [2]. Sort-Seq Tools provides a framework into which such modeling capabilities can be incorporated in the future, and through which the results of different modeling strategies can be compared in a transparent way.

Availability of data and materials

- Project name: Sort-Seq Tools
- Project homepage: github.com/jbkinney/sortseq_tools
- Archived version: [TO BE INCLUDED]
- Operating systems: Platform independent
- Programming language: Python, Cython
- Other requirements: POSIX command line, Python 2.7.9, multiple Python packages.
- License: New BSD
- Restrictions on use by non-academics: None.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

WTI and JBK participated in all aspects of this research.

Acknowledgements

We thank Douglas Fowler for providing preprocessed DMS data from [7]. We also thank Tarjei Mikkelsen for posting the preprocessed MPRA data from [5] on NCBI GEO. WTI was supported by NIH grants DP1 OD000217 and R01 GM085286. The work of JBK was supported by the Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory.

Author details

[1]Department of Physics, California Institute of Technology, 91125, Pasadena, CA. [2]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, 11375, Cold Spring Harbor, NY.

References

1. Shendure, J., Lieberman Aiden, E.: The expanding scope of DNA sequencing. Nat Biotechnol **30**(11), 1084–1094 (2012)
2. Kinney, J.B., Murugan, A., Callan, C.G., Cox, E.C.: Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence. Proc Natl Acad Sci USA **107**(20), 9158–9163 (2010)
3. Sharon, E., Kalma, Y., Sharp, A., Raveh-Sadka, T., Levo, M., Zeevi, D., Keren, L., Yakhini, Z., Weinberger, A., Segal, E.: Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. Nat Biotechnol **30**(6), 521–530 (2012)
4. Peterman, N., Levine, E.: Sort-seq under the hood: implications of design choices on large-scale characterization of sequence-function relations. BMC Genomics **17**(1), 206 (2016)
5. Melnikov, A., Murugan, A., Zhang, X., Tesileanu, T., Wang, L., Rogov, P., Feizi, S., Gnirke, A., Callan, C.G., Kinney, J.B., Kellis, M., Lander, E.S., Mikkelsen, T.S.: Systematic dissection and optimization of inducible enhancers in human cells using a massively parallel reporter assay. Nat Biotechnol **30**(3), 271–277 (2012)

[6] The software package FIRE [43] enables information-based inference of short motifs within long regulatory sequences. However, the severe length limitation that FIRE places on motifs (which must be $\lesssim 10$ bp) makes this software inapplicable to most massively parallel datasets.

6. White, M.A.: Understanding how cis-regulatory function is encoded in DNA sequence using massively parallel reporter assays and designed sequences. Genomics **106**(3), 165–170 (2015)
7. Fowler, D.M., Araya, C.L., Fleishman, S.J., Kellogg, E.H., Stephany, J.J., Baker, D., Fields, S.: High-resolution mapping of protein sequence-function relationships. Nat Methods **7**(9), 741–746 (2010)
8. Fowler, D.M., Fields, S.: Deep mutational scanning: a new style of protein science. Nat Methods **11**(8), 801–807 (2014)
9. Patwardhan, R.P., Lee, C., Litvin, O., Young, D.L., Pe'er, D., Shendure, J.: High-resolution analysis of DNA regulatory elements by synthetic saturation mutagenesis. Nat Biotechnol **27**(12), 1173–1175 (2009)
10. Patwardhan, R.P., Hiatt, J.B., Witten, D.M., Kim, M.J., Smith, R.P., May, D., Lee, C., Andrie, J.M., Lee, S.-I., Cooper, G.M., Ahituv, N., Pennacchio, L.A., Shendure, J.: Massively parallel functional dissection of mammalian enhancers in vivo. Nat Biotechnol **30**(3), 265–270 (2012)
11. Kwasnieski, J.C., Mogno, I., Myers, C.A., Corbo, J.C., Cohen, B.A.: Complex effects of nucleotide variants in a mammalian cis-regulatory element. Proc Natl Acad Sci USA **109**(47), 19498–19503 (2012)
12. Hietpas, R.T., Jensen, J.D., Bolon, D.N.A.: Experimental illumination of a fitness landscape. Proc Natl Acad Sci USA **108**(19), 7896–7901 (2011)
13. Adkar, B.V., Tripathi, A., Sahoo, A., Bajaj, K., Goswami, D., Chakrabarti, P., Swarnkar, M.K., Gokhale, R.S., Varadarajan, R.: Protein model discrimination using mutational sensitivity derived from deep sequencing. Structure **20**(2), 371–381 (2012)
14. Whitehead, T.A., Chevalier, A., Song, Y., Dreyfus, C., Fleishman, S.J., De Mattos, C., Myers, C.A., Kamisetty, H., Blair, P., Wilson, I.A., Baker, D.: Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. Nat Biotechnol **30**(6), 543–548 (2012)
15. Schlinkmann, K.M., Honegger, A., Türeci, E., Robison, K.E., Lipovšek, D., Plückthun, A.: Critical features for biosynthesis, stability, and functionality of a G protein-coupled receptor uncovered by all-versus-all mutations. Proc Natl Acad Sci USA **109**(25), 9810–9815 (2012)
16. Holmqvist, E., Reimegård, J., Wagner, E.G.H.: Massive functional mapping of a 5'-UTR by saturation mutagenesis, phenotypic sorting and deep sequencing. Nucl Acids Res **41**(12), 122 (2013)
17. Peterman, N., Lavi-Itzkovitz, A., Levine, E.: Large-scale mapping of sequence-function relations in small regulatory RNAs reveals plasticity and modularity. Nucl Acids Res **42**(19), 12177–12188 (2014)
18. Liachko, I., Youngblood, R.A., Keich, U., Dunham, M.J.: High-resolution mapping, characterization, and optimization of autonomously replicating sequences in yeast. Genome Res **23**(4), 698–704 (2013)
19. Zykovich, A., Korf, I., Segal, D.J.: Bind-n-Seq: high-throughput analysis of in vitro protein-DNA interactions using massively parallel sequencing. Nucl Acids Res **37**(22), 151–151 (2009)
20. Zhao, Y., Granas, D., Stormo, G.D.: Inferring binding energies from selected binding sites. PLoS Comput Biol **5**(12), 1000590 (2009)
21. Jolma, A., Kivioja, T., Toivonen, J., Cheng, L., Wei, G., Enge, M., Taipale, M., Vaquerizas, J.M., Yan, J., Sillanpää, M.J., Bonke, M., Palin, K., Talukder, S., Hughes, T.R., Luscombe, N.M., Ukkonen, E., Taipale, J.: Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. Genome Res **20**(6), 861–873 (2010)
22. Wong, D., Teixeira, A., Oikonomopoulos, S., Humburg, P., Lone, I.N., Saliba, D., Siggers, T., Bulyk, M., Angelov, D., Dimitrov, S., Udalova, I.A., Ragoussis, J.: Extensive characterization of NF-κB binding uncovers non-canonical motifs and advances the interpretation of genetic functional traits. Genome Biol **12**(7), 70 (2011)
23. Slattery, M., Riley, T., Liu, P., Abe, N., Gomez-Alcala, P., Dror, I., Zhou, T., Rohs, R., Honig, B., Bussemaker, H.J., Mann, R.S.: Cofactor binding evokes latent differences in DNA binding specificity between Hox proteins. Cell **147**(6), 1270–1282 (2011)
24. Jolma, A., Yan, J., Whitington, T., Toivonen, J., Nitta, K.R., Rastas, P., Morgunova, E., Enge, M., Taipale, M., Wei, G., Palin, K., Vaquerizas, J.M., Vincentelli, R., Luscombe, N.M., Hughes, T.R., Lemaire, P., Ukkonen, E., Kivioja, T., Taipale, J.: DNA-binding specificities of human transcription factors. Cell **152**(1-2), 327–339 (2013)
25. Smith, R.P., Taher, L., Patwardhan, R.P., Kim, M.J., Inoue, F., Shendure, J., Ovcharenko, I., Ahituv, N.: Massively parallel decoding of mammalian regulatory sequences supports a flexible organizational model. Nat Genet **45**(9), 1021–1028 (2013)
26. Kheradpour, P., Ernst, J., Melnikov, A., Rogov, P., Wang, L., Zhang, X., Alston, J., Mikkelsen, T.S., Kellis, M.: Systematic dissection of regulatory motifs in 2000 predicted human enhancers using a massively parallel reporter assay. Genome Res **23**(5), 800–811 (2013)
27. Mogno, I., Kwasnieski, J.C., Cohen, B.A.: Massively parallel synthetic promoter assays reveal the in vivo effects of binding site variants. Genome Res **23**(11), 1908–1915 (2013)
28. Fowler, D.M., Araya, C.L., Gerard, W., Fields, S.: Enrich: software for analysis of protein function by enrichment and depletion of variants. Bioinformatics **27**(24), 3430–3431 (2011)
29. Alam, K.K., Chang, J.L., Burke, D.H.: FASTAptamer: A Bioinformatic Toolkit for High-throughput Sequence Analysis of Combinatorial Selections. Mol Ther Nucleic Acids **4**(3), 230 (2015)
30. Bloom, J.D.: Software for the analysis and visualization of deep mutational scanning data. BMC Bioinformatics **16**, 168 (2015)
31. Berg, O., von Hippel, P.: Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. J Mol Biol **193**(4), 723–750 (1987)
32. Berg, O., von Hippel, P.: Selection of DNA binding sites by regulatory proteins. II. The binding specificity of cyclic AMP receptor protein to recognition sites. J Mol Biol **200**(4), 709–723 (1988)
33. Stormo, G., Fields, D.: Specificity, free energy and information content in protein-DNA interactions. Trends Biochem Sci **23**(3), 109–113 (1998)
34. Mustonen, V., Kinney, J.B., Callan, C.G., Lässig, M.: Energy-dependent fitness: a quantitative model for the evolution of yeast transcription factor binding sites. Proc Natl Acad Sci USA **105**(34), 12376–12381 (2008)
35. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2nd edn. Springer, ??? (2011)
36. Press, W., Teukolsky, S., Wetterling, W., Flannery, B.: Numerical Recipes in C : the Art of Scientific Computing, (1997)
37. Atwal, G.S., Kinney, J.B.: Learning Quantitative Sequence–Function Relationships from Massively Parallel Experiments. J Stat Phys **162**(5), 1203–1243 (2016)
38. Kinney, J.B., Tkacik, G., Callan, C.G.: Precise physical models of protein-DNA interaction from high-throughput data. Proc Natl Acad Sci USA **104**(2), 501–506 (2007)
39. Kinney, J.B., Atwal, G.S.: Parametric inference in the large data limit using maximally informative models. Neural Comput **26**(4), 637–653 (2014)
40. Sharpee, T., Rust, N., Bialek, W.: Analyzing neural responses to natural signals: maximally informative dimensions. Neural Comput **16**(2), 223–250 (2004)
41. Paninski, L.: Convergence properties of three spike-triggered analysis techniques. Network-Comp Neural **14**(3), 437–464 (2003)
42. Sharpee, T., Sugihara, H., Kurgansky, A., Rebrik, S., Stryker, M., Miller, K.: Adaptive filtering enhances information transmission in visual cortex. Nature **439**(7079), 936–942 (2006)
43. Elemento, O., Slonim, N., Tavazoie, S.: A universal framework for regulatory element discovery across all genomes and data types. Mol Cell **28**(2), 337–350 (2007)
44. Kinney, J.B.: Estimation of probability densities using scale-free field theories. Phys Rev E, 011301 (2014)
45. Kinney, J.B.: Unification of field theory and maximum entropy methods for learning probability densities. Phys Rev E **92**(3-1), 032107 (2015)
46. Otwinowski, J., Nemenman, I.: Genotype to phenotype mapping and the fitness landscape of the E. coli lac promoter. PLoS ONE **8**(5), 61570 (2013)
47. Sharon, E., Lubliner, S., Segal, E.: A feature-based approach to modeling protein-DNA interactions. PLoS Comput Biol **4**(8), 1000154 (2008)
48. Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J.: Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol **33**(8), 831–838 (2015)

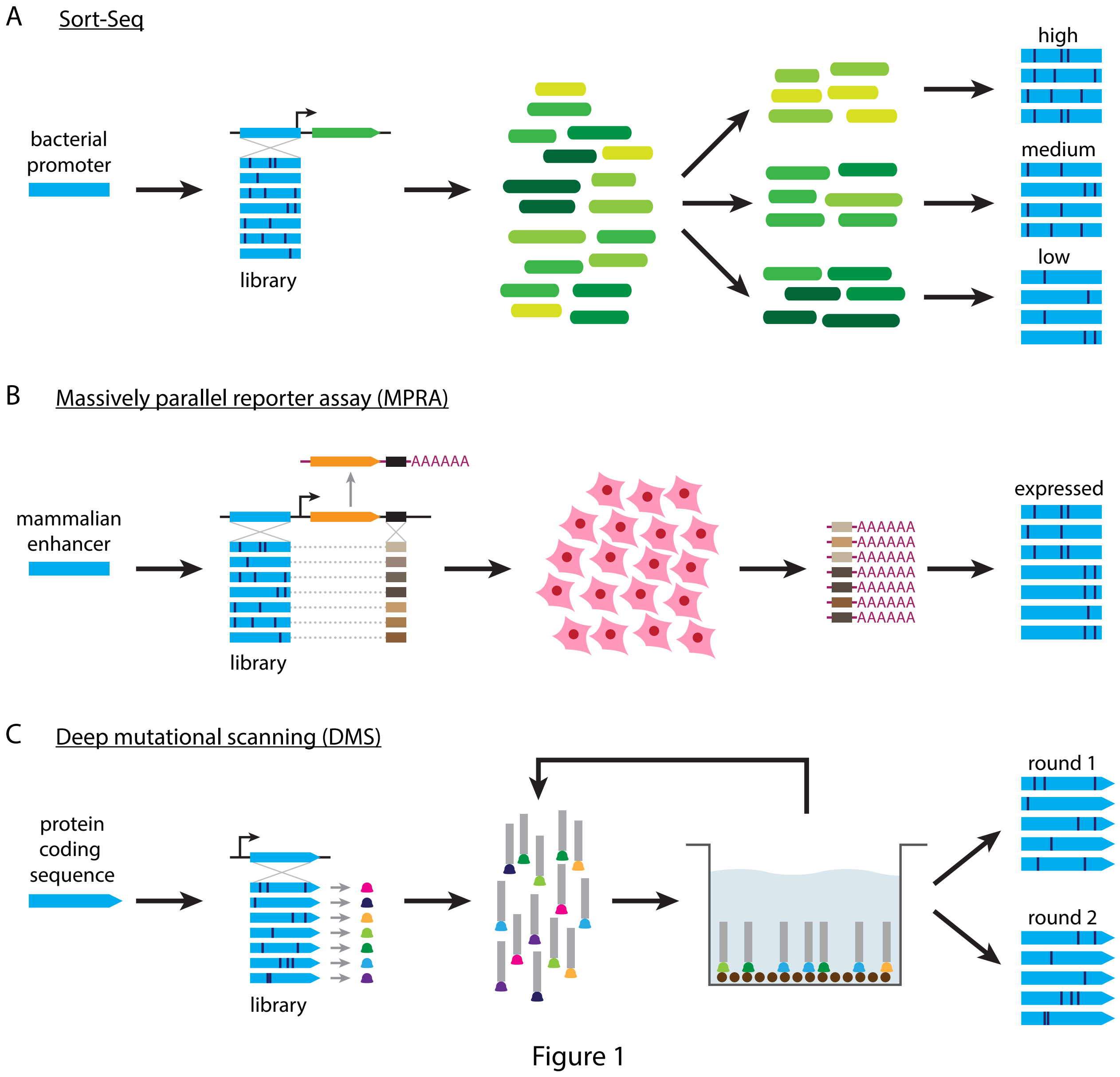
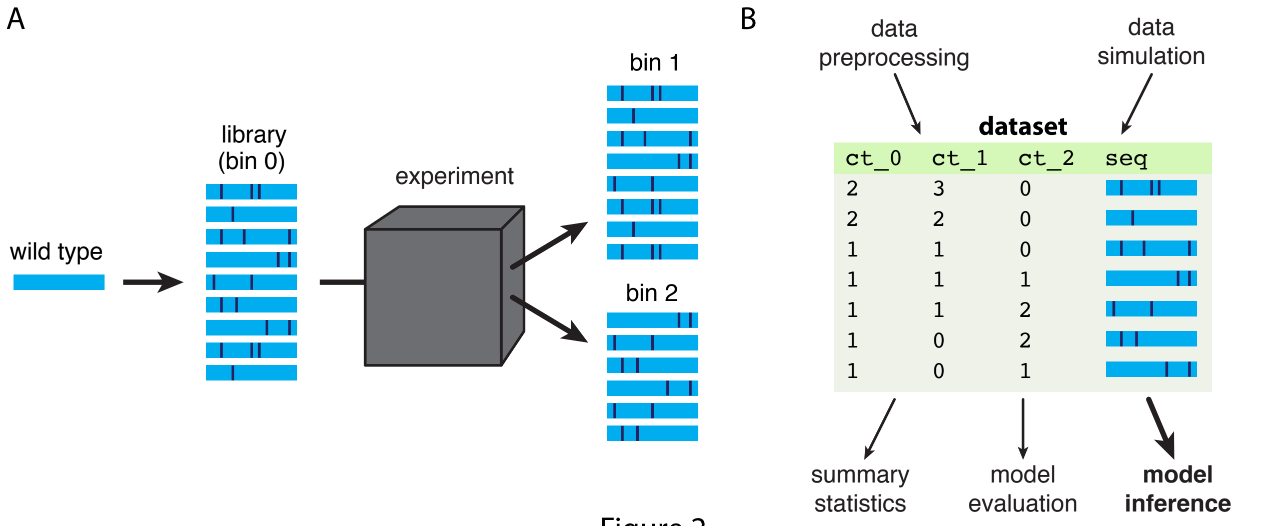


Figure 1 Three different massively parallel experiments. (A) The Sort-Seq assay of [2]. A plasmid library is generated in which mutagenized versions of a bacterial promoter (blue) drive the expression of a fluorescent protein (green). Cells carrying these plasmids are then sorted according to measured fluorescence using fluorescence-activated cell sorting (FACS). The variant promoters in each bin of sorted cells are then sequenced. (B) The MPRA assay of [5]. Variant enhancers (blue) are used to drive the transcription of RNA that contains enhancer-specific tags (shades of brown). Expression constructs are transfected into cell culture, after which tag-containing RNA is isolated and sequenced. Output sequences consist of the variant enhancers that correspond to expressed tags. (C) The DMS assay of [7]. Randomly mutagenized gene sequences (blue) produce variant proteins (colored bells) that are expressed on the surface of phage (gray rectangles). Panning is used to enrich for phage that express proteins that bind a specific ligand of interest (brown circles). The variant coding regions enriched after one or more rounds of panning are then sequenced.



ct_0	ct_1	ct_2	seq
2	3	0	
2	2	0	
1	1	0	
1	1	1	
1	1	2	
1	0	2	
1	0	1	

Figure 2

Figure 2 Overview of Sort-Seq Tools. (A) In all massively parallel assays, a library of sequences is used as input to an experiment (black box) that outputs these sequences into one or more bins. The prevalence of each sequence in each bin depends on the assayed activity of that sequence. Sort-Seq Tools can be used to analyze data from such experiments when the input library consists of substitution-mutated versions of a specific "wild type" sequence. (B) The data from such experiments can be represented as a table listing the number of occurrences of each unique sequence in each bin. Sort-Seq Tools provides routines for inferring quantitative models from data sets that have this form. Routines are also provided for data preprocessing, for simulating data, for computing summary statistics, and for evaluating models on arbitrary sequences.

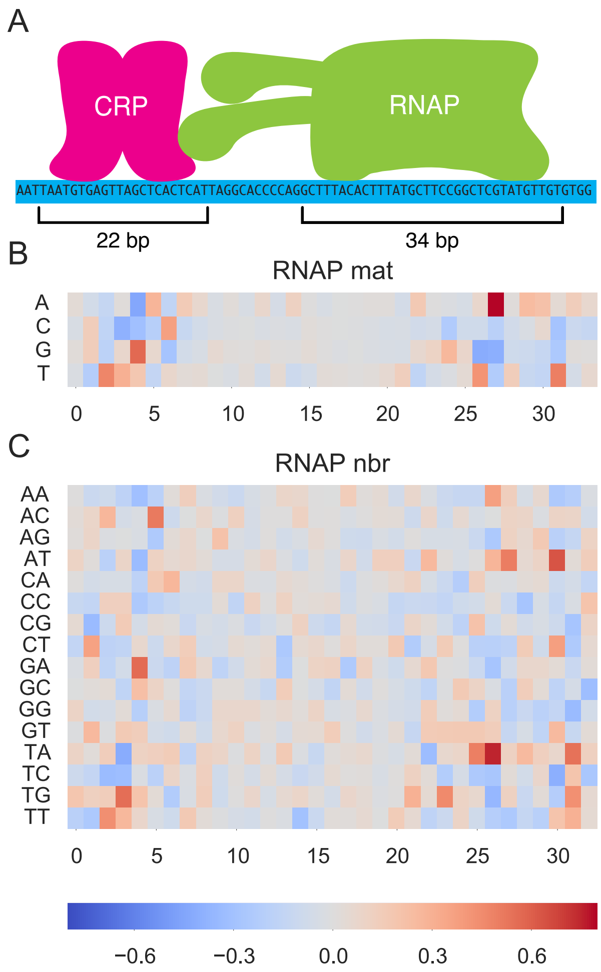


Figure 3

Figure 3 Examples of quantitative models. (A) A 75 bp region of the *E. coli lac* promoter, containing binding sites for CRP and RNAP, was assayed in the Sort-Seq experiments of [2]. Multiple types of quantitative models for both CRP and RNAP (spanning the two indicated regions) were inferred from the multiple data sets of [2] using multiple different inference methods. (B) A matrix model for RNAP, inferred from the full-wt experiment of [2] via information maximization. (C) A neighbor model for RNAP spanning the same region and fit to the same data as in panel B, again inferred using information maximization. The parameters shown in panels (B) and (C) are centered and normalized as described in the text.

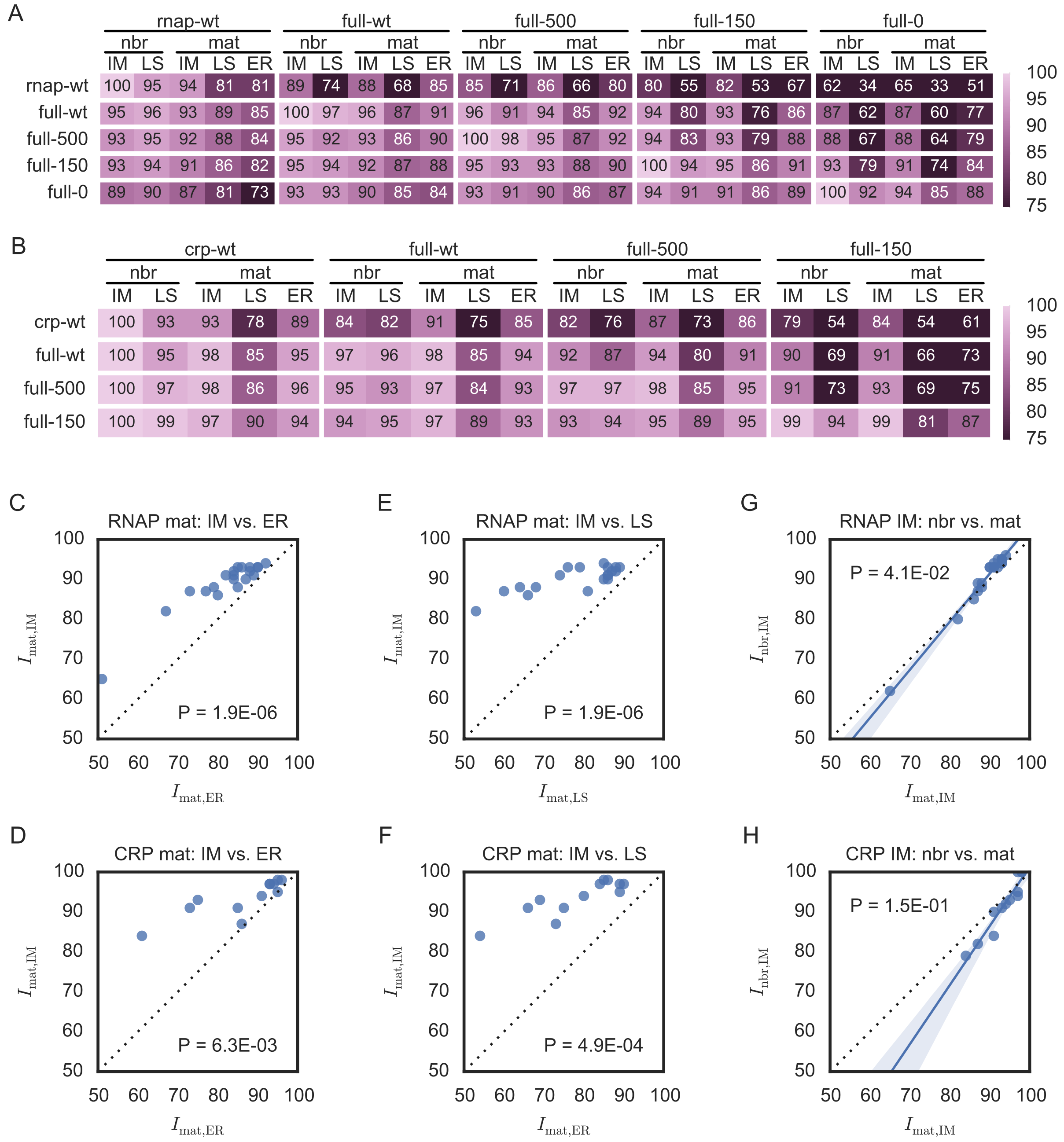


Figure 4

Figure 4 Analysis of Sort-Seq data. (A,B) Performance of (A) RNAP and (B) CRP models inferred from and evaluated on Sort-Seq data from [2]. Each column corresponds to an inferred model; column headers indicate the data set (rnap-wt, crp-wt, full-wt, full-500, full-150, or full-0) used to train the model, the type of model inferred (neighbor (nbr) or matrix (mat)), and the inference method used (information maximization (IM), least squares (LS), or enrichment ratios (ER)). Rows indicate the data sets used to evaluate model performance. Heatmap values give the predictive information of each inferred model (column) on each test set (row). These values are expressed as a percentage of the maximal predictive information achieved on each test set (i.e., along each row). (C-H) Scatter plot comparisons of predictive information values for (C,D) matrix models fit using IM ($I_{\mathrm{mat,IM}}$) vs. using ER ($I_{\mathrm{mat,ER}}$), (E,F) matrix models fit using IM vs. using LS ($I_{\mathrm{mat,LS}}$), and (G,H) IM-inferred matrix models versus IM-inferred neighbor models ($I_{\mathrm{nbr,IM}}$). Data points in panels C-H indicate model performance on non-training data only. In panels G and H, regression lines and 95% bootstrap confidence intervals are shown.

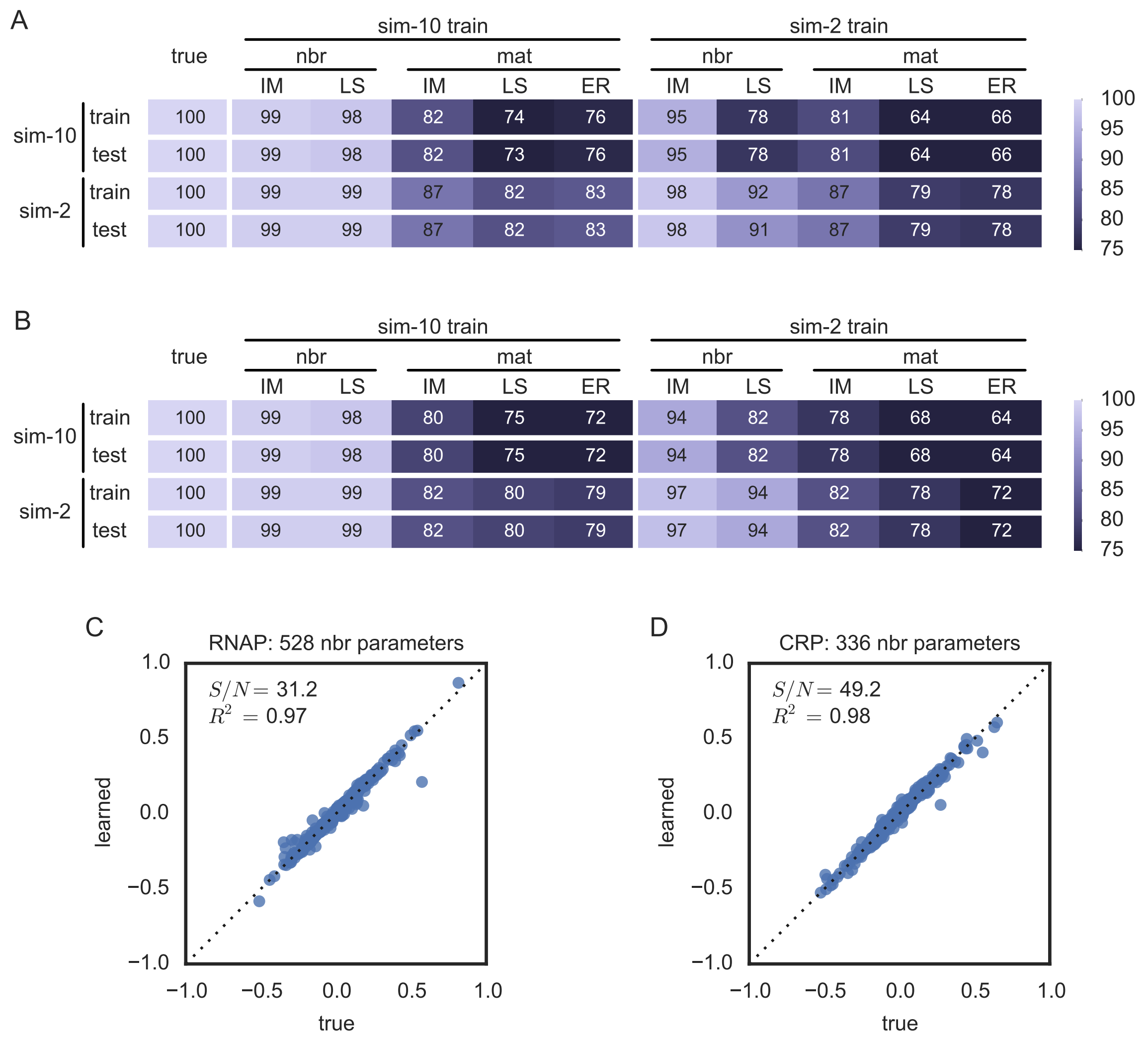
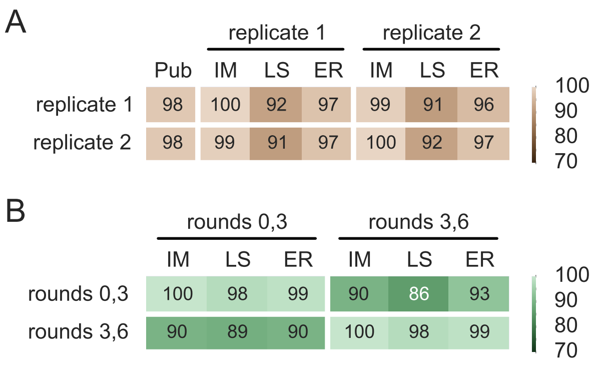


Figure 5

Figure 5 Analysis of simulated data. Sort-Seq data was simulated using the RNAP and CRP neighbor models inferred via information maximization from the full-wt data of [2]. Four data sets were generated for each model: one training and one test set were generated by sorting into 10 bins, while one training and one test set generated by sorting into 2 bins. (A,B) Performance of (A) RNAP and (B) CRP models inferred from and evaluated on these simulated data sets. Columns indicate the data set used to train the model, the type of model inferred (nbr or mat), and the inference method used for training (IM, LS, or ER). Rows indicate data used to evaluate model performance. As in Figs. 3A and 3B, heatmaps show predictive information values expressed as a percentage of the maximal predictive information achieved on each data set. (C,D) Comparison of the parameters of the neighbor models used in these simulations to the parameters of the neighbor models fit to the corresponding "sim-10 train" data via information maximization. Also shown is the signal-to-noise ratio, defined as the variance in the abcissa divided by the variance in the deivation of the ordinate from the diagonal.



		replicate 1			replicate 2		
	Pub	IM	LS	ER	IM	LS	ER
replicate 1	98	100	92	97	99	91	96
replicate 2	98	99	91	97	100	92	97

	rounds 0,3			rounds 3,6		
	IM	LS	ER	IM	LS	ER
rounds 0,3	100	98	99	90	86	93
rounds 3,6	90	89	90	100	98	99

Figure 6

Figure 6 Analysis of MPRA and DMS data. (A) Cross-comparison of matrix models fit to data from two replicate MPRA experiments reported in [5]. The performance of the matrix model reported in the original publication (Pub) is also shown. (B) In the DMS experiments of [7], sequence data was gathered after 0, 3, and 6 rounds of selection. Shown is a cross-comparison of matrix models fit to data from either rounds 0 and 3, or to data from rounds 3 and 6.

Additional Files

Additional file 1 — Sample additional file title
Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title
Additional file descriptions text.