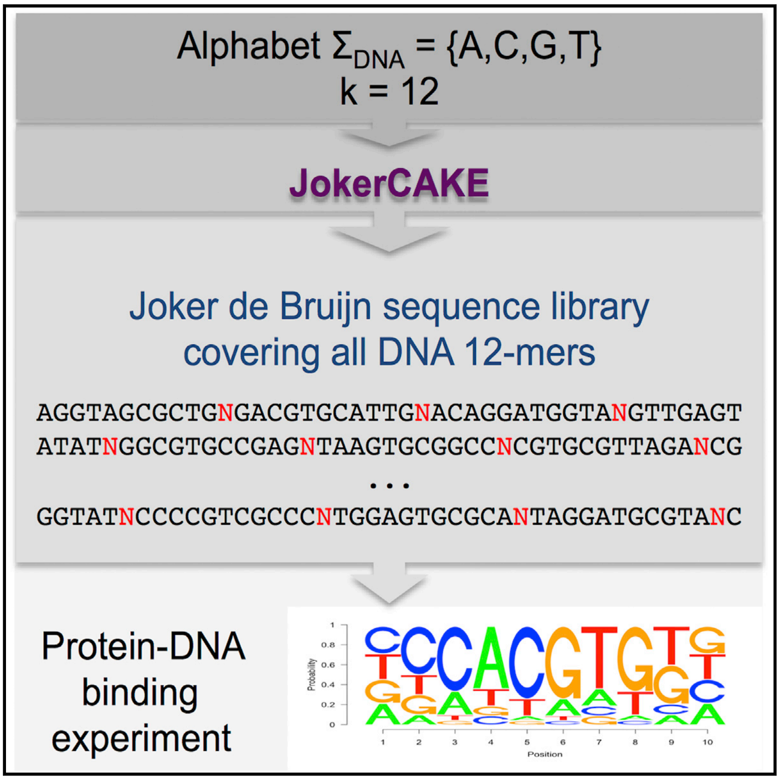
Cell Systems

Report

Optimized Sequence Library Design for Efficient *In Vitro* Interaction Mapping

Graphical Abstract



Alphabet $\Sigma_{DNA} = \{A,C,G,T\}$
k = 12

JokerCAKE

Joker de Bruijn sequence library covering all DNA 12-mers

```
AGGTAGCGCTGNGACGTGCATTGNACAGGATGGTANGTTGAGT
ATATNGGCGTGCCGAGNTAAGTGCGGCCNCGTGCGTTAGANCG
...
GGTATNCCCCGTCGCCCNTGGAGTGCGCANTAGGATGCGTANC
```

Protein-DNA binding experiment

Authors

Yaron Orenstein, Robert Puccinelli,
Ryan Kim, Polly Fordyce,
Bonnie Berger

Correspondence

bab@mit.edu

In Brief

We present a new compact sequence design that covers all k-mers utilizing joker characters and develop an efficient algorithm to generate such designs. We show through simulations and experimental validation that these sequence designs are useful for identifying high-affinity binding sites at significantly reduced cost and space.

Highlights

- A new sequence design that covers all possible k-mers by using joker characters
- We developed an algorithm to generate such designs given an alphabet and k
- Results demonstrate the ability to search a larger sequence space at reduced cost
- Experimental validation proves the ability to identify high-affinity binding sites

Orenstein et al., 2017, Cell Systems *5*, 230–236
September 27, 2017 © 2017 The Authors. Published by Elsevier Inc.
http://dx.doi.org/10.1016/j.cels.2017.07.006

OPEN
ACCESS
CellPress

Cell Systems
Focus on RECOMB
Report

Optimized Sequence Library Design for Efficient *In Vitro* Interaction Mapping

Yaron Orenstein,[1] Robert Puccinelli,[2] Ryan Kim,[3] Polly Fordyce,[2,4,5,6] and Bonnie Berger[1,7,8,*]
[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[2]Department of Genetics, Stanford University, Stanford, CA 94305, USA
[3]Research Science Institute, Center for Excellence in Education, McLean, VA 22102, USA
[4]Department of Bioengineering, Stanford University, Stanford, CA 94305, USA
[5]ChEM-H Institute, Stanford University, Stanford, CA 94305, USA
[6]Chan Zuckerberg Biohub, San Francisco, CA 94158, USA
[7]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[8]Lead Contact
*Correspondence: bab@mit.edu
http://dx.doi.org/10.1016/j.cels.2017.07.006

SUMMARY

Sequence libraries that cover all k-mers enable universal, unbiased measurements of binding to both oligonucleotides and peptides. While the number of k-mers grows exponentially in k, space on all experimental platforms is limited. Here, we shrink k-mer library sizes by using joker characters, which represent all characters in the alphabet simultaneously. We present the JokerCAKE (joker covering all k-mers) algorithm for generating a short sequence such that each k-mer appears at least p times with at most one joker character per k-mer. By running our algorithm on a range of parameters and alphabets, we show that JokerCAKE produces near-optimal sequences. Moreover, through comparison with data from hundreds of DNA-protein binding experiments and with new experimental results for both standard and JokerCAKE libraries, we establish that accurate binding scores can be inferred for high-affinity k-mers using JokerCAKE libraries. JokerCAKE libraries allow researchers to search a significantly larger sequence space using the same number of experimental measurements and at the same cost.

INTRODUCTION

Protein-DNA, -RNA, and -peptide interactions drive many cellular processes. High-throughput experimental data describing the strength and specificity of individual protein interactions through universal, unbiased libraries provide critical information for predicting targets *in vivo* and reconstructing interaction networks. These experiments typically attempt to directly measure protein binding to sequence libraries that cover all possible DNA, RNA, or amino acid k-mers. Universal, or complete, coverage guarantees that specificities can be identified *de novo* for any protein, without any prior knowledge of its preferences or the conditions under which it is active. Microarrays that cover all k-mers have been used successfully in various biotechnologies to measure protein-DNA, -RNA, and -peptide binding (Berger et al., 2006; Fordyce et al., 2010; Gurard-Levin et al., 2010; O'Donoghue et al., 2012; Ray et al., 2009; Smith et al., 2013).

While these technologies have been used successfully to measure protein interactions, they all face a similar challenge: the space on the experimental device and the sequence length that can be used are both limited, restricting the total sequence space that can be probed in a single experiment. In particular, increasing k poses difficulties since the number of sequences needed to cover all k-mers increases exponentially with k, as the number of k-mers over alphabet $\sum$ is $|\sum|^k$. Several algorithmic solutions have been proposed to generate sequence libraries that cover all possible k-mers in the most compact space possible. A de Bruijn sequence is the shortest sequence in which each k-mer appears exactly p times, with the total sequence length given by $|\sum|^k p + k - 1$. De Bruijn sequences and variants of them have been the basis of several microarray designs (O'Donoghue et al., 2012; Orenstein and Berger, 2016 Orenstein and Shamir, 2013; Philippakis et al., 2008; Ray et al., 2013; Smith et al., 2013). The shared limitation of all of these designs is that all k-mers must occur in the initial unbiased sequence set, thus their total length is at least the number of k-mers $|\sum|^k$.

Here, we generate smaller libraries that cover all k-mers by using joker characters, thereby maximizing the ability to probe sequence preferences within a constrained experimental space. Joker characters represent degenerate nucleotides (or amino acids) that cover all characters in the alphabet (e.g., joker character N within an oligonucleotide represents {A,C,G,T}). Oligonucleotides containing such degenerate nucleotides (or amino acids) can be ordered directly from the vendor at no extra cost. When degenerate characters are specified within an oligonucleotide sequence, vendors simply substitute near-equimolar mixtures of nucleotides (adjusted to compensate for small differences in coupling efficiencies) in place of a single nucleotide species during the coupling reactions. This substitution thereby produces a pool of oligonucleotides, with approximately 25% containing each of A, C, G, and T at that position. Thus far, however, they have been excluded from unbiased library designs. The use of joker characters has the potential to introduce degeneracy, which lowers the statistical robustness of the

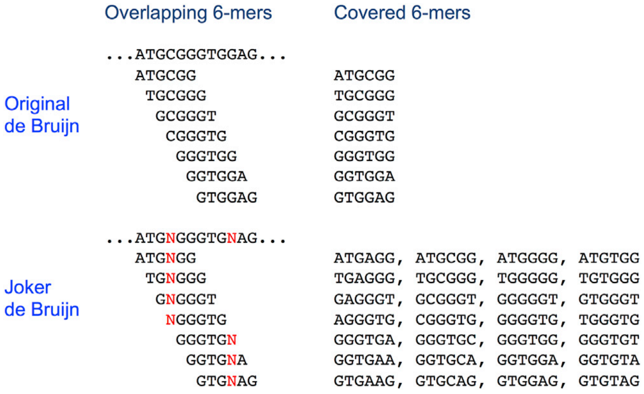


Figure 1. An Illustration of Subsequence of a Joker de Bruijn Sequence of Order k = 6 over DNA Alphabet Compared with an Original de Bruijn Sequence

measurements: a measurement of a single microarray spot is now assigned to multiple sequences instead of just one. Experimentally, the effective concentration of a high-affinity binder can be reduced up to 4-fold, leading to a concomitant decrease in the dynamic range of measured intensities. Thus, we limit the use of joker characters to one joker character per k-mer (Figure 1). Previous theoretical studies have considered the problem of covering all k-mers using joker characters, but with different restrictions and limitations, making them impractical for library design applications (Blanchet-Sadri et al., 2010; Chen et al., 2016b; Goeckner et al., 2016; Wyatt, 2013). None of these works considered the problem with the restriction that we defined, i.e., coverage of all k-mers with the limitation of one joker character per k-mer.

In this work, we study the problem of generating a minimum-length sequence to cover all k-mers, each at least p times, with at most one joker character per k-mer. We first present an overview of our novel algorithm, JokerCAKE, for generating compact joker de Bruijn sequences. JokerCAKE is based on two algorithmic steps: a greedy heuristic and an integer linear programming (ILP) formulation. We compare our results with the original de Bruijn sequence as well as a theoretical lower bound, and show that our approach achieves results that are near optimal. In addition, we simulate nearly 1,000 publicly available experiments that measure protein-DNA binding using the joker library and demonstrate that accurate binding scores for high-affinity k-mers can be inferred from them. Finally, we experimentally test protein-DNA binding on a joker library that covers all DNA 8-mers and present results in high agreement with our computational results. JokerCAKE and the universal sequences generated by it are freely available at: http://jokercake.csail.mit.edu and Data S1.

RESULTS

High-Level Description of JokerCAKE

We start with a high-level outline of the method and refer the reader to the STAR Methods for a detailed description of JokerCAKE, its implementation, and runtime and memory usage results. JokerCAKE (Joker Covering All K-mErs) is an algorithm for generating a short sequence that covers all k-mers using joker characters. The solution is based on two steps: (1) a greedy heuristic; and (1) an ILP formulation. The greedy heuristic examines at each step an addition of a joker character followed by $k-1$ characters from $\sum$. The addition that covers the most k-mers that are yet to be covered p times is chosen and added to the current sequence. The algorithm terminates when all k-mers have been covered at least p times. The ILP formulation minimizes the number of k-mers in the sequence under two sets of constraints. The first requires that each k-mer occurs at least p times. The second guarantees that the k-mer occurrences can form a sequence. The ILP is solved using Gurobi ILP solver version 6.5.2 (Gurobi Optimization, 2014), where it is given the greedy solution as a starting solution.

The two algorithms differ in runtime and optimality guarantees. The greedy approach is bounded in runtime by $O(|\sum|^{2k}\ p)$. Thanks to an efficient implementation, the runtime for k = 10 on a DNA alphabet takes less than 20 min. Our empirical results show that JokerCAKE produces sequences that are very close to the theoretical lower bound, implying near optimality. The ILP formulation solves the problem optimally but has no feasible bounds on the runtime. Thus, we limit the runtime in our tests. Note that even though the time limit we used is high (4 weeks), it has to be run only once to produce a sequence that covers all k-mers. Henceforth, the same sequence can be used for numerous technological implementations that require this value of k in their k-mer coverage. This sequence length is independent of oligo lengths in the experimental device, as the sequence can be cut into pieces of variable lengths. Moreover, the ILP solver benefits from running on multiple threads, so with more available computational resources, it can produce better results faster.

We demonstrate the reduced sequence size achieved by running JokerCAKE on variable combinations of the parameters (Figure 2): k, multiplicity p, and alphabet. We start by evaluating the greedy approach with p = 1 (i.e., covering each k-mer at least once) on two different alphabets: DNA and amino acid. For the DNA alphabet, we also added a feature to cover k-mers in reverse complement pairs, which enables a reduction by half in sequence length. Results show that the greedy approach produces sequences that are very close to the theoretical lower bound (Figures 2A–2C). To demonstrate the benefit of adding k characters at a time, we also applied a greedy approach, which adds one character at a time (compared with k characters). Moreover, the ILP reduces the sequence length even further, bringing it very close to the theoretical lower bound. We further evaluated the results as a function of the multiplicity p, i.e., how many times each k-mer has to be covered. Here, we observe fast convergence to the theoretical lower bound with p (Figures 2D–2F). We believe that this is due to the fact that the greedy algorithm can take many more “optimal steps” until it reaches the remaining “suboptimal steps” that are needed to cover all k-mers. This is also true for the greedy approach that adds one character at a time in the case of the amino acid alphabet. We did not run the ILP in the multiplicity test since the greedy results were near optimal.

JokerCAKE Libraries Perform Well Against Experimentally Captured Binding Scores

We used simulated data to demonstrate that the binding scores inferred for our joker library compare favorably with the original

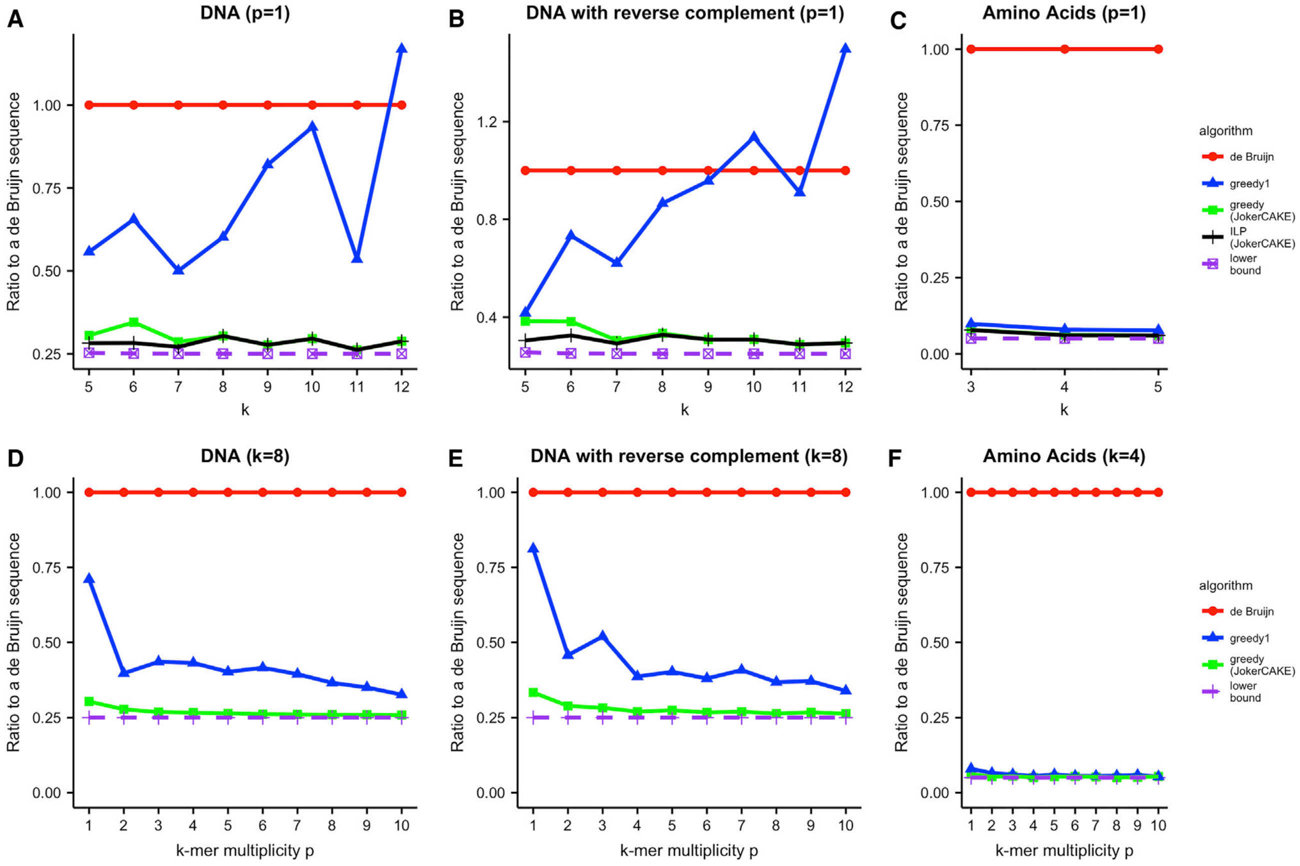


Figure 2. Results of JokerCAKE Compared with Original de Bruijn Sequences, a Simpler Approach, and Theoretical Lower Bound

We ran JokerCAKE on different combinations of k value, alphabet, and multiplicity p. Performance is measured as ratio of sequence length produced by JokerCAKE or greedy1 compared with a de Bruijn sequence.

(A–C) The performance is a function of k, where p = 1. (A) DNA; (B) DNA with reverse complement; (C) Amino acids.

(D–F) The performance is a function of p, where k = 8 for DNA and k = 4 for amino acid alphabets. (D) DNA; (E) DNA with reverse complement; (F) Amino acids. Greedy1 stands for the results for a greedy approach adding 1 character at time. Greedy stands for the results after the first greedy step of JokerCAKE. ILP stands for the result after improving the greedy solution using integer linear programming (ILP). A comparison of the runtimes and memory usage of the greedy algorithm and ILP solver are presented in Figures S1 and S2, respectively. Improvements in the ILP solution as a function of runtime are presented in Figure S3.

experimentally measured scores. After proving that JokerCAKE can efficiently reduce library size while at the same time covering all k-mers, we sought to determine how much information is lost in this reduction. To answer this question, we turned to UniPROBE, a database that includes data from 987 protein-binding microarray (PBM) experiments covering 528 different transcription factors (TFs) from multiple structural families and various species. Each PBM experiment includes binding scores of a specific TF to almost 42,000 35- to 36-long probe sequences designed to cover all 10-mers. For each experiment, we calculated 8-mer binding scores by computing the average binding intensity of all probes in which they occur. We then simulated results for experiments measuring TF binding to different libraries by assigning binding scores to each sequence in the library. The assigned score was the maximum 8-mer binding score among the 8-mers it contained. To compare the simulation with the original experiment, we calculated 8-mer binding scores in the same manner and compared the simulated and experimental results via Pearson correlation. Moreover, we calculated the success rate of consensus binding-site identification. We performed this test for three input libraries: (1) 0-joker: de Bruijn library of 38,387 DNA sequence covering all 10-mers with no joker characters; (2) 1-joker: joker library of 11,482 DNA sequences covering all 10-mers, with at most one joker character per 10-mer; and (3) 2-joker: joker library of 3,107 DNA sequences covering all 10-mers, with at most two joker characters per 10-mer. 0-joker and 2-joker libraries serve as an upper and lower bound on 1-joker, respectively. See STAR Methods for a detailed description of the simulation and testing.

Figure 3 shows the results of our experimental simulations comparing joker and de Bruijn libraries in measuring protein-DNA binding. The median Pearson correlation is 0.79 ± 0.08, 0.72 ± 0.09, and 0.59 ± 0.12 for the 0-joker, 1-joker, and 2-joker libraries, respectively (Figure 3A). While we see a small decrease in Pearson correlation (0.07 on average) when introducing 1 joker character per 10-mer, the increase is more significant when 2 joker characters are introduced (0.20 on average, with increased variance); in some cases the 2-joker correlation results even reach 0. However, those motifs determined to have the highest affinity in the original experiments consistently remain

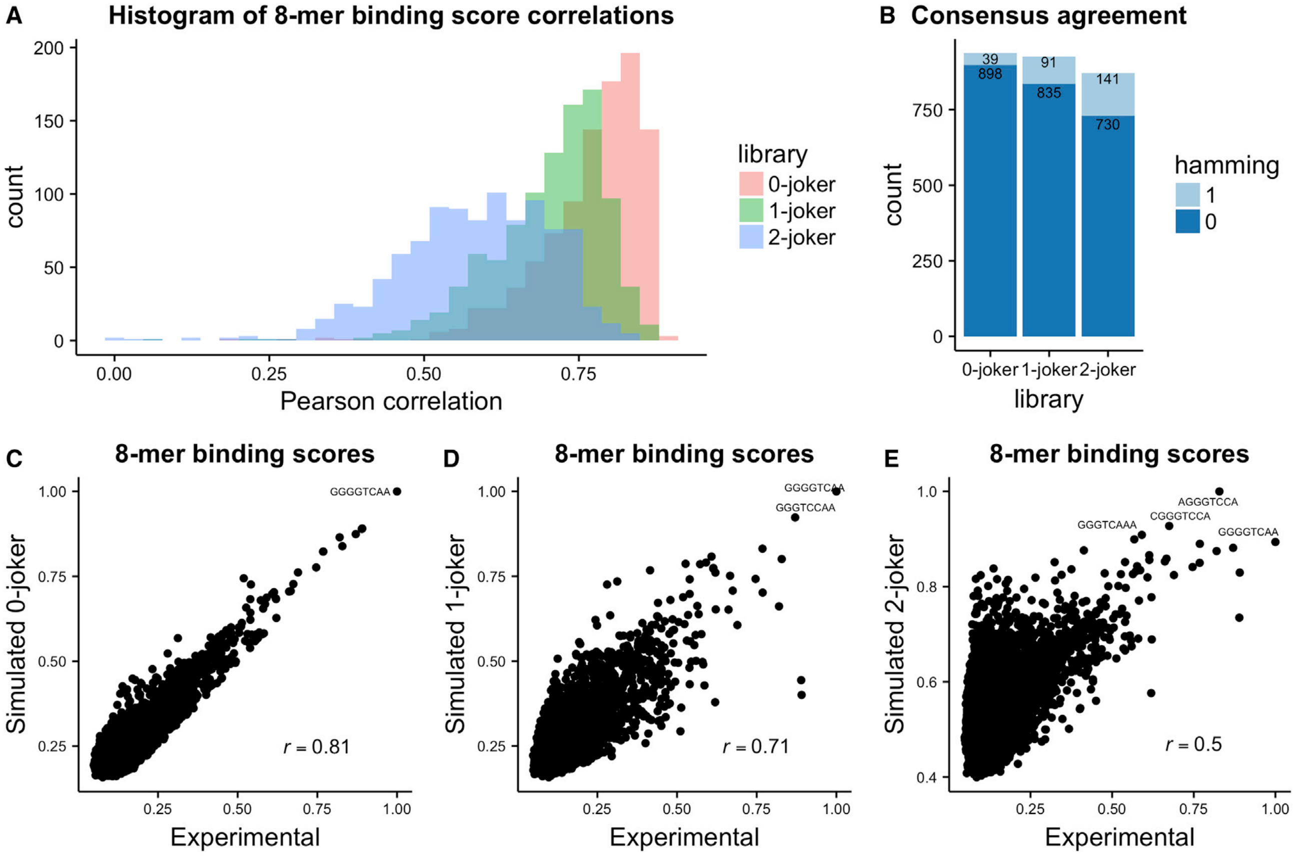


Figure 3. Simulation Results in Inference of Protein-DNA Binding Preferences Using Joker de Bruijn Libraries

For three different libraries covering all 10-mers, with at most 0/1/2 joker characters per 10-mer, binding scores were simulated for each PBM experiment out of 987.

(A) Histogram of Pearson correlations of 8-mer binding scores per experiment. For each experiment, experimental binding scores were compared with simulated scores on the three libraries.

(B) Identification of consensus binding sites in hamming distance. For each experiment, the hamming distance of the closest 6-mer between the top experimental and top simulated 8-mers was calculated.

(C–E) 8-mer binding scores of protein Hnf4a (binding GGGGTCAA; Hume et al., 2015). (C) 0-joker; (D) 1-joker; (E) 2-joker. The PBM experiment achieved median Pearson correlation on the 1-joker library.

among the highestscoring motifs in the simulated results for the joker libraries, confirming that this approach can identify global high-affinity binders and provide a "foothold" for subsequent experimental refinement. When counting the number of consensus binding sites identified correctly, we see that 0-joker and 1-joker libraries have similar performance of 94% and 93%, respectively, while the 2-joker library drops to an 88% success rate (Figure 3B). Thus, we effectively retain the power of correct consensus identification with a library that is smaller by a factor of almost four.

We highlight the enhanced performance by further focusing on one PBM experiment on which the median Pearson correlation was achieved (Hnf4a_2640.2_v2). For this experiment, we plot 8-mer binding scores inferred in simulation on the different libraries versus the original experimental binding scores (Figures 3C–3E). As expected, we observe a reduction in correlation with the usage of more joker characters. However, when only 1 joker character is used, scores of high-affinity 8-mers are correctly inferred, while accuracy is lost only for low-affinity 8-mers (Figure 3D).

JokeCAKE Library Performs Well in Experimental Validation

To validate our approach, we synthesized a joker library that covers all 8-mers in reverse complement pairs and experimentally measured binding of a well-characterized TF from *Saccharomyces cerevisiae* (Pho4) using the MITOMI platform (Fordyce et al., 2010; Maerkl and Quake, 2007). This joker library contained only 240 52-bp-long DNA sequences compared with an original library that required 740 52-bp-long oligonucleotides to cover all 8-mers. We gauged the accuracy of the new library in comparison with the original one by comparing k-mer binding scores obtained from each. As each 8-mer occurs at least once, each k-mer for k $\leq$ 6 occurs multiple times, allowing for inference of accurate k-mer binding scores. We also constructed a position weight matrix (PWM, a common model to represent protein-DNA binding preferences) from each experiment and visualized it as sequence logo.

The results of the experimental validation are in high concordance with our simulated results. Plots comparing k-mer scores

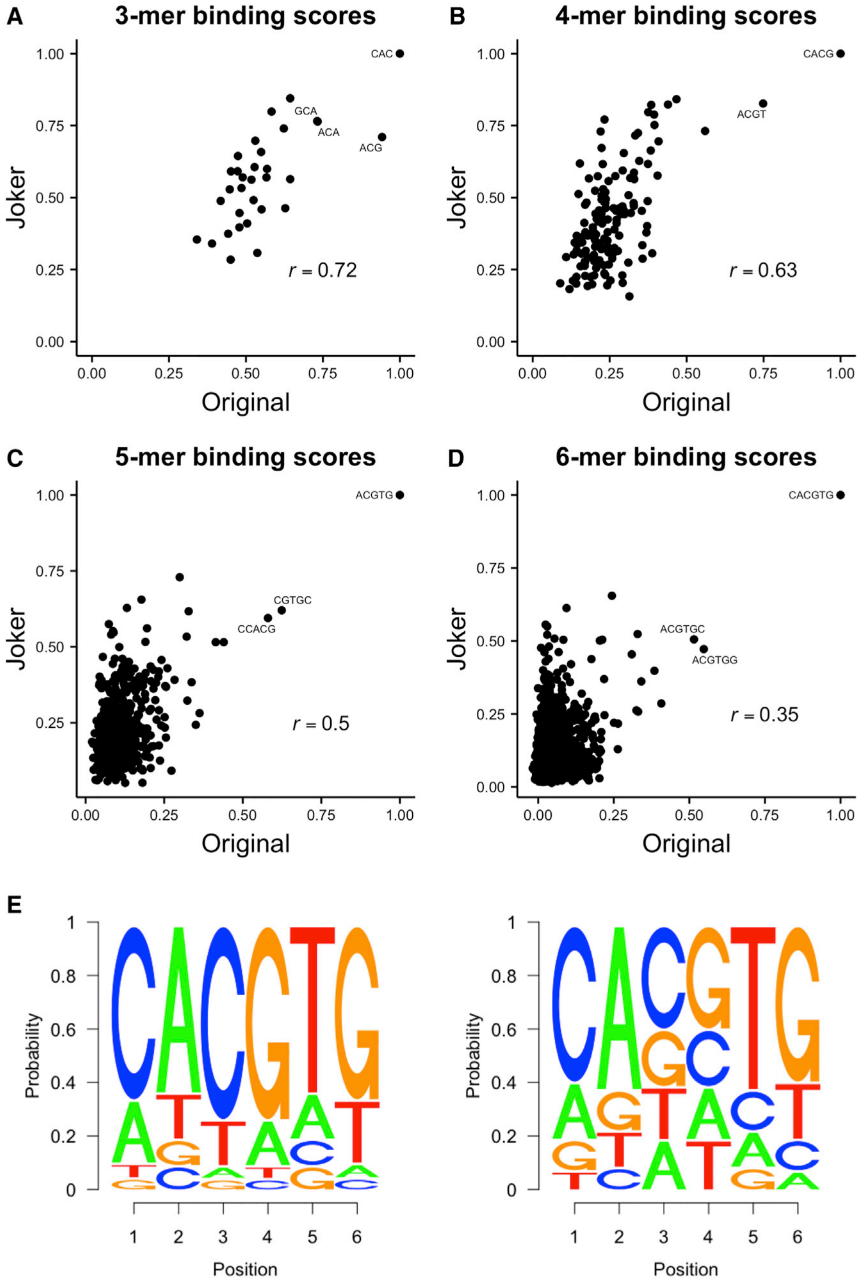


Figure 4. Results of a MITOMI Experiment on Joker Library Covering all 8-mers Compared with an Original MITOMI Experiment Measuring Pho4-DNA Binding

(A–D) Pearson correlation between k-mer scores derived from both experiments. (A) 3-mer binding scores; (B) 4-mer bindings scores; (C) 5-mer binding scores; (D) 6-mer binding scores.

(E) Sequence logos of PWMs generated from the original (left) and joker (right) experiments.

for 3 ≤ k ≤ 6 show that we can accurately infer k-mer scores for high-affinity k-mers, and the accuracy improves for low-affinity k-mers as k decreases (Figures 4A–4D). This finding is expected since as k decreases, k-mer occurrences increase; as a consequence, the statistical robustness improves. Pho4 is known to prefer CACGTG target sites, and the returned sequence logos show that CACGTG was successfully identified as the consensus binding site in both experiments (Figure 4E). Although the sequence logo generated from the joker experiment is less strict as the binding scores for lower-affinity k-mers are blurred (Figure 4D), these experiments establish that the use of joker characters can significantly reduce the library size while preserving the ability to retrieve high-affinity k-mers that can be directly probed in a second set of experiments.

DISCUSSION

While the use of joker characters can limit the ability to quantitatively identify both high- and low-affinity binders in a single experiment, this limitation is not a significant bottleneck for experimental protocols in which protein binding specificities are determined via a two-step experimental process. In the first “discovery” step, libraries that cover all k-mers, including joker characters, can be used to globally identify high-affinity candidate binding sequences via an unbiased search. In the second “refinement” step, a second set of experiments quantifying binding to a series of motifs containing systematic substitutions to the candidate consensus can be used to break the degeneracy, extend the length of the motif, and identify probable regulatory targets *in vivo*. Many MITOMI experiments already make use of such a two-step process, suggesting that introducing joker characters would not drastically change experimental workflows (Fordyce et al., 2012; Hernday et al., 2013; Lohse et al., 2013; Nelson et al., 2013).

Here, we demonstrate results for Pho4, a basic-helix-loop-helix transcription factor known to bind a relatively compact motif. However, we expect that the ability to extend k-mer search space within current experimental techniques will likely have the greatest impact for structural families that have proven difficult to study. The ability to extend k-mer search space is particularly useful for transcription factors known to bind half sites separated by a variable spacing, such as the poorly characterized fungal Zn_2Cys_6 TFs (Najafabadi et al., 2015) and other families known to bind extended motifs (e.g., homeodomain TFs; Yang et al., 2017).

Another clear advantage of our solution is its generality and flexibility. The alphabet is given as input to JokerCAKE, enabling a solution to any set of characters, including both oligonucleotide analogs and unnatural amino acids in the amino acid alphabet. Moreover, with a simple modification, both the greedy heuristic and ILP formulation can solve the problem of covering a specific set of k-mers, e.g., exclusion of specific k-mers for technical reasons (e.g., enzyme restriction sites as in RNAcompete (Ray et al., 2009)). More generally, our solution can be modified for variable k-mer multiplicities and inclusion of more than one joker character per k-mer.

We see several limitations in our study. First, our algorithm is not guaranteed to produce an optimal result in polynomial time. While the greedy heuristic is not guaranteed to produce an optimal result, we show empirically that it performs very well and produces a result that approaches the lower bound as the multiplicity p increases. The ILP solver is guaranteed to produce an optimal result but is not guaranteed to terminate in polynomial time; however, it too performed reasonably in practice. From our experience, we recommend using it for smaller alphabets and values of k, e.g., DNA alphabet and $k \leq 7$. With increased computational power and development of more efficient solvers, the ILP solution will be useful for larger alphabets and values of k. Second, the joker library introduces ambiguity in the measurements. Shrinking the library size comes at a cost of a smaller sample size, thus lowering the statistical robustness of the inferred scores. Still, in our simulated experiments and experimental validation, we were able to infer accurate binding scores for high-affinity k-mers, thereby identifying global minima within the binding specificity landscape and enabling detailed follow-up experiments to explore the local topography.

In summary, this work presents a new library design that covers all k-mers within a size that is almost $1/|\sum|$ smaller than current libraries. Our design enables the ability to measure interactions of longer k-mers with reduced costs. While for a DNA alphabet the savings may seem modest, they are significantly greater for an amino acid alphabet, where our design is 20 times smaller; for example, the ability to now handle k = 4 as opposed to 3 corresponds to an increase of 133% in information measured. We have made the implementation and calculated universal libraries freely available for researchers to use in designing unbiased library sequences. With our newly designed smaller libraries at increased k, we expect measurement of protein-DNA, -RNA, and -peptide interactions and the resulting research to significantly advance.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- CONTACT FOR REAGENT AND RESOURCE SHARING
- EXPERIMENTAL MODEL AND SUBJECT DETAILS
- METHODS DETAILS
  - ○ Experimental Validation of Joker Library
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - ○ Notation
  - ○ Greedy Heuristic
  - ○ ILP Formulation
  - ○ RC-Covering all K-mers
  - ○ Implementation
  - ○ Theoretical Lower Bound
  - ○ Open Questions
  - ○ Testing JokerCAKE Performance
  - ○ Simulation Experiments on Joker Library
  - ○ Comparison of Standard and Joker Library
- DATA AND SOFTWARE AVAILABILITY

SUPPLEMENTAL INFORMATION

Supplemental Information includes three figures and one data file and can be found with this article online at http://dx.doi.org/10.1016/j.cels.2017.07.006.

AUTHOR CONTRIBUTIONS

Y.O. and P.F. conceived the study. Y.O., R.K., and B.B. developed the greedy algorithm. Y.O. developed the ILP solution. All algorithms were developed under the supervision of B.B. Y.O. generated the sequence files and performed the simulations; Y.O., B.B., and P.F. evaluated the results. R.P. performed the binding experiment under the supervision of P.F. All authors contributed to writing the manuscript.

ACKNOWLEDGMENTS

This work was supported by the NIH (grant R01GM081871 to B.B., grant R00GM09984804 to P.F.). Part of this work was done while Y.O. was visiting the Simons Institute for the Theory of Computing. Part of this work was done while R.K. was visiting the Research Science Institute and was supported by the Center for Excellence in Education and their sponsors. P.F. is a Chan Zuckerberg Biohub Investigator and also acknowledges the support of a Gabilan and McCormick Fellowship for this work. An early version of this paper was submitted to and peer reviewed at the 2017 Annual International Conference on Research in Computational Molecular Biology (RECOMB). The manuscript was revised and then independently further reviewed at *Cell Systems.*

Received: February 25, 2017
Revised: April 14, 2017
Accepted: July 27, 2017
Published: September 27, 2017

REFERENCES

Berger, M.F., Philippakis, A.A., Qureshi, A.M., He, F.S., Estep, P.W., and Bulyk, M.L. (2006). Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. Nat. Biotechnol. *24*, 1429–1435.

Blanchet-Sadri, F., Schwartz, J., Stich, S., and Wyatt, B.J. (2010). Binary De Bruijn partial words with one hole. In Theory and Applications of Models of Computation. TAMC 2010, *vol 6108*, J. Kratochvíl, A. Li, J. Fiala, and P. Kolman, eds. (Springer), pp. 128–138.

Chen, D., Orenstein, Y., Golodnitsky, R., Pellach, M., Avrahami, D., Wachtel, C., Ovadia-Shochat, A., Shir-Shapira, H., Kedmi, A., Juven-Gershon, T., et al. (2016a). SELMAP - SELEX affinity landscape MAPping of transcription factor binding sites using integrated microfluidics. Sci. Rep. *6*, 33351.

Chen, H.Z.Q., Kitaev, S., Sun, B.Y. (2016b). On universal partial words over binary alphabets. arXiv:1601.06456.

Fordyce, P.M., Gerber, D., Tran, D., Zheng, J., Li, H., DeRisi, J.L., and Quake, S.R. (2010). De novo identification and biophysical characterization of transcription-factor binding sites with microfluidic affinity analysis. Nat. Biotechnol. *28*, 970–975.

Fordyce, P.M., Pincus, D., Kimmig, P., Nelson, C.S., El-Samad, H., Walter, P., and DeRisi, J.L. (2012). Basic leucine zipper transcription factor Hac1 binds DNA in two distinct modes as revealed by microfluidic analyses. Proc. Natl. Acad. Sci. USA *109*, E3084–E3093.

Goeckner, B., Groothuis, C., Hettle, C., Kell, B., Kirkpatrick, P., Kirsch, R., Solava, R. (2016). Universal partial words over non-binary alphabets. arXiv:1611.03928

Gurard-Levin, Z.A., Kilian, K.A., Kim, J., Bähr, K., and Mrksich, M. (2010). Peptide arrays identify isoform-selective substrates for profiling endogenous lysine deacetylase activity. ACS Chem. Biol. *5*, 863–873.

Gurobi Optimization. (2014). Gurobi Optimizer Reference Manual. www.Gurobi.Com 6, 572.

Hernday, A.D., Lohse, M.B., Fordyce, P.M., Nobile, C.J., DeRisi, J.L., and Johnson, A.D. (2013). Structure of the transcriptional network controlling white-opaque switching in *Candida albicans*. Mol. Microbiol. *90*, 22–35.

Hume, M.A., Barrera, L.A., Gisselbrecht, S.S., and Bulyk, M.L. (2015). UniPROBE, update 2015: new tools and content for the online database of protein-binding microarray data on protein-DNA interactions. Nucleic Acids Res. *43*, D117–D122.

Jolma, A., Kivioja, T., Toivonen, J., Cheng, L., Wei, G., Enge, M., Taipale, M., Vaquerizas, J.M., Yan, J., Sillanpää, M.J., et al. (2010). Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. Genome Res. *20*, 861–873.

Lohse, M.B., Hernday, A.D., Fordyce, P.M., Noiman, L., Sorrells, T.R., Hanson-Smith, V., Nobile, C.J., DeRisi, J.L., and Johnson, A.D. (2013). Identification and characterization of a previously undescribed family of sequence-specific DNA-binding domains. Proc. Natl. Acad. Sci. USA *110*, 7660–7665.

Maerkl, S.J., and Quake, S.R. (2007). A systems approach to measuring the binding energy landscapes of transcription factors. Science *315*, 233–237.

Najafabadi, H.S., Mnaimneh, S., Schmitges, F.W., Garton, M., Lam, K.N., Yang, A., Albu, M., Weirauch, M.T., Radovani, E., Kim, P.M., et al. (2015). C2H2 zinc finger proteins greatly expand the human regulatory lexicon. Nat. Biotechnol. *33*, 555–562.

Nelson, C.S., Fuller, C.K., Fordyce, P.M., Greninger, A.L., Li, H., and DeRisi, J.L. (2013). Microfluidic affinity and ChIP-seq analyses converge on a conserved FOXP2-binding motif in chimp and human, which enables the detection of evolutionarily novel targets. Nucleic Acids Res. *41*, 5991–6004.

O'Donoghue, A.J., Eroy-Reveles, A.A., Knudsen, G.M., Ingram, J., Zhou, M., Statnekov, J.B., Greninger, A.L., Hostetter, D.R., Qu, G., Maltby, D.A., et al. (2012). Global identification of peptidase specificity by multiplex substrate profiling. Nat. Methods *9*, 1095–1100.

Orenstein, Y., and Berger, B. (2016). Efficient design of compact unstructured RNA libraries covering all k-mers. J. Comput. Biol. *23*, 67–79.

Orenstein, Y., Mick, E., and Shamir, R. (2013). RAP: accurate and fast motif finding based on protein-binding microarray data. J. Comput. Biol. *20*, 375–382.

Orenstein, Y., and Shamir, R. (2013). Design of shortest double-stranded DNA sequences covering all k-mers with applications to protein-binding microarrays and synthetic enhancers. Bioinformatics *29*, i71–i79.

Philippakis, A.A., Qureshi, A.M., Berger, M.F., and Bulyk, M.L. (2008). Design of compact, universal DNA microarrays for protein binding microarray experiments. J. Comput. Biol. *15*, 655–665.

Ray, D., Kazan, H., Chan, E.T., Peña Castillo, L., Chaudhry, S., Talukder, S., Blencowe, B.J., Morris, Q., and Hughes, T.R. (2009). Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. Nat. Biotechnol. *27*, 667–670.

Ray, D., Kazan, H., Cook, K.B., Weirauch, M.T., Najafabadi, H.S., Li, X., Gueroussov, S., Albu, M., Zheng, H., Yang, A., et al. (2013). A compendium of RNA-binding motifs for decoding gene regulation. Nature *499*, 172–177.

Smith, R.P., Riesenfeld, S.J., Holloway, A.K., Li, Q., Murphy, K.K., Feliciano, N.M., Orecchia, L., Oksenberg, N., Pollard, K.S., and Ahituv, N. (2013). A compact, in vivo screen of all 6-mers reveals drivers of tissue-specific expression and guides synthetic regulatory element design. Genome Biol. *14*, R72.

Wyatt, B.J. (2013). De Bruijn Partial Words (The University of North Carolina at Greensboro).

Yang, L., Orenstein, Y., Jolma, A., Yin, Y., Taipale, J., Shamir, R., and Rohs, R. (2017). Transcription factor family-specific DNA shape readout revealed by quantitative specificity models. Mol. Syst. Biol. *13*, 910.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Antibodies		
Anti 6x His tag antibody (biotin)	Abcam	27025; RRID: AB_470880
Chemicals, Peptides, and Recombinant Proteins		
Pierce Bovine Serum Albumin, Biotinylated	Thermo Fisher	29130
NeutrAvidin	Thermo Fisher	31000
Klenow Fragment, exonuclease -	New England Biolab	M0212L
Critical Commercial Assays		
TnT T7 Quick Coupled In Vitro Transcription/Translation kit	Promega	L4610
Fluorotect Green BODIPY-labeled charged lysine tRNA	Promega	L5001
Deposited Data		
UniPROBE database	Harvard University	http://the_brain.bwh.harvard.edu/uniprobe/
Oligonucleotides		
Alexa-647-labeled 5'-A647-GTCATACCGCCGGA-3'	Integrated DNA technologies	Custom
Recombinant DNA		
Pho4-5xHis cloned into pTnT plasmid	AddGene	N/A
Software and Algorithms		
Java	Java	https://www.java.com/en/
Gurobi 6.5.2	Gurobi Optimizer	http://www.gurobi.com/
JokerCAKE	This study	N/A

CONTACT FOR REAGENT AND RESOURCE SHARING

Further information and requests for resources and reagents should be directed to and will be fulfilled by the Lead Contact, Bonnie Berger (bab@mit.edu).

EXPERIMENTAL MODEL AND SUBJECT DETAILS

Proteins used in these experiments were generated via *in vitro* transcription/translation of *S. cerevisiae* Pho4 in cell free extracts; no organisms were used.

METHODS DETAILS

Experimental Validation of Joker Library

A pseudorandom oligonucleotide library with wildcard characters was generated by specifying 4-fold degenerate nucleotides ('N') at wildcard positions within 70-bp oligonucleotides (Integrated DNA Technologies). Experiments measuring transcription factor binding to this wildcard library were performed largely as described previously (Fordyce et al., 2010, 2012). Briefly, each sequence in the library was fluorescently labeled and converted to double-stranded DNA via hybridization to a universal Alexa 647-labeled oligonucleotide (Integrated DNA Technologies) followed by extension with Klenow fragment, exonuclease minus (New England Biolabs). After synthesis, the library was printed using a custom-built robotic microarrayer onto epoxysilane-treated glass slides (ThermoFisher). A MITOMI microfluidic device was aligned to the microarray and the transcription factor affinity assay was performed by expressing Pho4 in rabbit reticulocyte lysate (TnT T7 Quick Coupled In Vitro Transcription/Translation kit, Promega) in the presence of BODIPY-labeled charged lysine tRNAs (Fluorotect Green, Promega), recruiting it to antibody-patterned surfaces (created by sequentially flowing biotinylated BSA (ThermoFisher), Neutravidin (ThermoFisher), and biotinylated anti-pentaHis antibodies (Abcam)), and mechanically trapping the transcription factor-oligonucleotide interactions using on-chip valves. The device was then imaged using an inverted fluorescence microscope (Nikon Ti-E or Ti-S) to quantify levels of surface-immobilized transcription factors and bound DNA. Images were automatically stitched using Fiji software and analyzed using custom image analysis software written in Matlab.

QUANTIFICATION AND STATISTICAL ANALYSIS

Notation

A k-mer is a word of length k over a given alphabet $\sum$. In this study, we refer to two alphabets $\sum_{AA}$={A,R,N,D,C,Q,E,G,H,I, L,K,M,F,P,S,T,W,Y,V} and $\sum_{DNA=}${A,C,G,T}. In the text below, we interchangeably refer to a k-mer as a word and an integer by the natural conversion in base $|\sum|$. For example, {A,C,G,T}={0,1,2,3} and AGC = $0 \cdot 4^0 + 2 \cdot 4^1 + 1 \cdot 4^2 = 24$.

A joker character, denoted by x, represents all characters in $\sum$, i.e. x representing {A,C,G,T}. K-mer w=($w_1,\ldots,w_k$) is covered by sequence S if there exists $0 \le i \le |S|$-k such that for $1 \le j \le k$: $S_{i+j} \in \{x, w_j\}$. We say that w occurs at index i in S. In other words, any original character of W may be replaced by the joker character.

We define a (k,p,$\sum$)-joker de Bruijn sequence as a sequence covering all k-mers, each at least p times, with at most one joker character per k consecutive characters. K-mer w is covered at least p times by sequence S if there are p distinct indices {$i_1,\ldots,i_p$} such that w occurs at index i_j in S for $1 \le j \le p$.

We also define reverse complementarity. A complement relation is a symmetric non-reflexive relation, i.e. $\overline{A} = T$ and $\overline{C} = G$. The reverse complement of k-mer $w = \{w_1, \ldots, w_k\}$ is $RC(w) = \{\overline{w_k}, \ldots, \overline{w_1}\}$. A k-mer is RC-covered by sequence S if it occurs in either S or RC(S). A (k,p,RC,$\sum$)-joker de Bruijn sequence RC-covers each k-mer over $\sum$ at least p times.

In this study, we consider the following problem and its version utilizing the reverse complement property.

MINIMUM-LENGTH (k, p, $\sum$)-JOKER DE BRUIJN SEQUENCE

INSTANCE: k value, multiplicity p, alphabet $\sum$.

VALID SOLUTION: (k, p, $\sum$)-joker de Bruijn sequence S.

GOAL: Minimize |S|.

Greedy Heuristic

We describe in detail the greedy algorithm, which is the first step in JokerCAKE, to find a (k, p, $\sum$)-joker de Bruijn sequence. It is based on a greedy heuristic that examines at each step an addition of a joker character followed by k-1 characters from $\sum$. The addition that covers the most k-mers that are yet to be covered p times is chosen and added to the current sequence. The algorithm terminates when all k-mers have been covered at least p times. The algorithm is summarized as Algorithm 1.

We bound the runtime of Algorithm 1. We first prove the following Lemma on the minimum number of k-mers covered in each iteration of the top while loop (line 4 in Algorithm 1).

```
Algorithm 1 Generate a (k, p, $\sum$)-joker de Bruijn sequence
1: Set CURR to be an arbitrary (k-1)-mer over $\sum$.
2: Initialize SEQ to CURR.
3: Initialize array A of k-mers counts to 0.
4: while there are still k-mer counts in A smaller than p do
5:   Initialize MAX to 0.
6:   for all (k-1)-mers over $\sum$ W do
7:     Set COUNT to number of unique k-mers CURR x W newly covers.
8:       if COUNT>MAX then
9:         MAX=COUNT.
10:          $MAX_{K-1MER}$=W.
11:      end if
12:  end for
13: Set SEQ= SEQ x $MAX_{K-1MER}$.
14: Update array A according to newly covered k-mers by CURR x $MAX_{K-1MER}$.
15: Set CURR=$MAX_{K-1MER}$.
16: end while
17: Output sequence SEQ.
```

Lemma 1. In each iteration of the while loop in Algorithm 1 at least one k-mer is newly covered.

Proof. Denote W a k-mer that is yet to be covered p times. The inner for loop (line 6) iterates over all possible (k-1)-mers, including the (k-1)-suffix of W, denoted by s_{k-1}(W). Thus, CURR x s_{k-1}(W) newly covers W. Since the for loop finds the maximum, it has to be at least one.

Corollary 1. The number of iterations of the while loop in Algorithm 1 is bounded by p$|\sum|^k$.

Proof. The number of k-mers that have to be covered is p$|\sum|^k$. By Lemma 1 at least one k-mer is newly covered at each iteration. Thus, the bound on the total number of iterations is p$|\sum|^k$.

Theorem 1. The running time of Algorithm 1 is bounded by O(p$|\sum|^{2k-1}$k).

Proof. The while loop runs at most p$|\sum|^k$ iterations by Corollary 1. The inner for loop runs $|\sum|^{k-1}$ iterations since it iterates over all (k-1)-mers. Inside the if statement exactly 2k-1 k-mers in CURR x $MAX_{K\text{-}1MER}$ are examined. We assume that to examine each k-mer takes constant time O(1) as it is one array operation. Thus, the total running time is O(p$|\sum|^{2k-1}$k).

ILP Formulation

Next, we describe in detail the ILP formulation, which is the second step in JokerCAKE, to solve the MINIMUM-LENGTH (k, p, $\sum$)-JOKER DE BRUIJN problem. We start with defining the variables. X variables are k-mer counts of k-mers with no joker character. Y variables are k-mer counts of k-mers that include one joker character. A and Z variables define the start and end of the sequence. See the following definition:

1. $|\sum|^k$ integer variables X_i. Each X_i corresponds to the number of times the exact k-mer occurs in the sequence (with no joker character).
2. k•$|\sum|^{k-1}$ integer variables $Y_{i,j}$. Each $Y_{i,j}$ corresponds to the number of times a k-mer with one joker character at position j and the rest of the positions as (k-1)-mer i occurs in the sequence.
3. 2$|\sum|^{k-1}$ binary variables. A_i/Z_i corresponds to the starting/ending (k-1)-mer of the sequence, respectively.

As we aim for the shortest sequence, the objective function is

$$min \sum_{i=1}^{|\sum|^k} X_i + \sum_{i=1}^{|\sum|^{k-1}} \sum_{j=1}^{k} Y_{i,j}$$

The first constraint is the coverage constraint, which requires that all k-mers occur at least p times. Let f(i,j) be the (k-1)-mer of all positions but j of k-mer i.

$$X_i + \sum_{j=1}^{j=k} Y_{f(i,j),j} \geq p \quad 1 \leq i \leq \left|\sum\right|^k$$

The second constraint guarantees that the k-mer occurrences can form a sequence. We require that for each (k-1)-mer (including those with one joker character) the number of k-mers with that (k-1)-mer in their suffix is equal to the number of k-mers with that (k-1)-mer in their prefix (except for two, which allows the formation of a sequence instead of requiring a cycle). Denote p_x(i) and s_x(i) the x-long prefix and suffix of i, respectively.

For (k-1)-mers with no joker character:

$$A_i + Y_{i,1} + \sum_{s_{k-1}\left(i'\right)=i} X_{i'} = Z_i + Y_{i,k} + \sum_{p_{k-1}\left(i'\right)=i} X_{i'} \quad 1 \leq i \leq \left|\sum\right|^{k-1}$$

For (k-1)-mers with a joker character at position 1 ≤ j ≤ k-1:

$$\sum_{s_{k-2}\left(i'\right)=i} Y_{i',j+1} = \sum_{p_{k-2}\left(i'\right)=i} Y_{i',j} \quad 1 \leq i \leq \left|\sum\right|^{k-2},\ 1 \leq j \leq k-1$$

And to ensure that only one (k-1)-mer is at the beginning of the sequence and one at the end, we require:

$$\sum_{i=1}^{|\sum|^{k-1}} A_i = \sum_{i=1}^{|\sum|^{k-1}} Z_i \leq 1$$

RC-Covering all K-mers

To further shrink libraries over double-stranded DNA, we utilize the reverse complement property and generate a (k, p, RC, $\sum$)-joker de Bruijn sequence. We made two modifications to the algorithms above. For Algorithm 1 whenever we consider and choose a new addition of k-1 characters and a joker character (lines 7 and 14), we need to account for both the k-mers and their reverse complement. For the ILP formulation we modified the coverage constraint. The modified constraint is:

$$X_i + X_{RC(i)} \sum_{j=1}^{j=k} Y_{f(i,j),j} + Y_{f(RC(i),j),j} \geq p \quad 1 \leq i \leq \left|\sum\right|^k$$

Implementation

We implemented the algorithms in Java. We used Gurobi ILP solver version 6.5.2 (Gurobi Optimization, 2014). We set the Method parameter in Gurobi to 3 as recommended to improve the running time of the root relaxation process. We set a time limit for the ILP solver since solutions for $k \geq 5$ for DNA and $k \geq 3$ for amino acid alphabet did not terminate based on the default criteria. Running times were benchmarked on a single CPU of a 20-CPU Intel Xeon E5-2650 (2.3GHz) machine with 384GB 2133MHz RAM.

Theoretical Lower Bound

We prove theoretical lower bounds for (k, p, $\sum$)-de Bruijn and (k, p, RC, $\sum$)-de Bruijn sequences.

Theorem 2. Denote by n(k, p, $\sum$) and n(k, p, RC, $\sum$) the lengths of a (k, p, $\sum$)-de Bruijn sequence and (k, p, RC, $\sum$)-de Bruijn sequence, respectively. Then,

$$n\left(k,\ p,\sum\right) \geq \left|\sum\right|^{k-1} + k - 1$$

$$n\left(k,\ p,\sum\right) = \begin{cases} \dfrac{\left|\sum\right|^{k-1}}{2} + k - 1, & k \text{ is odd} \\ \dfrac{\left|\sum\right|^{k-1} + \left|\sum\right|^{k/2-1}}{2} + k - 1, & k \text{ is even} \end{cases}$$

Proof. The number of k-mers over alphabet $|\sum|$ is $|\sum|^k$. The number of reverse complement k-mer pairs is $|\sum|^k/2$ for odd k and $(|\sum|^k + |\sum|^{k/2})/2$ for even k due to reverse complement palindromes. Since there is at most one joker character per k-mer, the number of k-mers in the sequence can be reduced by at most $|\sum|$. For a non-cyclic sequence, k-1 characters need to be added.

Open Questions

Several open questions remain from our study. First, is there an optimal solution that runs in time polynomial in $O(p|\sum|^k)$? Second, is there a good enough heuristic that runs in time linear in the output length, i.e. $O(p|\sum|^k)$, or at least asymptotically faster than Algorithm 1? Third, can we provide tighter lower and upper bounds?

Testing JokerCAKE Performance

We ran JokerCAKE with p=1 on DNA alphabet with $5 \leq k \leq 12$, DNA alphabet in reverse complement pairs with $5 \leq k \leq 12$ and amino acid alphabet with $3 \leq k \leq 5$. We also ran it with $1 \leq p \leq 10$ on these alphabets with k=8, 8 and 4, respectively. We compared the results with a length of an original de Bruijn sequence $|\sum|^k p+k-1$ over DNA and amino acid alphabets, and approximately half when considering reverse complement pairs. We also compared to a greedy approach adding 1 character at a time. We added a theoretical lower bound, which is approximately $1/|\sum|$ of a length of an original de Bruijn sequence.

Simulation Experiments on Joker Library

We downloaded all protein binding microarray (PBM) experiments from UniPROBE database (Hume et al., 2015), a total of 987 experiments. Each experiment contains almost 42,000 35-36-long DNA sequences covering all 10-mers together with corresponding binding intensities of a specific protein. For each experiment, we inferred 8-mer binding scores by calculating the average binding intensities of the probes they appear in (including as reverse complement) (Orenstein et al., 2013). We simulated a PBM experiment on three different libraries: 0-joker, 1-joker, 2-joker. All cover all 10-mers, with the difference in the numbers of jokers per 10-mer (0,1,2, respectively). The 0-joker was generated by a de Bruijn sequence, 1-joker by JokerCAKE and 2-joker by a variant of JokerCAKE allowing 1 joker per 5-mer while covering all 10-mers. We note that having more than one joker character in a k-mer is undesirable due to the high degeneracy, and thus we did not implement this feature in JokerCAKE. Each sequence was chopped into 36-long DNA sequences with an overlap of 9bp not to lose any 10-mer. For each sequence in this library we assigned the maximum 8-mer score that occurs in it, where for 8-mers that contain joker characters we took the average score of the 8-mers it represents. Finally, we calculated 8-mer binding scores on the simulated experiment in the same fashion as on the experimental PBM data. Moreover, we identified a consensus sequence for each experiment as the 8-mer whose sum of scores of itself and all its neighbors in one hamming distance was the highest. We calculated the similarity between two consensus 8-mers as the hamming distance between the closest 6-mers they contain (taking into account the reverse complement). We considered a hamming distance ≤ 1 to the consensus of the original experiment as correctly identified consensus.

Comparison of Standard and Joker Library

We compared this experiment to an experiment with the same 8-mer coverage but with no joker characters. For each experiment we inferred k-mer binding scores for $k \leq 6$ by calculating the average binding intensities of the oligos they occur in. These were compared by Pearson correlation. PWMs were generated by the highest-affinity 6-mer and its 1-hamming distance neighbors as was recently done for high-throughput SELEX data (Chen et al., 2016a; Jolma et al., 2010). For each position in the PWM the nucleotide weights corresponded to the scores of the 6-mers that vary in that position. For example, scores of CACGTG, AACGTG, GACGTG and TACGTG were used as the weights in the first position of the PWM. We could not use the approach that was previously used for MITOMI data as it cannot be applied to degenerate sequences (Fordyce et al., 2010).

DATA AND SOFTWARE AVAILABILITY

JokerCAKE and the universal sequences generated by it are freely available at: http://jokercake.csail.mit.edu and Data S1 supplemental file. The MITOMI experiments on Pho4 protein using the standard and joker libraries have been deposited in the GEO database under accession numbers GEO: GSE99723, GSM2650866 and GPL23547.

Cell Systems, Volume *5*

Supplemental Information

Optimized Sequence Library Design for Efficient *In Vitro* Interaction Mapping

Yaron Orenstein, Robert Puccinelli, Ryan Kim, Polly Fordyce, and Bonnie Berger

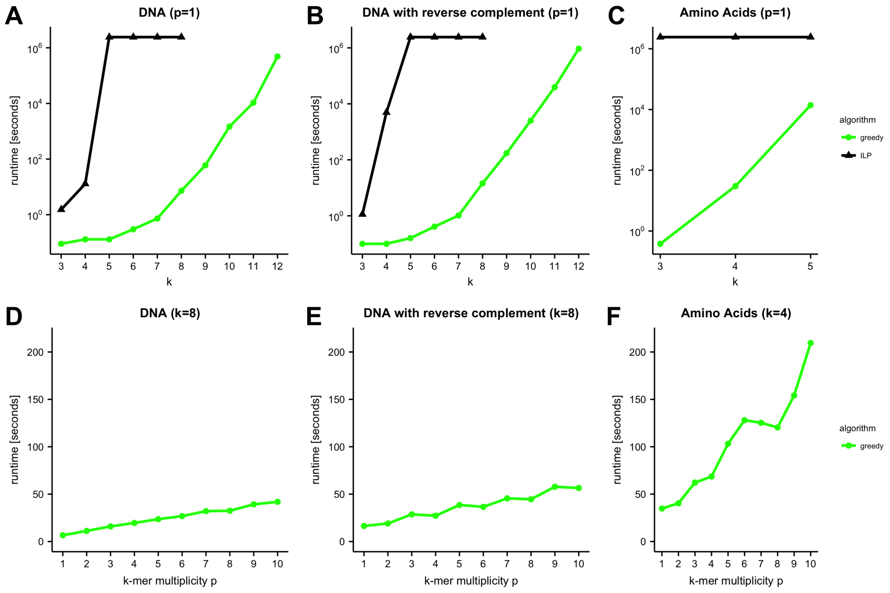


Figure S1. Runtimes of the greedy algorithm and ILP solver. Related to Figure 2. Green and black lines correspond to greedy algorithm and ILP solver, respectively. In panels A, B, C the performance is a function of k, where $p = 1$. In panels D, E, F the performance is a function of p, where $k = 8, 4$ for DNA and amino acids alphabets, respectively. Running times were benchmarked on a single CPU of a 20-CPU Intel Xeon E5-2650 (2.3 GHz) machine with 384 GB 2133 MHz RAM. ILP results for DNA alphabet and $k > 8$ are not available since Gurobi ILP solver did not complete the initialization for $k = 8$ in the given time limit. Note that in A,B,C the y-axis is in logarithmic scale.

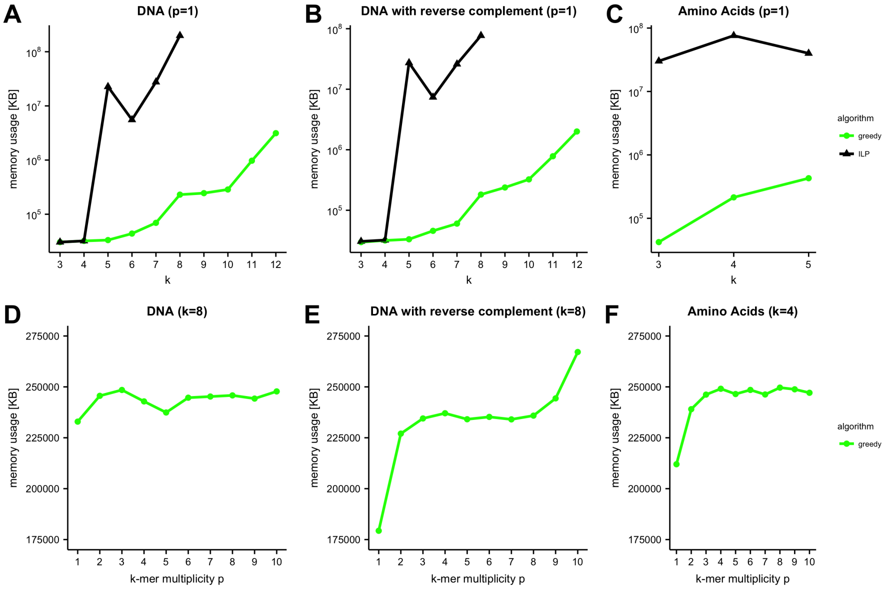


Figure S2. Memory usage for the greedy algorithm and ILP solver. Related to Figure 2. Green and black lines correspond to greedy algorithm and ILP solver, respectively. In panels A, B, C the performance is a function of k, where $p = 1$. In panels D, E, F the performance is a function of p, where $k = 8, 4$ for DNA and amino acids alphabets, respectively. Memory usage was benchmarked on a single CPU of a 20-CPU Intel Xeon E5-2650 (2.3 GHz) machine with 384 GB 2133 MHz RAM. ILP results for DNA alphabet and $k > 8$ are not available since Gurobi ILP solver did not complete the initialization for $k = 8$ in the given time limit. Note that in A,B,C the y-axis is in logarithmic scale.

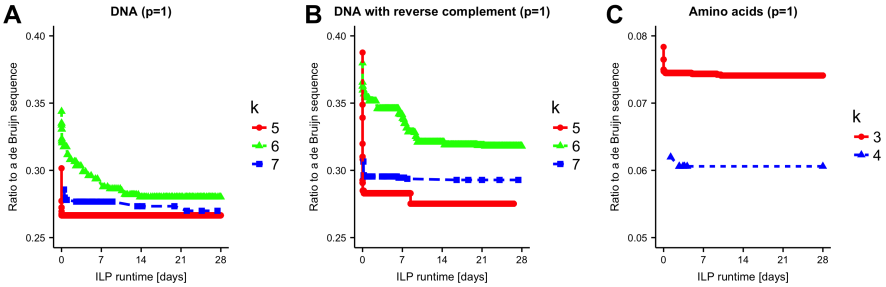


Figure S3. ILP solution improvements as a function of runtime. Related to Figure 2. In panels A and B colors red, green and blue correspond to $k = 5, 6, 7$, respectively, on DNA alphabet. In panel C colors red and blue correspond to $k = 3, 4$, respectively, on amino acids alphabet. Ratio of ILP solution sizes to a de Bruijn sequence are plotted as a function of the runtime up to 28 days.