RICE UNIVERSITY

Machine Learning in Large-scale Genomics: Sensing, Processing, and Analysis

by

Amirali Aghazadeh

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Richard G. Baraniuk, Chair
Victor E. Cameron Professor of Electrical and Computer Engineering

Rebekah A. Drezek
Professor of Bioengineering and Electrical and Computer Engineering

Anshumali Shrivastava
Assistant Professor of Computer Science

Houston, Texas

April, 2017

ABSTRACT

Machine Learning in Large-scale Genomics:
Sensing, Processing, and Analysis

by

Amirali Aghazadeh

Advances in the field of genomics, a branch of biology concerning with the structure, function, and evolution of genomes, has led to dramatic reductions in the price of sequencing machines. As a result, torrents of genomic data is being produced every day which pose huge challenges and opportunities for engineers, scientists, and researchers in various fields. Here, we propose novel machine learning tools and algorithms to more efficiently sense, process, and analyze large-scale genomic data.

To begin with, we develop a novel universal microbial diagnostics (UMD) platform to sense microbial organisms in an infectious sample, using a small number of random DNA probes that are agnostic to the target genomic DNA sequences. Our platform leverages the theory of sparse signal recovery (compressive sensing) to identify the composition of a microbial sample that potentially contains thousands of novel or mutant species.

We next develop a new sensor selection algorithm that finds the subset of sensors that best recovers a sparse vector in sparse recovery problems. Our proposed algorithm, Insense, minimizes a coherence-based cost function that is adapted from classical results in sparse recovery theory and outperforms traditional selection algorithms in finding optimal DNA probes for microbial diagnostics problem.

Inspired by recent progress in robust optimization, we then develop a novel hashing al-

gorithm, dubbed RHash, that minimizes the worst-case distortion among pairs of points in a dataset using an ℓ_∞-norm minimization technique. We develop practical and efficient implementations of RHash based on the alternating direction method of multipliers (ADMM) framework and column generation that scale well to large datasets.

Finally, we develop a novel machine learning algorithm using techniques in deep learning and natural language processing literature to embed DNA sequences of arbitrary length into a single low-dimensional space. Our so-called *K*mer2Vec platform learns biological concepts such as drug-resistance by parsing raw DNA sequences of microbial organisms with no prior biology knowledge.

Acknowledgments

First and for most I want to thank my advisor Prof. Richard Baraniuk who has been the biggest supporter in my exploration into the unknown territories of science: from signal processing to machine learning, from diagnostics to genomics, and from shallow learning to deep learning. He taught me how to work in interdisciplinary environments and how to communicate with, learn from, and collaborate with scientists who might talk in different scientific language. He taught me how to think big, how to find big problems, and how to be patient to solve the big problems of the world.

I have been very fortunate to have Prof. Rebekah Drezek as my PhD committee member. She has always been supporting me in the long journey that my first project took. During these years, she encouraged me to reach my goals and guided me in the fields of biology that I had just started to explore. I also want to thank my PhD committee member, Anshumali Shrivastava. He introduced me to the field of large-scale machine learning. Our meetings and conversations could not have been more exciting, helpful and thought-provoking.

I want to thank Mona Sheikh, my first academic mentor at Rice. She helped me a lot starting off my research in diagnostics. I want to thank Mr. Lan with whom I share a lot not only in the academic world but in the world outside our office – from playing and watching soccer to exlporing the cheapest restaurants in Houston. I want to thank Rice DSP's students Jason Laska, Chinmay Hegde, Eva Dyer, Andrew Waters, Ali Mousavi, Ali Ayremlou, Chris Metzler, Minh Tan Nguyen, CJ Barberan, Jasper Tan, and postdocs, Aswin Sankaranarayanan, Christoph Studer, Tom Goldstein, Divyanshu Vats, Arian Maleki, Eric Chi, Moahammad Golbabaee, Ankit Patel, Salman Asif, and Gautam Dasarathy with whom I had the most interesting discussions. I also want to thank all the members in Drezek

Lab including Adam Lin, Allen Chen, Vishu Asthana, and Pallavi Buga, whom I enjoyed collaborating wth.

This would not be possible without non-stopping supports of my family. My mother, who inspired me to work in the research direction I am in, my father who motivates me to pursue my dreams everyday, and my sister who has alway been there for me.

I finally want to thank my partner in life, Vida, who believed in me more than I do, who helped me to stand the hardest moments of this journey, and who encouraged me to never give up. She made our PhD life *orders of magnitude* more fun and exciting.

“Dedicated to my mother, Hayedeh”

Contents

Abstract ii

Acknowledgments iv

List of Illustrations xii

List of Tables xxiii

1 Introduction 1

2 Background 5

2.1 Preliminaries 5

2.1.1 Notation 5

2.1.2 Vector Norm 5

2.1.3 Sparse signals 6

2.2 Compressive sensing 7

2.2.1 Compressive sensing theory 7

2.2.2 Compressive sensing for non-negative signals 8

2.2.3 Greedy recovery algorithms 9

2.2.4 Basis pursuit denoising 10

2.3 Whole genomic sequence analysis tools 11

2.3.1 DNA thermodynamic hybridization model 13

2.4 Molecular beacons 16

2.4.1 Principle of operation 16

2.4.2 Advantages for microbial detection 16

2.4.3 Sloppy molecular meacons 17

3 Universal Microbial Diagnostics Using Random DNA Probes* **19**

3.1 Introduction . . . 19

3.2 The universal microbial diagnostics platfrom . . . 21

3.3 Results . . . 25

3.3.1 Experimental proof of concept . . . 25

3.3.2 Extension by simulation . . . 29

3.4 Conclusion . . . 33

3.5 Materials and Methods . . . 34

3.5.1 Random DNA probe construction and preparation . . . 34

3.5.2 Random DNA probe design . . . 35

3.5.3 Generation of random probe characteristic curves . . . 36

3.5.4 Bacterial DNA extraction . . . 37

3.5.5 Random probe and bacterial DNA hybridization . . . 38

3.5.6 Measuring FRET ratio through fluorescence as indicator of random probe-bacteria hybridization . . . 38

3.5.7 Determining DNA hybridization affinity via SantaLucia thermodynamic models . . . 39

3.5.8 Linearity assumption considerations in UMD . . . 40

3.5.9 Receiver-operator curve (ROC) analysis . . . 40

3.5.10 Greedy probe selection . . . 41

4 Insense: Incoherent Sensor Selection for Sparse Signals† **42**

4.1 Introduction . . . 42

4.1.1 Contributions . . . 44

4.1.2 Paper organization . . . 45

4.1.3 Related work . . . 46

*This chapter is a revised version of an article, see ref. [1]

†This chapter is a revised version of an article, see ref. [2]

4.1.4 Conventional sensor selection algorithms . . . 46
4.1.5 Connections to compressive sensing . . . 46
4.2 Problem statement . . . 47
4.3 The Insense algorithm . . . 49
4.3.1 The SBS projection . . . 50
4.4 Experiments . . . 53
4.4.1 Baselines and performance metrics . . . 53
4.4.2 Unstructured synthetic datasets . . . 55
4.4.3 Highly structured synthetic datasets . . . 58
4.4.4 Real-world datasets . . . 60
4.5 Conclusions . . . 66

5 RHash: Robust Hashing via ℓ_∞-norm Distortion[‡] 67
5.1 Introduction . . . 67
5.1.1 Contributions . . . 70
5.2 Robust hashing via ℓ_∞-norm distortion . . . 71
5.2.1 Problem formulation . . . 72
5.2.2 The RHash algorithm . . . 73
5.2.3 Empirical convergence of the RHash algorithm . . . 74
5.2.4 Accelerated RHash for large-scale datasets . . . 76
5.2.5 RHash and NN preservation guarantee . . . 78
5.3 Experiments . . . 83
5.3.1 Performance metrics and datasets . . . 84
5.3.2 Small- and medium-scale experiments . . . 85
5.3.3 Large-scale experiments . . . 89
5.3.4 Discussion . . . 89

[‡]This chapter is a revised version of an under review article [3]

6 Biologically Relevant Representations of DNA Sequences Using The *K*mer2Vec Platform 91

6.1 Introduction . . . 91

6.1.1 Background . . . 93

6.2 *K*mer2Vec . . . 93

6.2.1 The *K*mer2Vec embedding . . . 94

6.2.2 *K*mer2Vec representation . . . 96

6.3 Results . . . 97

6.3.1 Learning drug-resistance concept . . . 97

6.4 Conclusion . . . 100

6.5 Materials and Methods . . . 100

6.5.1 *K*mer2Vec implementaion . . . 100

6.5.2 Drug-resistant strains . . . 101

7 Conclusion and Future Directions 102

7.1 Computational sensing . . . 102

7.2 Large-scale processing and inference . . . 105

A Mathematical formulation of the compressive sensing (CS) detection and estimation algorithms 107

Appendix A 107

B Human pathogenic bacterial species dataset 112

Appendix B 112

C UMD supplementary figures 116

Appendix C **116**

Bibliography **124**

Illustrations

2.1 Molecular Beacons (MBs) [4] Principle of Operation. The probe sequence in the loop binds spontaneously to the target sequence which is exact complement to the probe seqeunce. Adapted from [5]. 17

3.1 Schematic of Universal Microbial Diagnostics (UMD) platform. (A) Genomic DNA is extracted from a bacterial sample and thermal cycled with M random DNA probes. The genome probe binding is quantified, producing a probe-binding vector y; in this study the random probes are in the form of molecular beacons (MBs), and the DNA-probe binding is quantified by the ratio of open/hybridized to closed/non-hybridized MBs. (B) The hybridization binding level of each probe to a potentially large reference database of N bacterial genomes (B1, B2,. . . , BN) is predicted using a thermodynamic model and stored in an $M \times N$ hybridization affinity matrix . (C) Assuming K bacterial species comprises the sample, the probe-binding vector y is a sparse linear combination of the corresponding K columns of the matrix weighted by the bacterial concentrations x, i.e., $y = x + n$, where the vector n accounts for noise and modeling errors. When K is small enough and M is large enough, can be effectively inverted using techniques from compressive sensing, yielding the estimate for the microbial makeup of the sample x; in this illustration the $K = 2$ bacteria labeled B2 and B7 are present in the sample. 22

3.2 Random probe design and hybridization affinity computation process. (A) DNA sequence structure of five test random DNA probes. (B) Both strands of the bacterial genome (blue lines) are first thermodynamically aligned with the probe sequence. The sequence of the bacteria is segmented into fragments of roughly equal length ($\sim 100 - 200$ nts), each containing a significant hybridization affinity with the probe. Then, all of the bacterial fragments and probe sequences along with the experimental conditions are fed into the DNA software [6] to predict all stable probe-bacteria complexes and concentrations. These concentrations, in aggregate, determine the concentration of opened molecular beacons, which is defined as the hybridization affinity of the probes to the bacterial genome. (C) Example of a predicted probe-bacteria fragment binding with many base pair mismatches. 24

3.3 Binding patterns of five random probes correctly identifies the bacteria present in nine diverse bacterial samples. (A) Experimentally measured FRET ratios to quantify hybridization between bacterial DNA and probes 1, 2, 3, 4, and 5. (B) Hybridization affinity between DNA samples and probes, converted from FRET ratio through the probe characteristic curve fit equations (table S1 and fig. S2). (C) Heat map of normalized inner products between the experimentally obtained hybridization affinity and predicted hybridization affinities (by thermodynamic model) for nine DNA samples as a measure of the similarity of the probe measurements to the bacteria in the dataset. DNA samples are clustered into three groups: I. Exact sequence known, II. Exact sequence unknown, and III. Clinical isolates (whose exact sequence is unknown). UMD correctly recovers the diagonally highlighted bacterium (with inner product > 0.9). (D) The average receiver operating characteristic (ROC) curve of UMD in detecting nine bacteria, assuming the independence of the different experiments. Each point on the curve corresponds to a threshold value between $[-1, 1]$. UMD achieves high values of the area under the curve (AUC > 0.9). (E) Correlation of measured and simulated hybridization affinities and the normalized root-mean-square error (NRMSE) of the prediction (straight line corresponds to maximum correlation). All experiments were performed in triplicate, and the results shown here average over the trials with the error bars representing the standard error of the mean (s.e.m.). 27

3.4 Performance of UMD platform in genus-level recovery of 40 species listed as the most common human infectious genera by Centers for Disease Control and Prevention (CDC) with different number of random probes M and noise variance σ. (A) The ROC curve in detecting single bacterium ($K = 1$) with different noise levels. $\sigma_0 = 2.4 \times 10^{-8}$ M denotes the variance of the additive white Gaussian noise (AWGN) used in the simulation. This value is obtained from the experiments in Fig. 3.3.1 by calculating the propagated variance of measured FRET ratios. UMD performs more accurately with lower noise variance. The detection is almost perfect (AUC $>$ 0.95) under noise variance $\sigma = \sigma_0/5$. (B) The average ROC curve in detecting single bacterium using different number of random probes M and fixed noise variance $\sigma = \sigma_0$. The detection performance universally improves over all the 40 species by increasing the number of random probes. With 15 random probes, UMD achieves almost perfect detection performance (AUC $>$ 0.95). (C) The percentage of simulated trials, where K bacteria present in the samples were recovered correctly with zero false positives, among all possible bacteria mixtures (blue and red curves corresponding to $K = 2$ and $K = 3$ bacteria, respectively). Simulations were repeated 1000 times with randomly selected MBs, and error bars represent standard deviation (SD). (D, E) Confusion matrices illustrating the detection result of UMD using $M = 3$ and $M = 10$ probes selected by the greedy probe selection (GPS) algorithm. 31

4.1 Schematic of the sensor selection problem for sparse signals. Here, $M = 3$ sensors indexed by $\Omega = \{2, 8, 17\}$ are selected from $D = 20$ available sensors to recover a $K = 2$-sparse vector $x \in \mathbb{R}^N$, $N = 10$, from the linear system $y_\Omega = \Phi_\Omega x$. 43

4.2 Comparison of Insense against the baseline algorithms in minimizing the average coherence μ_{avg} and maximum coherence μ_{max} of the selected sensing matrix Φ_Ω from random sensing matrices with independent Gaussian (a,b), Uniform (c,d), and Bernoulli (e,f) entries ($D = N = 100$). Results are averaged over 20 trials with different random matrices. 54

4.3 Visualizations of the $M = 10$ sensing matrices Φ_Ω selected by Insense and Convex SS from a structured Uniform/Gaussian matrix. Insense selects 10/10 Gaussian rows (sensors), while Convex SS selects only 4/10 Gaussian rows. 60

4.4 Visualizations of the $M = 10$ time instances selected by Insense (blue lines), FrameSense (red lines), and Convex SS (red lines) from a matrix for the crack localization problem. Insense selects time instances that are more distributed along the time axis and variant along the intensity axis. . . 64

5.1 Comparison of the robustness and nearest neighbor (NN) preservation performance of embeddings based on minimizing the ℓ_2-norm (average distortion) vs. the ℓ_∞-norm (worst-case distortion) on a subset of the MNIST handwritten digit dataset projected onto its first two principal components. The subset consists of the 50 NN (in the 2-dimensional ambient space) of the centroid q of the cluster of "8" digits. (a) Optimal embeddings for both distortion measures computed using a grid search over the orientation of the line representing the embedding. (b) Robustness of the embedding orientations to the addition of a small amount of white Gaussian noise to each data point. This plot of the mean square error of the orientation of the ℓ_2-optimal embedding divided by the mean square error of the orientation of the ℓ_∞-optimal embedding indicates that the latter embedding is significantly more robust to perturbations in the data points. (c) Comparison of the top 5 NNs of the query point q obtained in the ambient space using the ℓ_∞- and ℓ_2-optimal embeddings (no added noise). (d) Projections of the data points onto the ℓ_∞- and ℓ_2-optimal embeddings (no added noise). 69

5.2 Comparison of the RHash and RHash-CG algorithms against several baseline binary hashing algorithms using three small-scale datasets (Q = 100). The performance of RHash-CG closely follows that of RHash, and both outperform all of the other algorithms in terms of the maximum distortion δ (superior NN distance preservation), mean average precision MAP of training samples (superior retrieval performance), and Kendall τ rank correlation coefficient (superior ranking preservation). 75

5.3 Empirical convergence behavior of the RHash algorithm. Both the maximum distortion parameter δ and the loss function $||\lambda \mathbf{v} - \mathbf{c}||_\infty$ that approximates δ gradually decrease and converge as the number of iterations increases. We see that the loss function of RHash closely matches the behavior of the actual distortion parameter in each iteration. . . 76

5.4 Hamming ranking performance comparison on three medium-scale datasets ($Q = 1000$). The top-50 neighbors are used to report MAP over a test data of same size. 78

5.5 Comparison of RHash and RHash-CG against several state-of-the-art binary hashing algorithms in preserving distances on MNIST data. (a) RHash-CG outperforms the other algorithms in minimizes the distance distortion parameter on unseen data δ_{test}. (b) RHash and RHash-CG provide better distance preservation guarantee with a small sacrifice to universality. 88

6.1 **Illustration of the *K*mer2vec embedding pipeline. A**) We start by segmenting the sequence of a large dictionary of different organisms's genome into short non-overlapping subsequences of length *K* called *K*mers. We produce a *bag-of-Kmers* representation of the N extracted *K*mers from DNA sequences. **B**) Next, we train a two layer neural network which predicts the occurrence of a *K*mer given its surrounding *K*mers. The weight matrix $\mathbf{W}$ in the trained neural network act as a embedding operator into a lower ($M \ll N$) dimensional space. **C**) We use the weight matrix $\mathbf{W}$ to embed the *K*mers into a low-dimensional space in $\mathbb{R}^M$. The new so-called *K*mer2Vec embedding of *K*mers preserve the semantics of the original DNA sequences. 95

6.2 **Illustration of the *K*mer2Vec representation of DNA sequences**. **A**, **B**) To find the *K*mer2Vec representation of a DNA sequence S, we first find the occurrence frequency of *K*mers ($K = 5$ in the illustration) and store them in a *K*mer frequency vector $\mathbf{f} \in \mathbb{R}^N$, where N counts the number of distinct *K*mers in S. **C**) We obtain the *K*mer2Vec representation of S by adding the *K*mer2Vec embedding of the *K*mers in $\mathbf{f}$ as given by $\mathbf{r} = \mathbf{f} * \mathbf{G} = \sum_{i=1}^{N} f_i \mathbf{G}_i \in \mathbb{R}^M$, where $\mathbf{G}_i$ is the *K*mer2Vec embedding of the i^{th} *K*mer. 96

6.3 **Learning drug-resistance vector**. *K*mer2Vec representation of susceptible bacterial strains (S613 and S447), their drug-resistant strains (R712 and R446), and drug-resistance vectors ($\mathbf{v}_1$ and $\mathbf{v}_2$) correspondingly in low dimensional space $\mathbb{R}^M$ ($M = 200$ in our simulations). Biological relevance of the low dimensional representation is measured by the angle between drug-resistance vectors $\theta = \measuredangle(\mathbf{v}_1, \mathbf{v}_2)$ and the norm of difference vector $\rho = \|\mathbf{v}_1 - \mathbf{v}_2\|_2/(\|\mathbf{v}_1\|_2 + \|\mathbf{v}_2\|_2)$. Smaller values of θ and ρ corresponds to more biologically relevant representations in unsupervised learning of drug-resistance from raw genome sequence. 98

6.4 **Comparison of Kmer2Vec and Kmer frequency representations in learning Daptomycin-resistance cencept.** *K*mer2Vec learns a representation with smaller angle θ and magnitude ρ (see Fig. 6.3 for exact definition) between the Daptomycin-resistance vectors of *E. faecalis* and *E. faecium* strains. *K*mer2Vec enables a better biologically relevant representation of microbial genomic sequences in terms of drug-resistance by leveraging the semantics of *K*mer in genomic data in an supervised fashion. 99

C.1 Comparison of the sloppiness of random probes to other molecular beacons. The predicted number of opened probes when the *E. coli* genome is exposed to a specific (in the same thermodynamic condition as the experiment) traditional molecular beacon (MB) (*20*), sloppy MB (*19*), and an MB created according to our random design rules. While the nucleotides in the loop were determined randomly, the percentage of opened beacons (hybridization affinity to *E. coli*) is substantially increased for the random probe MB, thanks to the choice of stem/loop length. 116

C.2 Random probes' characteristic curves. Probe characteristic curves for random probes 1–5 for determining the experimental relationship between FRET ratio and open MB concentration (hybridization affinity). The characteristic curve is fitted to the measured FRET ratio as a function of the concentration of the probe's exact complement for five random probes. . 117

C.3 Experimentally measured FRET ratios to quantify hybridization between 11 bacterial DNA samples and random probes 1–5. The error bars represent the standard error of the mean (s.e.m.). 117

C.4 Hybridization affinity between 11 bacterial DNA samples and random probes 1–5. converted from FRET ratio through the probe characteristic curve fit equations. The error bars represent the standard error of the mean (s.e.m.). 118

C.5 Detection performance of 11 bacterial samples using five random probes. (**A**) The heat map and raw values of inner product of the experimentally obtained hybridization affinity and predicted hybridization affinity (by thermodynamic model) for eleven DNA samples as a measure of the similarity of the probe measurements. For detection of the two strains marked with asterisk (*), only four random probes (Probes 1, 2, 3, and 5) were utilized. (**B**) The average receiver operating characteristic (ROC) curve of UMD in detecting eleven bacteria, assuming the independence of different test trials. Each point on the curve corresponds to a threshold value between $[-1, 1]$. UMD achieves high values of the area under the curve using experimentally obtained $\mathbf{\Phi}_{\text{exp}}$ (red line) and simulated $\mathbf{\Phi}_{\text{sim}}$ using the DNA software. 119

C.6 Comparison of the predicted concentrations of bacterial DNA with the experimentally measured values. Concentration of bacterial samples are estimated by compressive sensing recovery algorithm. 120

C.7 Performance of UMD in species-level recovery of 24 strains of Staphylococcus and 23 strains of Vibrio. (**A**) UMD's confusion matrix in identifying 24 strains of *Staphylococcus*. With $M = 11$ random probes, UMD identifies all the species of *Staphylococcus* genera with AUC $>$ 0.95. (**B**) UMD's confusion matrix in identifying 23 strains of *Vibrio*. With $M = 18$ random probes, UMD identifies all the species of *Vibrio* with AUC $>$ 0.95. 120

C.8 Performance of UMD at identifying pathogens in genus-level using 15 GPS probes. The UMD detection performance increases with a large number of GPS probes. With M = 15 probes, UMD alleviates all the potential false positives and provides a more noise-robust identification of pathogens at the genus-level. 121

C.9 Performance of UMD in identifying eight pathogenic and one nonpathogenic E.coli strains using GPS probes. GPS selects 6 UMD probes that differentiate between eight pathogenic and one nonpathogenic *E.coli* strains *in silico*. 122

C.10 Performance of UMD in identifying the composition of several complex samples. We simulate the number of required random probes M to identify the composition of complex bacterial samples with 100% accuracy using UMD. The complexity of each sample is measured by the number of present bacterial species K among a total of N = 1500 bacterial genera. To obtain each data point in the curve, we randomly selected K bacterial genera from the database and created a sample by mixing K sample species from each genera with equal concentrations. We used UMD to recover the composition of the mixture using random probes. We repeated this same experiment for 1000 trials and reported the minimum number of probes that recovered all 1000 mixtures accurately. UMD requires orders of magnitudes less number of random probes than the size of the database N (i.e., number of probes typically required by conventional methods) to recover complex samples containing mixtures of K species (with equal concentrations) among N = 1500 genera. Furthermore, number of required UMD probes closely ($R^2 = 0.98$) follows the number of probes predicted by the compressive sensing theory $M = ck\log(N/K)$ with $c = 2.94$. 123

Tables

4.1 Comparison of Insense against the baseline algorithms on selecting $M = 10$ rows from a structured Identity/Gaussian Φ. Insense selects the set of sensors with the smallest μ_{avg} and achieves the best BP recovery performance. 57

4.2 Comparison of Insense against the baseline algorithms on selecting $M = 10$ rows from a structured Uniform/Gaussian Φ. Insense selects the set of sensors with the smallest μ_{avg} and achieves the best BP recovery performance. 61

4.3 Comparison of Insense against the baseline algorithms on selecting M DNA probes to identify pathogenic samples containing K bacterial organisms. Insense selects the sets of DNA probes that achieve the best pathogen identification performance. 62

4.4 Comparison of Insense against the baseline algorithms on sampling M time instances that best localize cracks on a plane in a structural health monitoring problem. Insense selects the set of sensors with the smallest μ_{avg} and achieves the best damage localization performance. 64

5.1 Comparison of RHash-CG against several baseline binary hashing algorithms on large-scale *MNIST+rotation* dataset with over 5 billion secants $|\mathcal{S}(\mathcal{X})|$. We tabulate the Hamming ranking performance in terms of mean average precision MAP for different sizes of the dataset. All training times are in seconds. 86

C.1 The fitted parameters to the probes' characteristic curves. Characteristic curve parameters a, b, and n for five test random probes, and the associated curve fit performance criteria, R^2 and root mean square error (RMSE) are tabulated. Parameters were fit to the following curve: $\text{FRET}(c) = \text{FRET}_0 + \frac{a}{(1+b(10^{-6}-c)^{-n})}$ (*22*). 118

Chapter 1

Introduction

We live in the $post-genomic$ era, when DNA sequence data is growing exponentially. The cost for sequencing the Human genome has decreased by a factor of million $\times 10^6$ since 2001. The availability of genomic data is opening new doors for machine learning and signal processing tools, algorithms, and theories to better understand the biological phenomenon and engineer biological sensing instruments. Today, the complete sequencing of genome offers the scope to apply machine learning intelligence to the processing and acquisition of genomic data. This tribute of artificial intelligence in the genomic era will have impacts in a variety of fields vital to human health: personalized medicine, cancer therapy, diagnostics, antibiotic resistance, among others.

After setting few notations and backgrounds in chapter 2, we start this thesis, in chapter 3, focusing on rapid microbial diagnostic challenge problem which shows the power of intelligent signal processing and machine learning designs in biological sensing problems. Early identification of pathogens is essential for limiting development of therapy-resistant pathogens and mitigating infectious disease outbreaks. Most bacterial detection schemes use target-specific probes to differentiate pathogen species, creating time and cost inefficiencies in identifying newly discovered organisms. In chapter 3, we present a novel universal microbial diagnostics (UMD) platform to screen for microbial organisms in an infectious sample, using a small number of random DNA probes that are agnostic to the target DNA sequences. Our platform leverages the theory of sparse signal recovery (compressive sensing) to identify the composition of a microbial sample that potentially contains

novel or mutant species. We validated the UMD platform in vitro using five random probes to recover 11 pathogenic bacteria. We further demonstrated *in silico* that UMD can be generalized to screen for common human pathogens in different taxonomy levels. UMD's unorthodox sensing approach opens the door to more efficient and universal molecular diagnostics.

The UMD platform emphasizes the use of random DNA probes to identify pathogens. Given a large dictionary of random DNA probes, however, there exist a smaller set of probes that significantly outperform the average performance of random probes. In chapter 4, we focus on the problem of *sensor selection* to identify this optimal set of probes for sparse recovery problems. Sensor selection refers to the problem of intelligently selecting a small subset of a collection of available sensors to reduce the sensing cost while preserving signal acquisition performance. The majority of sensor selection algorithms find the subset of sensors that best recovers an arbitrary signal from a number of linear measurements that is larger than the dimension of the signal. In this chapter, we develop a new sensor selection algorithm for sparse (or near sparse) signals that finds a subset of sensors that best recovers such signals from a number of measurements that is much smaller than the dimension of the signal. Existing sensor selection algorithms cannot be applied in such situations. Our proposed Incoherent Sensor Selection (Insense) algorithm minimizes a coherence-based cost function that is adapted from recent results in sparse recovery theory. Using six datasets, including two real-world datasets on microbial diagnostics and structural health monitoring, we demonstrate the superior performance of Insense for sparse-signal sensor selection.

In many point-of-care diagnostic applications in resource-limited enviroments, the data acquired from sensors can not be analyzed on the sensing device. The sensing device is too small to do any sort of processing in the field. The acquired has to be reliably and robustly transmitted to a computing server, maybe in the cloud, to do the analysis. In the next chapter

of the thesis we focus on the problem of fast nearest neighbor retrieval in a resource limited environment where the data has to be embedded or *hashed* into a lower-dimensional space for rapid data retrieval. Hashing is an important tool in large-scale machine learning. Unfortunately, current data-dependent hashing algorithms are not robust to small perturbations of the data points, which has a negative effect on the performance of nearest neighbor (NN) search. The problem stems from minimizing the ℓ_2-norm, average distortion among pairs of points to find the hash function. Inspired by recent progress in robust optimization, we develop a novel hashing algorithm, dubbed RHash, that instead minimizes the worst-case distortion among pairs of points using ℓ_∞-norm minimization. We develop practical and efficient implementations of RHash based on the alternating direction method of multipliers (ADMM) framework and column generation that scale well to large datasets. A range of experimental evaluations demonstrate the superiority of RHash over ten state-of-the-art binary hashing schemes.

Continuing our journey from sensing, processing, and analysis of genomic data, we develop a new platform to represent DNA sequences In chapter 6. The new platform is based on a new way of looking at the building blocks of DNA sequences, i.e., *K*mers. *K*mers are length-*K* subsequences of a DNA sequence. They provide a rich representation of the DNA sequence and due to their flexibility and computational advantages, they are a key part in many important algorithms in bioinformatics such as sequence alignment and assembly. The *K*mer's frequency representation of DNA sequences, however, ignores the *semantics* buried in the sequence; it loses the biological knowledge related to the ordering of *K*mers within the sequence. In this paper, we borrow ideas from the traditional Word2Vec algorithm in natural language processing and develope a new platform, dubbed as *K*mer2Vec, that learns a new representation of DNA sequences based on the semantics of their *K*mers. Trained on 43 microbial organisms' genome sequences, we demonstrate that our semantic-

aware *K*mer2Vec representation learns biological concepts, such as drug-resistance, solely based on the raw microbial genome sequences with no prior microbiology knowledge. Our results suggests that *K*mer2Vec representaion is a more powerful alternative for *K*mer representation that respects the semantics buried in the genomic datasets. We finish with concluding remarks in chapter 7.

Chapter 2

Background

2.1 Preliminaries

2.1.1 Notation

We provide a brief summary of the notation used in this thesis. Signals are real valued vectors defined over domains that are either continuous or discrete. In general, lower case letters (e.g., $\mathbf{x}$, $\mathbf{y}$) will be used to denote discrete column signals; upper case boldface letters (e.g., $\mathbf{\Phi}$, $\mathbf{D}$, $\mathbf{P}$) will be used to denote matrices; and unbolded italic letters (e.g, M, N, λ) will be used to denote scalar quantities. Calligraphic letters (e.g., $\mathcal{S}$, $\mathcal{P}$) will be used to denote sets or set operators. Four upper case letters will be used to denote four DNA nucleobases Adenine (A), Cytosine (C), Guanine (G), and Thymine (T).

2.1.2 Vector Norm

In linear algebra, a norm is a function that assigns a strictly real positive number to each vector in a $vector\ space$. For example, for a finite-length real signal $\mathbf{x}$ of size N, which can be viewed as an element of N-dimensional vector space $\mathbb{R}^N$, i.e., $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$, well known $\ell_p - norms$ are defined as,

$$||\mathbf{x}||_p = \begin{cases} (\sum_{i=1}^{N} |x_i|^p)^{1/p}, & p \in (0, \infty); \\ \max_{i=1,2,\ldots,N} |x_i|, & p = \infty; \\ \sum_{i=1}^{N} \mathbf{1}_{x(i)\neq 0}, & p = 0, \end{cases} \tag{2.1}$$

where $\mathbf{1}$ denotes the indicator function. We have to note that the ℓ_p-norm does not satisfy the criteria for a well-defined norm when $p < 1$. A very common and useful norm is the ℓ_2-norm which is also commonly known as $Euclidian$ norm. It is easy to see that the ℓ_2-norm can be written as the sauce root of the inner product of the vector with itself, i.e., $||\mathbf{x}||_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\sum_{i=1}^{N} \mathbf{x}_i^2}$. The ℓ_2-norm is particularly useful in characterizing the *mean square error* (MSE) between pairs of signals; the MSE between $\mathbf{x}_1$,$\mathbf{x}_2 \in \mathbb{R}^N$ is defined as

$$\frac{1}{N}||\mathbf{x}_1 - \mathbf{x}_2||_2^2. \tag{2.2}$$

The *sum of absolute differences* (SAD) is another common error metric based on the ℓ_1-norm. Also known as $Manhattan\ distance$, SAD between $\mathbf{x}_1$,$\mathbf{x}_2 \in \mathbb{R}^N$ is defined as

$$||\mathbf{x}_1 - \mathbf{x}_2||_1. \tag{2.3}$$

2.1.3 Sparse signals

Given a vector space $\mathcal{F}$, a $dictionary$ is simply a collection of vectors in $\mathcal{F}$ (also called *atoms*) whose linear superpositions can be used to represent signals of interest. For the signals living in N-dimensional space, any dictionary can be equivalently represented as a matrix $\mathbf{D} \in \mathbb{R}^{N \times D}$, with the columns of $\mathbf{D}$ representing the atoms. An *orthonormall basis* (ONB), or simply, a basis is a special type of dictionary with $D = N$ normalized and orthogonal columns. Given a signal $\mathbf{x} \in \mathbb{R}^N$ and a basis $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$, we can $expand$ $\mathbf{x}$ in terms of its basis coefficients $\boldsymbol{\alpha}$, i.e., $\mathbf{x} = \mathbf{\Psi}^{-1}\boldsymbol{\alpha} = \mathbf{\Psi}^T\boldsymbol{\alpha}$. Common example of orthonormal bases is the *canonical basis* (where $\mathbf{\Psi}$ equals the identity matrix and $\boldsymbol{\alpha} = \mathbf{x}$). Given a general dictionary $\mathbf{D}$, we are often interested in the set of signals that can be written as the linear combination of only a few elements of $\mathbf{D}$. Such signals are known as $sparse$ signals. Formally, a signal $\mathbf{x} \in \mathbb{R}^N$ is said to be k-$sparse$ in $\mathbf{D}$ if $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$ and no more than K coefficients of $\boldsymbol{\alpha}$ are nonzero. The $support$ of $\mathbf{x}$, $\mathcal{S}(\mathbf{x})$, is defined as the

set of indices corresponding to nonzero entries of $\boldsymbol{\alpha}$. The $sparsity$ of $\mathbf{x}$ is the number of nonzero, or the ℓ_0-norm, of its dictionary representation $\boldsymbol{\alpha}$ (denotes as $||\boldsymbol{\alpha}||_0$). An elegant feature of a K-sprase signal $\mathbf{x} \in \mathbb{R}^N$ is that is can be concisely represented using $2K$ pieces of information, corresponding to the *locations* and the *values* of the nonzero coefficients of $\boldsymbol{\alpha}$. In the regime where $K \ll N$, it is often vastly beneficial to store, sense, and process signals entirely in the space of coefficients.

2.2 Compressive sensing

2.2.1 Compressive sensing theory

Compressive Sensing (CS) is an alternative to Shannon-Nyquist sampling theory for the acquisition of sparse or compressible signals in an N-dimentional space. Instead of taking many periodic samples, CS measures inner products with $M \ll N$ random vectors and then recovers the signal via a sparsity-seeking optimization or greedy algorithm. The main result of CS, introduced first by Candes, Romberg, and Tao [7] and Donoho [8], is that robust signal recovery is possible from $M = \mathcal{O}(K \log(N/K))$ measurements. More rigorously, CS obtains a random linear set of measurements, $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}$, where $\mathbf{x}$ is the $N \times 1$ sparse signal we aim to sense, $\mathbf{y}$ is an $M \times 1$ measurement vector, and the $measurement\ matrix$ $\boldsymbol{\Phi}$ is an $M \times N$ matrix. In the presence of measurement noise, the model becomes, $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x} + \mathbf{n}$ where $\mathbf{n}$ is the noise vector. Since $M \ll N$ simply inverting the matrix $\boldsymbol{\Phi}$ is impossible in this case, since it has many more columns than rows. The recovery of $\mathbf{x}$ from $\mathbf{y}$ is known to be ill-posed using the underdetermined system. Fortunately the sparsity of vector $\mathbf{x}$ makes the problem invertible under some assumption imposed by CS theory.

In order to apply the standard compressive sensing theory, the columns of $\boldsymbol{\Phi}$ should satisfy the so-called Restricted Isometry Property (RIP) [9]. It has been shown that a matrix

satisfies the RIP if its columns are sufficiently $incoherent$ [10], i.e., when the largest normalized inner product between any two columns of $\mathbf{\Phi}$, $\mu = \max_{\forall i,j:i\neq j} \frac{\phi_i^T \phi_j}{||\phi_i||||\phi_j||}$, known as the $coherence$, is bounded above by a small constant. More specifically, it has been shown [10] that $\mu < \frac{1}{2K-1}$ is a sufficient condition to exactly recover a K-sparse signal with only $M = \mathcal{O}(K \log(N/K))$ measurements in the noise-free scenario (i.e. when $\mathbf{n} = 0$). In the presence of noise, the same sufficiency bound holds if the magnitudes of the non-zero elements of $\mathbf{x}$ are sufficiently large compare to the noise variance [11]. When these conditions hold, we can both detect the presence of bacteria and estimate their concentration using standard CS signal recovery algorithms.

2.2.2 Compressive sensing for non-negative signals

In a special class of problems where both the hybridization matrix $\mathbf{\Phi}$ and the sparse concentration vector $\mathbf{x}$ are non-negative a more optimistic recovery bound has been proven to hold. In particular [12] defines an alternative notion of incoherence that improves the recovery guaranty. To state the new recovery bound it is essential to introduce the alternative notion for coherence known as one-sided coherence. For an arbitrary matrix $\mathbf{\Phi}$, the one-sided coherence is defined as $\rho(\mathbf{\Phi}) = \max_{\forall i,j:i\neq j} \frac{\phi_i^T \phi_j}{||\phi_i||_2^2}$. It is shown in [12] that, for a non-negative matrix $\mathbf{\Phi}$, if a non-negative K-sparse solution exists such that $\rho(\mathbf{PD}) < \frac{1}{2K-1}$, then the solution is unique and CS recovery algorithms can find it. Here $\mathbf{D}$ is defined as the column ℓ_1 (sum) normalized matrix of $\mathbf{\Phi}$, and $\mathbf{P}$ is column mean subtraction operator [12]. While the one-sided coherence recovery bound is based on the better-conditioned matrix $\mathbf{PD}$, it remains pessimistic, and better recovery performance is typically achieved empirically in practice.

2.2.3 Greedy recovery algorithms

The concentration vector $\mathbf{x}$ is recovered from the measurement vector $\mathbf{y}$ via a sparsity-penalized optimization of the form,

$$\min_{\mathbf{x}}||\mathbf{x}||_0, \text{ subject to } ||\mathbf{y} - \mathbf{\Phi x}||_2 < \sigma.$$

Here $||\mathbf{x}||_0$, known as ℓ_0 norm, counts the number of non-zero values in the vector $\mathbf{x}$, and σ estimates the noise standard deviation. While this optimization problem has exponential complexity, a variety of different greedy algorithms have been developed to solve it approximately. Orthogonal Matching Pursuit (OMP) [13] is an iterative greedy algorithm that, at each step, selects the column of $\mathbf{\Phi}$ that is most correlated with a residual vector and then updates the residual by by projecting the measurements vector $\mathbf{y}$ onto the linear subspace spanned by the columns that already been selected. The primary advantages of OMP are its simplicity and fast convergence. Moreover, if the sparsity level K (the number of bacteria is known, then the algorithm can use it as the stopping criterion.

In non-negative CS setting we can leverage the fact that both the hybridization affinity matrix $\mathbf{\Phi}$ and the sparse concentration vector $\mathbf{x}$ are non-negative to improve the performance of OMP. We utilize a variant of OMP described in [12] that is adapted to recover non-negative sparse solutions from non-negative sensing matrix $\mathbf{\Phi}$. Instead of directly working on $\mathbf{\Phi}$, this algorithm operates on the canonical matrix **PD**, where **D** is the column ℓ_1 (sum) normalized version of matrix $\mathbf{\Phi}$ defined as: $\mathbf{D} = \mathbf{\Phi W^{-1}}$, where **W** is an $M \times M$ diagonal matrix containing the column sums of the hybridization matrix $\mathbf{\Phi}$ and **P** is the pre-conditioner matrix and can be chosen as any invertible $N \times N$ matrix. In the case of positive matrix **D** an efficient preconditioning can be obtained just by subtracting the weighted mean of each column of **D**: $\mathbf{PD} = (I - \frac{1-\epsilon}{\mathbf{N}}\mathbf{E})\mathbf{D}$, where **E** is $N \times N$ matrix of ones, **I** is the identity matrix, and $0 < \epsilon < 1$ is a weighing constant to make the **P** matrix

invertible. Working with the preconditioned matrix **PD** does not change the solution of the problem [12]; however it significantly improves the OMP algorithm behavior and performance guarantees. This OMP algorithm variant from [12] is constructed as follows:

Algorithm 1: Non-negative OMP

Data: **D**, **y**

Result: $\mathbf{x}^i$

Initialization;

The temporary solution: $\mathbf{x}_i = \mathbf{0}$; The temporary residual: $\mathbf{r}^i = \mathbf{y}$; The temporary solution support $\mathcal{S}^i = support\{\mathbf{x}^i\} = \emptyset$

while $||\mathbf{r}||_2^2 < T$ **do**

Sweep: $\forall j \in [1, 2, \ldots, N]$: find $\epsilon(j) = \min_{x \geq 0} ||\mathbf{D}_j \mathbf{x}_j - \mathbf{r}||_2^2$;

Update support: find j_0 such that $\forall j \in \mathcal{S}^i$, $\epsilon(j_0) \leq \epsilon(j)$, and update $\mathcal{S}^i = \mathcal{S}^{i-1} \bigcup \{j_0\}$;

Update solution: compute $x^i : x^i = \min_{\mathbf{x}} ||\mathbf{D}\mathbf{x} - \mathbf{y}||_2^2$ subject to $support\{\mathbf{x}^i\} = \mathcal{S}^i$ and $\mathbf{x} \geq 0$;

Update residual: compute $\mathbf{r}^i = \mathbf{y} - \mathbf{D}\mathbf{x}^i$;

end

2.2.4 Basis pursuit denoising

While greedy algorithms such as OMP attempt to solve the exact ℓ_0 norm penalized problem, another important class of CS recovery algorithm relax the problem by replacing the ℓ_0 norm with the more tractable ℓ_1 norm:

$$\min_{\mathbf{x}} ||\mathbf{x}||_1, \text{ subject to} ||\mathbf{y} - \mathbf{\Phi}\mathbf{x}||_2 < \sigma.$$

Here $||\mathbf{x}||_1$ sums the absolute value of the entries in the vector $\mathbf{x}$. The ℓ_1 norm penalized optimization problem above is convex and reduces to a linear program known as Basis Pursuit Denoising (BPDN) [9]. Surprisingly the ℓ_1 norm penalized problem, under the same sufficient conditions on mutual coherence state above, can exactly recover a K-sparse signal from the noisy measurements $\mathbf{y}$ with $M = \mathcal{O}(K \log(N/K))$ measurements.

For the special case where the sparse vector $\mathbf{x}$ is also non-negative, we can improve the performance of BPDN by adding a positivity constraint,

$$\min_{\mathbf{x}} ||\mathbf{x}||_1, \text{ subject to} ||\mathbf{y} - \mathbf{\Phi}\mathbf{x}||_2 < \sigma, \text{ and } \mathbf{x} \geq 0.$$

By replacing the objective function with the squared norm $||\mathbf{x}||_1^2$ and bringing the constraint into the cost function using a Lagrangian multiplier $\lambda > 0$, we have,

$$\min_{\mathbf{x}} ||\mathbf{x}||_1^2, \text{ subject to} ||\mathbf{y} - \mathbf{\Phi}\mathbf{x}||_2 < \sigma, \text{ and } \mathbf{x} \geq 0.$$

Since the ℓ_1 norm of a non-negative valued vector $\mathbf{x}$ is $||\mathbf{x}|| = \sum_{i=1}^{N} \mathbf{x}_i$, we can reformulate the above problems as a non-negative least squares problem

$$\min_{\mathbf{x}} ||\tilde{\mathbf{y}} - \tilde{\mathbf{\Phi}}\mathbf{x}||_2^2, \text{ subject to } \mathbf{x} \geq 0,$$

that can be solved using standard methods (e.g., the lsqnonneg command in Matlab). Here $\tilde{\mathbf{y}} = \begin{bmatrix} \sqrt{\lambda}\mathbf{y} \\ 0 \end{bmatrix}$, and $\tilde{\mathbf{\Phi}} = \begin{bmatrix} \sqrt{\lambda}\mathbf{\Phi} \\ 1 \ldots 1 \end{bmatrix}$, where the last column penalizes the ℓ_1 norm.

2.3 Whole genomic sequence analysis tools

The genome of an organism refers to its collective genetic material consisting of DNA molecules amino acid molecules in the nucleus of every cell. Incredibly, DNA naturally lends itself to a discrete structural description of its subunits, with respect to an ordering that is of great biological consequence. Each molecule of DNA is comprised of a linear

sequence of discrete nucleotides a sequence of nucleic bases against a sugar-phosphate backbone. The permutation of nucleotides in the genome is a blueprint for instructions governing every aspect of an organisms physical and mental being, so in fact this is a very useful representation to understand. It is no coincidence that the discrete nucleotide sequence of the genome has become synonymous with the genome itself. The transcriptome and proteome of an organism (comprised of its RNA and expressed proteins, respectively) also have similar discrete signal representations.

Traditionally, deriving a complete, discrete sample representation of the genome has been a challenge due to its minuscule size and the problems associated with making such molecular-level measurements. Instead, for decades there have existed several technologies to determine shorter, partial representations of the genome by indirect means, i.e. by observing the hybridization of short sections of DNA strands with their complements, and then inferring the original DNA strand composition from it. Some such technologies are Southern blots, microarrays, and PCR. In fact, the Southern blot, named after its inventor, Edwin Southern, was so influential that it inspired later techniques named Northern, Eastern, Western and Southwestern blots for RNA and protein detection.

Dramatic change occurred with the advent of the first sequencing method, Sanger sequencing. Today, next-generation sequencing (NGS) technology, buoyed by rapid advances in optics, nanotechnology and computer science, has made it possible to derive a complete, high-fidelity sampling of the genome. Instead of observing and measuring the whole-sequence hybridization of short DNA sections, we record base- by-base hybridization of DNA sections, using fast, redundant measurements, followed by a computational phase involving the alignment and assembly of those sections. The genomics world is at an exciting stage where several types of sequencing technologies have not only been developed, but are being further perfected every day while providing us with freely flowing genomic

data. For the first time, the genetic material of every organism has the potential to become a known quantity. Unlike analog-to-digital signal converters, technologies to sense genetic material are expensive. Sequencing may be the gold standard for the rigorous assessment of genomic material, but its cost is still extremely prohibitive for practical non-research use. Until the price of sequencing falls from bench-levels to bedside-levels, it is critical to focus on other sampling methods like microarrays, molecular beacons and microfluidic devices that are readily available and less expensive for practical purposes.

Until then, we can take advantage of the fact that an organisms DNA is almost unvarying over its lifetime, so it only needs to be sequenced once to be known. We can leverage our knowledge of fully-sequenced genomes to make informed pronouncements on the data that partial-sampling based technologies provide us. The technology that we focus on re-engineering is molecular beacons. We discuss DNA hybridization models in this section before delving into the molecular beacon technologies in the next section.

2.3.1 DNA thermodynamic hybridization model

Practically all present-day genomic sampling tools are based on complementary hybridization. The DNA molecule of interest binds with its complement; this binding is measured, and the original DNA molecules nucleic acid composition is inferred from it. Tools like molecular beacons use the complementary binding between probe and target DNA strands, and then measure emitted fluorescence of each bound section. Therefore it is especially important to establish a model to predict the binding strength between a probe and the target is intended for, in order to design them appropriately. Sequencing-by-synthesis technologies, as the vast majority of NGS technologies are, also depend on complementary hybridization. However they use single base binding and fluorescence to ascertain the DNA sequence one base at a time, so there is no probe design component and hence no critical dependence

on a hybridization model. Single base binding is essentially binary either it is bound and fluoresces, or not. There are several factors that influence the hybridization affinity of a DNA fragment with another (e.g. Smith-Waterman, etc.). We discuss a thermodynamic based approach here: the SantaLucia Nucleic Acid Thermodynamic model, which is based on the empirical and theoretical predictions of the bound and unbound free energies of any pair of nucleotides [12].

Our goal is to estimate the probe-target hybridization intensities at each probe spot, given probe and target DNA sequences. We aim to use a thermodynamic model for DNA-DNA hybridization to give us a physically realistic model for hybridization affinities. For instance an A-T bond formation is a double covalent bond between hydrogen atoms, which releases relatively less free energy since it is relatively less stable than a C-G bond, which is is a triple covalent bond between corresponding hydrogen atoms on each base. More complicated to quantify are the free energies when there are base mismatches.

Unlike many other models, this model, also incorporates thermodynamic parameters for mis-hybridizations between two DNA sequences (accredited to SantaLucia [12]). The frequent occurrence of base mismatches in probe-target hybridizations during the probe design phase makes it critical that we estimate affinities in the face of these mis-hybridizations as accurately as possible. Using Smith-Waterman sequence alignments to do so is inexact for cases of mismatches, since the algorithm is highly dependent on the type of penalty that gaps are assigned by the alignment scoring matrix in use. There is no consensus on which penalty values most accurately reflect hybridization reality. The SantaLucia thermodynamic model is commercially available from the company DNA Software, Inc. We purchased two component software packages: (1) ThermoBlast, which performs fast alignments of sequences against large genome databases to discover thermodynamically stable hybridizations, and (2) Visual OMP DE, which can simulate hybridization

experiments with detailed solution conditions through scripts launched from the command line, and generate results for melting temperatures (Tm), Gibbs free energy (ΔG), and the percentage-based concentration of each resultant species post-experiment. There is also the capability to visually generate the secondary structure for each monomer, homodimer and heterodimer species formed from the constituent probes and target fragments. We follow a two-step procedure in determining the hybridization intensities of a given set of probes against a given dictionary of potential targets. First, we run each probe against each double stranded target genome, using ThermoBlast, to discover all possible target fragments at which there is significant alignment. Second, we use Visual OMP DE to run a script to simulate each hybridization experiment between a single probe and a single target, using these target fragments to simulate the target. Every experiment simulation script contains information on: the probe sequence, the target fragments from ThermoBlast it was shown to hybridize with, and conditions for the experiment. Experimental conditions included probe and target concentrations, assay temperature, hybridization cocktail composition (Mg++, Glycerol, DMSO, Formamide, TMAC, Betaine concentrations), and pH. This procedure is repeated for each probe-target pair. The hybridization results of each experiment in Visual OMP DE produces a data file, which is parsed for the concentrations of resultant species at the end of the hybridization. In each data file we use the percentage of probe-target heterodimer structures formed, i.e. the percentage of target fragments that are bound to probes, as an estimate for the hybridization affinity of that probe-target pair.

2.4 Molecular beacons

2.4.1 Principle of operation

Molecular Beacons (MBs) [14] are single-stranded, free-floating oligonucleotide hybridization probes in a stem-loop structure with two fluorescent dyes, Cy3 and Cy5, bound to the $5'$ and $3'$ ends. When a target sequence presents a more stable binding with the loop section than the binding between its two stem ends, MBs open and report the presence of a target molecule through the fluorescence resonance energy transfer (FRET) between the Cy3 and Cy5 molecules (see Figure 2.4.1). The loop portion of a conventional MB is a probe sequence that is complementary to a predetermined sequence in a target nucleic acid. The stem is formed by the annealing of two complementary arm sequences that are on either side of the probe sequence. The stem keeps the fluorescent dyes in close proximity to each other, casuing the fluorescence of the fluorophore to be quenched by FRET. When the probe encounters a target molecule, it forms a hybrid that is longer and more stable than the hybrid formed by the arm sequences. Since nucleic acid double helics are relatively rigid, formation of a probe-target hybrid precludes the simultaneous existence of a hybrid formed by arm sequences. Therefore, the probes undergoes a conformational change that forces the arm sequences apart and causees the fluorophore and quencher to move away and fluoresces when illuminated by ultraviolet light.

2.4.2 Advantages for microbial detection

There are many advantages to using MBs for bacterial detection. First, they are free-floating rather than immobilized to a surface (as in microarray) and therefore have a great probablity of making contact with a target molecule, resulting in faster reaction kinetics. Second, the probe-target hybrids need not be isolated from their surplus probes prior to analysis, making

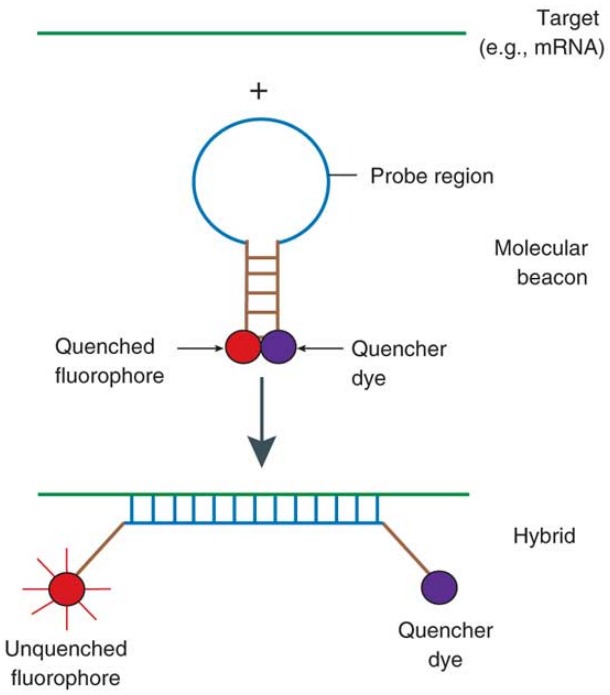

Figure 2.1 : Molecular Beacons (MBs) [4] Principle of Operation. The probe sequence in the loop binds spontaneously to the target sequence which is exact complement to the probe seqeunce. Adapted from [5].

them more amenable for field use. Another advantage of MBs is their relatively high signal-to-background ratio, providing higher sensitivity of detection [15]. Upon hybridization to its target, a well designed MB can generate as high as 200-fold fluorescence enhancement under optimal conditions [4]. This provides the MBs with a significant advantage over other flourescent probes in ultra-sensitive analysis.

2.4.3 Sloppy molecular meacons

A traditional MB only reacts to a target sequence that is compete complement to the loop section of the probe. This require upwards of 100 different species-specific probes in many PCR assay tubes for a comprehensive screening procedure. A different approach was introduced in [16] as a screening technique for bacterial identification that requires only a sin-

gle gene amplification assay containing a set of four differently colored molecular beacon probes of a unique design. After the completion of amplification, the molecular beacons in the reaction tube are hybridized to the amplicons at a relatively low temperature. The temperature is then slowly raised to determine, from the consequent changes in fluorescence intensity of each molecular beacon in the set, the temperature at which each of the four probe-target hybrids melts apart (Tm). The resulting set of four Tm values uniquely identifies the species present in the sample.

The key feature that enables this design to react to more that one target organism is the non specificity introduced in the probes. Unlike sequence-specific molecular beacons [4], which possess short probe sequences (18 to 26 nucleotides long) that form probe-target hybrids only with perfectly complementary, or nearly perfectly complementary, target sequences, these $sloppy$ molecular beacon probes are designed to form probe-target hybrids with the amplicons generated from all of the species that we wish to identify. In order to enable this property, the probe sequences in these molecular beacons are about 40 nucleotides long. Consequently, they form probe-target hybrids even if the duplexes possess a substantial number of mismatched base pairs.

Sloppy MBs are among the first novel practical approaches that enable non specific sensing of a group of bacteria. The design, however, is based on probes that are meticulously designed to target the 16S rDNA in different bacteria genomes and therefore inevitably have the limitation of other gene (marker)-specific approaches. The most important of them is probably that they can not be used to identify unknown bacterial species whose 16S is not available (sequenced) yet. The applicability of SMBs to other microbial organims (such as viruses) is also not investigated.

Chapter 3

Universal Microbial Diagnostics Using Random DNA Probes*

3.1 Introduction

The accurate, efficient, and rapid identification of microbial organisms such as bacteria and viruses is of mounting importance in the fields of health care, environmental monitoring, defense, and beyond [17, 18, 19]. Sepsis from bacterial infection is currently the 11^{th} leading cause of death in the United States, and the mortality rate of bloodstream infections is high (14-50 %) [20, 21, 22]. Conventional strategies for microbial detection are based on microbe-specific genomic or proteomic markers and protocols. Polymerase chain reaction (PCR)-based approaches rely on the binding of specific capture probes with unique genomic identifiers, such as the 16S ribosomal DNA subunit in bacteria. While these methods show promise as highly specific tools for microbial identification [23, 22], they have limitations in clinical, industrial, and defense settings [24, 25]. In the case of an epidemic, the detection of a newly mutated species using current PCR methods would require entirely new capture probes to be manufactured, introducing additional costs and delays. For bacterial detection, blood cultures typically require 48 to 72 hours to produce reliable results [26, 27, 28, 29]. During this waiting period, administration of broad-spectrum antibiotics breeds further threats of bacterial resistance and missed coverage [30]. DNA microarrays also require many target-specific probes to detect multiple pathogens and lie dormant

*This chapter is a revised version of an article, see ref. [1]

against unknown organisms. Whole genome sequencing (WGS), currently the most complete and accurate technique, is not yet conducive to point-of-care diagnostics; it requires millions of expensive sequencing reads to assemble or align with genomic identifiers. It follows that there is a critical need for a new means of microbial detection: a universal (i.e., works for bacteria outside of the target library), inexpensive (i.e., requires minimal resources for acquisition such as DNA probes and sequencing reads, etc.), and rapid sensing platform capable of identifying known and novel species with high phylogenetic power. In this paper, we report on the design and validation of a new microbial diagnostic platform that satisfies the above desiderata. In common with microarrays and PCR-based techniques, our Universal Microbial Diagnostics (UMD) platform exposes a microbial sample (which may contain more than one genus/species) to a collection of DNA probes. In sharp contrast to conventional methods, however, the probes are randomly generated (and hence target-agnostic) permutations of nucleotides (nts) that freely hybridize to different spots and to different extents on different bacterial genomes. By measuring the degree to which the sample hybridizes with the collection of random probes, we set up a statistical inverse problem to detect the presence and estimate the concentrations of the various bacteria in the sample. Using signal recovery techniques from the recently developed theory of compressive sensing [8, 9], we show below that it is possible to stably solve this inverse problem even when the number of probes is significantly smaller than the size of the library of possible bacteria of interest. Thanks to the random structure of the probes, and the variabilities that bacterial organisms exhibit in their genomes, UMD is universal, inexpensive, rapid, and phylogenetically informative (random probes bind to arbitrary spots on the genome). Moreover, due to the universal nature of its probe design, UMD can classify not only known organisms but also novel mutants with their closest known relatives.

3.2 The universal microbial diagnostics platfrom

In UMD Fig. 3.2A, the genomic DNA of an infectious sample is extracted and exposed to a small number M of DNA probes, which hybridize to the genomic DNA at various locations; this hybridization is experimentally quantified, producing a probe-binding (or hybridization) vector y whose entries correspond to the hybridization binding level of each probe with the microbial sample.

A priori, the hybridization binding level of each probe to a reference database of N bacterial genomes is obtained and stored in an $M \times N$ hybridization affinity matrix Fig. 3.2B. The hybridization affinity matrix can be measured either experimentally in vitro or predicted computationally in silico. To speed up the probe design and prove the concept of UMD, here we predict the affinity matrix using a thermodynamic model in silico. To compute the entry ϕ_{ij} in the matrix , the hybridization binding level of probe i to genome j, we first perform a rapid thermodynamic alignment of the sequence of the probe to the sequence of the genome using the alignment model described in [6]. Next, we extract sequence fragments from the genome sequence, which contain a significant hybridization affinity with the probe sequence. The fragment-probe mixture is then fed into a thermodynamics-based hybridization model [6]. This model predicts all possible stable probe-bacteria fragment bindings along with their resulting concentrations for a given set of experimental conditions (Fig. 3.2, B and C). The overall hybridization affinity ϕ_{ij} is computed by summing the concentrations of all predicted and stable probe-fragment bindings for a unit concentration of bacterial genome.

Due to an excess concentration of probes as compared to sample DNA, the probe-binding vector y can be closely approximated as a linear combination of the predicted hybridization affinities of the species in the reference genome database (the columns of the matrix) weighted by their concentrations x; i.e., $y = x + n$, where the vector n accounts for

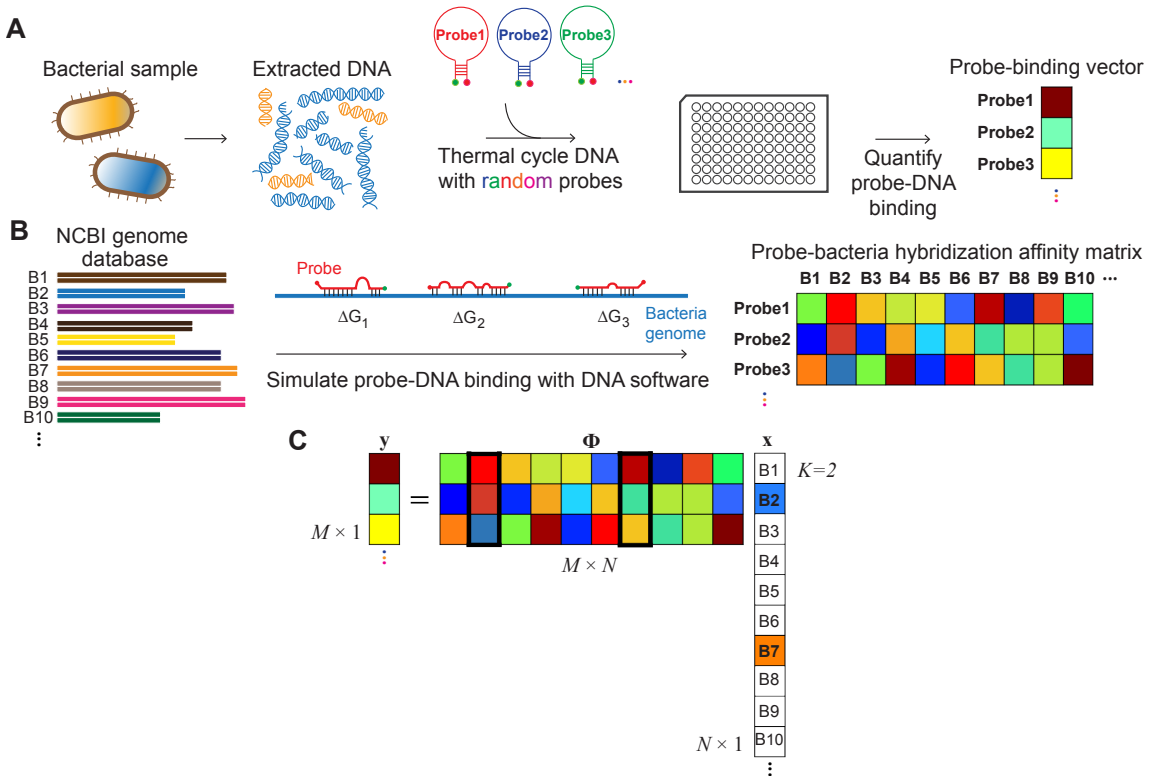

Figure 3.1 : Schematic of Universal Microbial Diagnostics (UMD) platform. (A) Genomic DNA is extracted from a bacterial sample and thermal cycled with M random DNA probes. The genome probe binding is quantified, producing a probe-binding vector y; in this study the random probes are in the form of molecular beacons (MBs), and the DNA-probe binding is quantified by the ratio of open/hybridized to closed/non-hybridized MBs. (B) The hybridization binding level of each probe to a potentially large reference database of N bacterial genomes (B1, B2,. . . , BN) is predicted using a thermodynamic model and stored in an $M \times N$ hybridization affinity matrix . (C) Assuming K bacterial species comprises the sample, the probe-binding vector y is a sparse linear combination of the corresponding K columns of the matrix weighted by the bacterial concentrations x, i.e., $y = x + n$, where the vector n accounts for noise and modeling errors. When K is small enough and M is large enough, can be effectively inverted using techniques from compressive sensing, yielding the estimate for the microbial makeup of the sample x; in this illustration the $K = 2$ bacteria labeled B2 and B7 are present in the sample.

noise and modeling errors (Fig. 3.2C) (see Materials and Methods).

The two key capabilities of the UMD platform are to (i) detect the presence and (ii) estimate the concentrations x of a potentially large number N of reference microbial genomes

in an infectious sample given only a small number M of probe-binding measurements y. Simply inverting the matrix is impossible in this case, since it has many more columns than rows. Fortunately, it is reasonable to assume that only a small number K of microbial genomes will be present in a given sample, in which case the concentration vector x is sparse, with K nonzero and $N - K$ zero (or close to zero) entries; when $K < M$, one can hope to invert to estimate the K nonzero concentrations. More rigorously (see the Supplementary Materials for details), when the columns of are sufficiently incoherent (close to orthogonal) and when $M = cK\log(N/K)$, where c is a small constant, we can apply the theory of compressive sensing [8, 9] to recover the concentrations x from the measurements y via a sparse optimization of the form

$$\min_x \|x\|_0, \text{subject to} \|y - \Phi x\|_2 < \Phi \tag{3.1}$$

Here, $\|x\|_0$ counts the number of non-zero values in the vector x, and σ bounds the energy of the noise vector n. Since $M = cK\log(N/K)$ scales logarithmically with N, the UMD platform has the potential to identify and estimate the concentrations of a large number N of potential microbial genomes using only a small number M of measurement probes. To ensure that the columns of are incoherent, we use DNA probes whose sequences are generated via a random permutation of nucleotides (Fig. 3.2A). We demonstrate that a small fixed set of randomly selected probes induce sufficiently incoherent hybridization patterns across the columns of and enable us to screen for a group of pathogenic organisms in vitro (Fig. 3.3.1). In addition, we showcase the average performance of several sets of randomly selected probes in universal pathogenic detection in silico (Fig. 3.3.2).

This universal sensing strategy can take on any physical embodiment (e.g., quantitative real-time PCR, DNA microarray, or WGS) for the detection of any DNA sequence (bacterial, viral, or fungal). To test and validate the concept, we recovered pathogenic bacteria using random probes in the form of mismatch-tolerant Sloppy Molecular Beacons

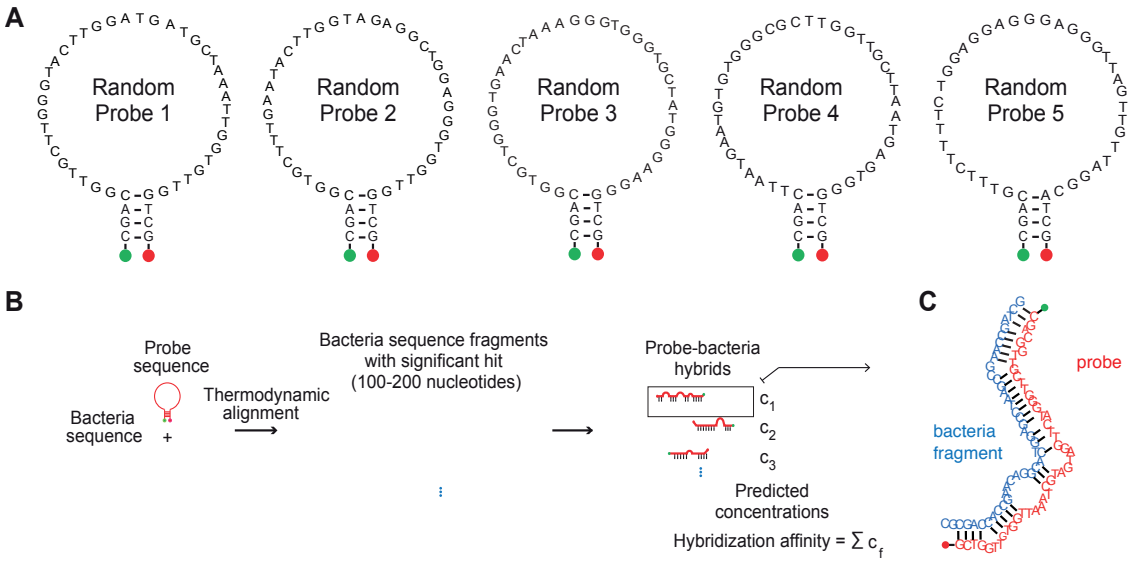

Figure 3.2 : Random probe design and hybridization affinity computation process. (A) DNA sequence structure of five test random DNA probes. (B) Both strands of the bacterial genome (blue lines) are first thermodynamically aligned with the probe sequence. The sequence of the bacteria is segmented into fragments of roughly equal length ($\sim 100 - 200$ nts), each containing a significant hybridization affinity with the probe. Then, all of the bacterial fragments and probe sequences along with the experimental conditions are fed into the DNA software [6] to predict all stable probe-bacteria complexes and concentrations. These concentrations, in aggregate, determine the concentration of opened molecular beacons, which is defined as the hybridization affinity of the probes to the bacterial genome. (C) Example of a predicted probe-bacteria fragment binding with many base pair mismatches.

(Sloppy MBs) [16]. In a conventional MB (20) for bacterial detection, the loop sequence is designed to target specific regions (e.g., 16S rDNA) within a single bacterium [23] or multiple bacteria [16].

In the MB probes for UMD, the loop sequence is selected as a random sequence (Fig. 3.2A, fig. S1, and Materials and Methods) of length 38 nts, and the 4 nt-long stem sequence is consistent across all probes, although other choices might be utilized (different design tradeoffs are discussed elsewhere [31]). The unusually long loop and short stem enable our random probes to form hybrids with several base pair mismatches across the entire bacterial genome and compensate for the lower signal intensity in the absence of DNA amplification methods such as PCR (Fig. 3.2C).

3.3 Results

3.3.1 Experimental proof of concept

To prove the UMD concept, we mixed five UMD MBs (as shown in Fig. 3.2A and characterized in fig. S2 with GC-contents 50, 56.5, 60.8, 50, and 52.7%, identical melting temperature of 40°C, and concentration 1M) in separate tubes (to prevent cross hybridization of probes) with genomic DNA from each of nine human infectious bacterial strains grouped into three categories: I. Exact sequence known (*Escherichia coli*, *Francisella tularensis*, *Staphylococcus aureus*, *Campylobacter jejuni*, and *Proteus mirabilis*), II. Exact sequence unknown (*Cupriavidus metallidurans* and *Micrococcus luteus*), and III. Clinical isolates, whose exact sequence is unknown (*Bacteroides fragilis* and *Enterobacter aerogenes*) (The identification of *Pseudomonas aeruginosa* and *Bifidobacterium dentium* strains, was tested using four random probes. See fig. S3-S5 for detection results). For bacteria in groups II and III, the DNA sequences in the database might not exactly match the sequences present

in the bacterial samples.

For each MB-bacterial species pair, equal volumes of probe and bacterial DNA were combined and subjected to a thermal cycling process of denaturing ($95°$C) and binding/cooling to $4°$C overnight. To quantify probe-DNA binding, the MB probes' Cy3 and Cy5 fluorescence intensities were measured with a fluorometer, and the fluorescence resonance energy transfer (FRET) ratios (a decrease represents MB opening due to DNA binding) were calculated by computing Cy5 intensity over total fluorescence intensity (Cy3 + Cy5). The FRET ratios from binding of the nine bacteria to the MBs are depicted in Fig. 3.3.1A.

In order to estimate the bacterial concentrations in physical units, we translated the FRET ratio of each bacterium-MB pair into the concentration of opened MBs or hybridization affinity, represented in units of molarity. For this, we experimentally obtained and fitted FRET ratios for each of the five MBs as a function of the concentration of their exact probe complements, using an optimization method described in [32] (fig. S2) (See table S1 and Materials and Methods for the fit curve parameters and fit method, respectively.). The R^2 values for the fits ranged from 0.97 to 0.99, suggesting a satisfactory fit. Based on the fit equations, the hybridization affinities corresponding to the FRET ratios for all bacteria were calculated. We refer to these measurements as the measured hybridization affinity vectors and show them in Fig. 3.3.1B. Our challenge is to decode the experimentally measured affinities of the bacterial species samples reacting to UMD MBs using compressive sensing recovery techniques. With the predicted hybridization affinities of $N = 9$ bacteria to $M = 5$ random probes stored in the computationally obtained 5×9, we used a variant of the Orthogonal Matching Pursuit (OMP) algorithm [33] and successfully identified the species present in each of the samples (Fig. 3). UMD estimated the relative bacterial concentrations with an average error of 11.5% (fig. S6).

To provide the physician or scientist with a metric quantifying how close the measured

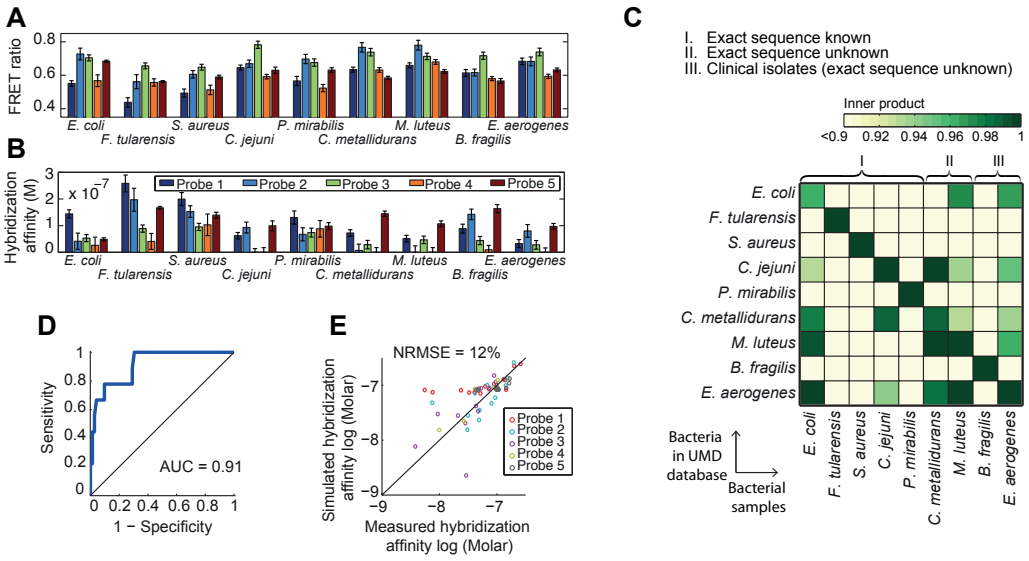

Figure 3.3 : Binding patterns of five random probes correctly identifies the bacteria present in nine diverse bacterial samples. (A) Experimentally measured FRET ratios to quantify hybridization between bacterial DNA and probes 1, 2, 3, 4, and 5. (B) Hybridization affinity between DNA samples and probes, converted from FRET ratio through the probe characteristic curve fit equations (table S1 and fig. S2). (C) Heat map of normalized inner products between the experimentally obtained hybridization affinity and predicted hybridization affinities (by thermodynamic model) for nine DNA samples as a measure of the similarity of the probe measurements to the bacteria in the dataset. DNA samples are clustered into three groups: I. Exact sequence known, II. Exact sequence unknown, and III. Clinical isolates (whose exact sequence is unknown). UMD correctly recovers the diagonally highlighted bacterium (with inner product > 0.9). (D) The average receiver operating characteristic (ROC) curve of UMD in detecting nine bacteria, assuming the independence of the different experiments. Each point on the curve corresponds to a threshold value between $[-1, 1]$. UMD achieves high values of the area under the curve (AUC > 0.9). (E) Correlation of measured and simulated hybridization affinities and the normalized root-mean-square error (NRMSE) of the prediction (straight line corresponds to maximum correlation). All experiments were performed in triplicate, and the results shown here average over the trials with the error bars representing the standard error of the mean (s.e.m.).

hybridization affinity vector is to that of each bacteria in the database and thus how confident the OMP detection results are, UMD can output the inner products between the normalized measured hybridization affinity vectors from the nine experiments and the columns of the centered and normalized matrix 5×9 (Fig. 3.3.1C). This metric measures the similarity of a pathogenic sample to bacteria in the UMD database. Using this metric, we characterized the performance of UMD in identifying the nine pathogens in terms of false positives and false negatives. We constructed the receiver operating characteristic (ROC) curve (Fig. 3.3.1D), where each point on the curve corresponds to a certain universal detection threshold in the range $[-1, 1]$ for all nine independent bacterial experiments. Inner product values above/below the detection threshold were considered as a positive/negative outcome, respectively. The area under the corresponding ROC curve (AUC = 0.91) suggests successful screening performance. Fig. 3.3.1E additionally shows the consistency of the measured and simulated hybridization affinities of nine bacteria to five random probes (different probes are shown in different colors). The normalized root-mean-square error NRMSE = 12%, suggests that our thermodynamic modeling of bacteria-probe hybridization is accurate.

Next, we assessed the performance of UMD in detecting our nine test species from a list of common pathogens using our five DNA probes. We expanded the reference genome database to contain 40 genera (i.e., $M = 5 \gg N = 40$), including bacterial pathogens listed by the Centers for Disease Control and Prevention (CDC) as the most common notifiable human diseases [30]. With the most common pathogens' genomes in the database, the detection performance remained above AUC = 0.84, suggesting a high recovery rate with only five random probes.

3.3.2 Extension by simulation

Thus far, we have presented an experimental proof-of-concept that validates UMD's ability to detect eleven test species among a list of pathogens using a fixed set of five randomly selected test probes. We next numerically demonstrate that, if a sufficient number of probes is used, then any group of randomly selected probes will detect the presence of one ($K = 1$) or a mixture of several ($K = 2, 3, \dots$) pathogenic organisms in a sample out of a database of 40 pathogenic organisms. We introduced additive white Gaussian noise to the simulated hybridization affinity vectors to capture the variance in the hybridization affinities among the independent test trials in Fig. 3.3.1. The noise levels were extrapolated from the above eleven test bacteria experiments, with the noise variance set to $\sigma_0 = 2.4 \times 10^{-8}$ M. In order to control for differences in the genome size of each organism, we normalized numerical simulations to unit weight of bacterial DNA.

In Fig. 3.3.2A, we first demonstrate the detection performance of UMD in identifying a single bacterium ($K = 1$) among the pathogen database at different noise levels. As the ROC curves suggest, UMD's detection performance improves when the noise variance decreases. With only a five-fold decrease in the noise variance, UMD identifies all 40 bacteria in the database almost perfectly (AUC > 0.95) using only five randomly selected probes.

The ability of UMD to universally detect target species can be improved by increasing the number of random probes. Fig. 3.3.2B demonstrates that UMD identifies all 40 bacteria in the CDC database almost perfectly (AUC = 0.95) with any $M = 15$ randomly selected MBs when the noise variance is similar to that measured experimentally (Fig. 3). UMD has the unique advantage that it can recover more than a single ($K > 1$) organism in an infectious sample. To evaluate the minimum number of probes M required for this task, we used the Basis Pursuit De-Noising (BPDN) algorithm (as described in the Supplementary Ma-

terials) to identify the composition of a sample containing $K = \{2, 3\}$ equi-concentration bacterial species (Fig. 4C). We found that any set of $M = 15$ randomly selected probes will recover all possible mixtures of $K = \{2, 3\}$ pathogenic species in the CDC database. The error bars show the standard deviation over 1000 test trials with different sets of random MBs. This result confirms that the incoherence requirement for compressive sensing is empirically satisfied for the pathogenic strains in the CDC database and thus that UMD is capable of screening for pathogenic bacteria at the genus-level.

We next evaluated UMD's performance for species-level bacterial detection. We focused on differentiating among 24 species of Staphylococcus genus and 23 species of Vibrio genus in silico. We identified the composition of samples containing Staphylococcus species using 11 random probes (fig. S7A) and the composition of samples containing Vibrio species using 18 random probes (fig. S7B) with high sensitivity and specificity (AUC $>$ 0.95). This underscores UMD's potential to differentiate pathogens at high taxonomic resolution. Using the UMD platform, one can tradeoff between universality (detecting species outside of the library) and cost efficiency (number of probes). That is, it is possible to select a set of probes that achieves better detection performance in terms of specificity and sensitivity than the average performance of random probe sets at the cost of universality. For example, in Fig. 4C, some of the probe sets achieved 100% accuracy using three fewer probes than the number required for universal recovery. To capitalize on this phenomenon, we developed a greedy probe selection (GPS) algorithm that rapidly selects these optimized probes given a very large database of genome-probe hybridization affinities (see Materials and Methods for more details on GPS).

Fig. 4, D and E, illustrate the UMD confusion matrices in detecting pathogenic bacteria using $M = 3$ and $M = 10$ probes selected using GPS. The false positive rate drops for all of the bacteria in the database as the number of probes increases from $M = 3$ (AUC $>$

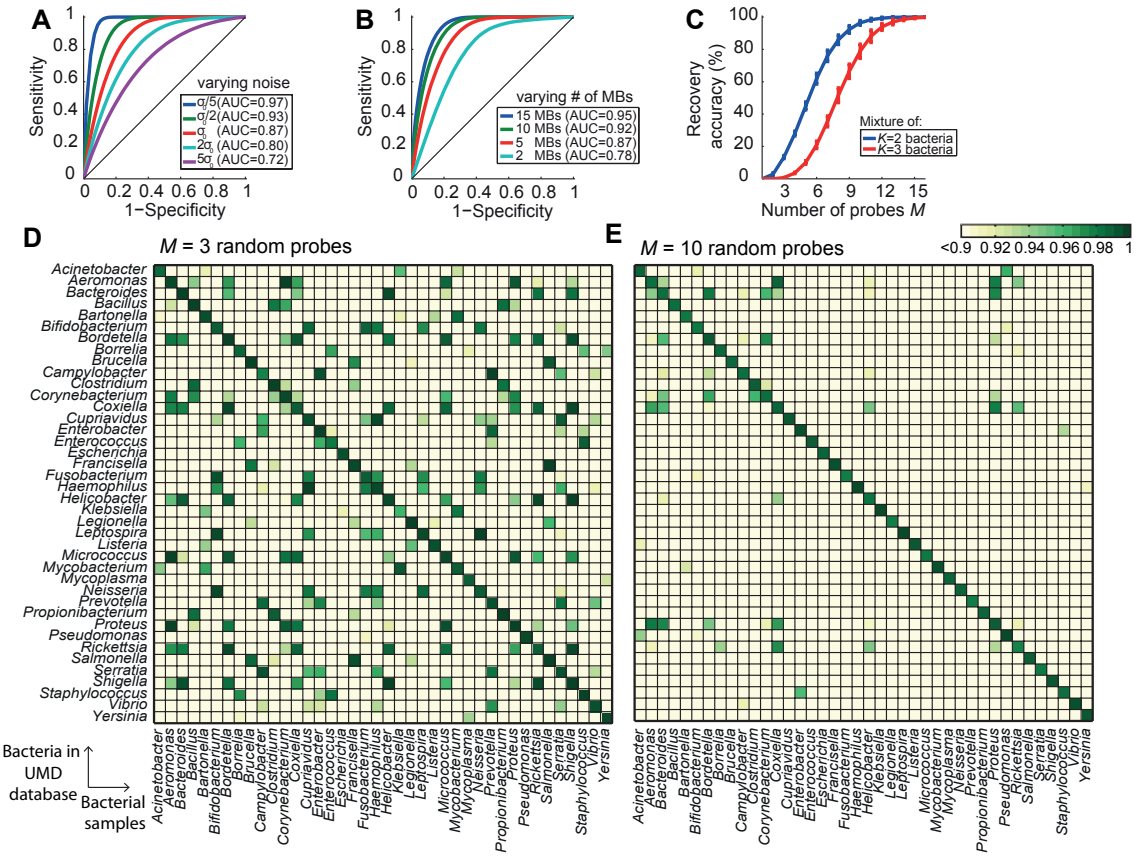

Figure 3.4 : Performance of UMD platform in genus-level recovery of 40 species listed as the most common human infectious genera by Centers for Disease Control and Prevention (CDC) with different number of random probes M and noise variance σ. (A) The ROC curve in detecting single bacterium ($K = 1$) with different noise levels. $\sigma_0 = 2.4 \times 10^{-8}$ M denotes the variance of the additive white Gaussian noise (AWGN) used in the simulation. This value is obtained from the experiments in Fig. 3.3.1 by calculating the propagated variance of measured FRET ratios. UMD performs more accurately with lower noise variance. The detection is almost perfect (AUC > 0.95) under noise variance $\sigma = \sigma_0/5$. (B) The average ROC curve in detecting single bacterium using different number of random probes M and fixed noise variance $\sigma = \sigma_0$. The detection performance universally improves over all the 40 species by increasing the number of random probes. With 15 random probes, UMD achieves almost perfect detection performance (AUC > 0.95). (C) The percentage of simulated trials, where K bacteria present in the samples were recovered correctly with zero false positives, among all possible bacteria mixtures (blue and red curves corresponding to $K = 2$ and $K = 3$ bacteria, respectively). Simulations were repeated 1000 times with randomly selected MBs, and error bars represent standard deviation (SD). (D, E) Confusion matrices illustrating the detection result of UMD using $M = 3$ and $M = 10$ probes selected by the greedy probe selection (GPS) algorithm.

0.95) to $M = 10$ (AUC > 0.99). While the performance detection is high (AUC > 0.99), the confusion matrix shows few cases where the inner product values for possible species (e.g., *Coxiella*, *Aeromonas*, and *Proteus* when the actual sample contains emphCoxiella) are only slightly separated. Fig S8 shows that greater separation between inner product values for candidate bacterial species can be achieved by using a larger number of UMD probes. This can increase the robustness of the UMD system (ensure low false positive rate) for noisier environments.

While mainly intended to rapidly screen for pathogens at higher taxonomy levels, UMD can also provide strain-level information to the physician using additional GPS-selected probes. In fig. S9 we demonstrate that GPS selects UMD probes that differentiate among 9 strains of *E. coli* (8 pathogenic and one nonpathogenic) with high detection accuracy (AUC > 0.95) in silico.

The theory behind the UMD can be extended to identify more complex samples using a relatively small number of random probes. We verified that the UMD platform can recover complex microbial samples containing up to one hundred active species out of a large dictionary of $N = 1500$ bacterial genera. We first computed the hybridization affinity of a set of random probes to $N = 1500$ representative species of all sequenced bacterial genera in NCBI website. Then we used the resulting hybridization affinity matrix to identify the composition of samples containing K unique species with equal concentrations (the most difficult case where the sample contains multiple species all at significant concentration). For each value of K, fig. S10 shows the minimum number of random probes M required to identify the composition of 1000 simulated complex samples containing K species randomly selected from $N = 1500$ genera. Fig. S10 illustrates that the number of required probes M closely ($R^2 = 0.98$) follows the compressive sensing theory $M = cK\log(N/K)$ with constant $c = 2.94$. That is, UMD requires a number of probes that grows logarith-

mically with the number of target bacteria N and sub-linearly with the number of active bacteria K in the sample.

3.4 Conclusion

UMD probes are universal in the sense that a fixed set of probes captures the salient information required to distinguish between members of a large and growing database of species [34]. This gives UMD a potentially important future proof property: a fixed set of measurement probes can be used to detect and estimate the concentration of newly sequenced species not yet present in the library. To detect a new organism, the software merely has to be adjusted to take into account how the new organism will react to the existing probe set; however, new capture probes are not required. Moreover, since the number of probes grows only logarithmically in the size of the library, the UMD platform naturally contends with the data deluge [35] from new microbial species being discovered and sequenced every day.

Several other pathogen detection schemes are currently under investigation [16, 36, 37, 38]. To the best of our knowledge, UMD is the only technique that enables a unified representation of bacterial organisms in a low-dimensional geometric space. The theory of compressive sensing provides rigorous recovery guarantees and suggests algorithms to leverage this geometry to both detect bacteria and estimate their concentrations efficiently. Our successful implementation of UMD confirms that a small number of random DNA probes satisfy the incoherency requirements of compressive sensing theory and can be used for universal microbial sensing.

The UMD platform has the potential to rapidly direct physicians to use appropriate antibiotics or treatment and thus minimize the risk of antibiotic resistance. It can also be utilized in biodefense applications to classify multiple novel and mutant agents. With further

optimization of the probe design and detection schemes, we expect that UMD will be able to sense an even wider range of organisms (e.g., viruses, fungi) and various biomolecules of interest (genes, proteins).

Finally, the theory behind the UMD platform can be applied to DNA sensing in several other incarnations, including reads from a sequencer, e.g., we may be able to quickly identify a bacterium from a subset of reads, rather than requiring full alignment or assembly. Application of such signal acquisition principles to biological sensing systems will shape the future of microbial diagnostics.

3.5 Materials and Methods

3.5.1 Random DNA probe construction and preparation

To implement the Universal Microbial Diagnostics (UMD) platform, we obtained DNA oligonucleotides for the random DNA probes and their exact complements from Integrated DNA Technologies (Coralville, IA). The sequences are provided below. $MgCl_2$, KCl, and sterile nuclease-free water for making the molecular beacon (MB) buffer were purchased from Fisher Scientific (Waltham, MA). 1M Tris-HCl solution (pH 8.3) and Tris-EDTA buffer (TE buffer; 10 mM Tris-HCl, 0.1 mM EDTA pH 8) were obtained from Teknova (Hollister, CA). To prevent nuclease contamination, all work surfaces and materials were routinely cleaned with RNAse OFF decontamination solution (Takara, Japan). Random probe 1: 5'- /5Cy5/CGA CGG TTG CTT GGG TAC TTG GAT GAT GCT AAA TTG GTG TTG GTC G/3Cy3Sp/ -3', Random probe 2: 5'- /5Cy5/CGA CGG TGC TTT GAA TAC TTG GTA GAG GCT GGA GGG TGG TTG GTC G/ 3Cy3Sp/ -3', Random probe 3: 5'- /5Cy5/CGA CGG TGC TGG GTG AAC TAA AGG GTG GGT GCT ATG GGA AGG GTC G/3Cy3Sp/ -3', Random probe 4: 5'/5Cy5/CGA CTT AAT GAA TGT GTG

GGC GCT TGG TTG CTT AAT GAG TGG GTC G/3Cy3Sp/ -3', and Random probe 5: 5'- /5Cy5/CGA CGT TTC TTT TCT GGA GGA GGG AGG GTT AGT TGT TAG GCA GTC G/3Cy3Sp/ -3'. Random probe complement 1: 5'- CGA CCA ACA CCA ATT TAG CAT CAT CCA AGT ACC CAA GCA ACC GTC G -3', Random probe complement 2: 5'- CGA CCA ACC ACC CTC CAG CCT CTA CCA AGT ATT CAA AGC ACC GTC G -3', Random probe complement 3: 5'- CGA CCC TTC CCA TAG CAC CCA CCC TTT AGT TCA CCC AGC ACC GTC G-3', Random probe complement 4: 5'- CGA CCC ACT CAT TAA GCA ACC AAG CGC CCA CAC ATT CAT TAA GTC G-3', and Random probe complement 5: 5'- CGA CTG CCT AAC AAC TAA CCC TCC CTC CTC CAG AAA AGA AAC GTC G-3'.

3.5.2 Random DNA probe design

In the design of molecular beacons [14] for random DNA probes, the length and GC content (ratio of G+C to other nucleotides) of the probe loop and stem sequences were considered to strike a balance between two factors: fluorescence signal level and probe stability. Signal intensity was especially important in our detection scheme, since no DNA amplification method (such as PCR) was utilized. Similar to sloppy molecular beacons (sloppy MBs) [16], we selected the random probe loop sequence to be longer that traditional MBs. In addition, we made the stem sequence one nucleotide shorter to introduce additional sloppiness (i.e., hybridization in presence of more base-pair mismatches). Our challenge was to find probes that maintain the MB's signature hairpin structure over a wide range of temperatures (4-50°C) after introducing additional sloppiness. To produce random MBs we followed the following procedure: We first generated one million random sequences of length 46 nucleotides with fixed stem sequences on both ends (Fig. 3.2A). Then, we used a package in the DNA software (Visual OMP DE) to generate all the possible stable

and secondary structures of the sequences in the experimental thermodynamic conditions. We parsed the output of the DNA software and filtered out the probes with undesired secondary structures or melting temperatures. Fig. S1 shows the gain in hybridization affinity obtained using the random MBs in comparison with traditional MBs and sloppy MBs in binding to the *E. coli* genome. By no means is this the only method to generate random probes for a UMD platform; any method that produces probes with a stable hairpin structure and uniform melting temperature while providing the required signal intensity can be utilized.

3.5.3 Generation of random probe characteristic curves

The experimentally measured fluorescence resonance energy transfer (FRET), defined as the ratio of Cy5 intensity over total fluorescence intensity (Cy3 + Cy5), is a function of the concentration of open random probes in the solution, i.e., the probe-target hybridization affinity. To discern the hybridization affinity between a probe and target in units of molarity rather than as a FRET ratio, a characteristic curve was constructed for each probe. These curves presented the FRET ratio as a function of the concentration of open probes in molarity. To obtain the characteristic curves, random probes were diluted to 1μM in 1 X MB buffer (4 mM $MgCl_2$, 50 mM KCl, 10 mM Tris-HCl, pH = 8, in sterile RNAse free water). DNA oligonucleotides perfectly complementary to the random probes were diluted using 1 X TE buffer to 10^{-5}, 10^{-6}, 8×10^{-7}, 6×10^{-7}, 4×10^{-7}, 2×10^{-7}, 10^{-7}, 8×10^{-8}, 6×10^{-8}, 4×10^{-8}, 2×10^{-8}, 10^{-8}, 10^{-9}, 10^{-10}, or 10^{-11} M concentration. 25 μL of 1 μM random DNA probes (diluted in MB buffer) were added to 25 L of perfect complement DNA of various concentrations, or to the TE buffer-only control. The DNA mixture was briefly centrifuged with a mini centrifuge (VWR) to collect all DNA to the bottom of the tube. Then the DNA was hybridized using a MyCycler Thermal Cycler (Bio-Rad) under

the following conditions: 95°C for 5 minutes, 50°C for 2 minutes, 30°C for 1 minute, 20 C for 1 minute, and 4 C for 2 minutes. 45 L of each thermal cycled mixture was added to 155 μL 1 X MB buffer in a black flat bottom 96 well plate (Corning) and kept at 4°C overnight. A non-linear optimization algorithm (22) was utilized to fit the parameters a, b, n and FRET_0 to the characteristic curve: $\text{FRET}(c) = \text{FRET}_0 + a/(1 + b(10^{-6} - c)^{-n})$ (fig. S2). The coefficient of determination (R^2) and root mean square error (RMSE) for the curve fits are reported in table S1.

3.5.4 Bacterial DNA extraction

Overnight cultures of *S. aureus USA 300* and *E. coli MG1655* were used to inoculate fresh cultures grown in 50-100 mL 2xTY or Luria-Bertani (LB) broth, respectively. *F. tularensis* LVS was obtained from Dynport Vaccine Company LLC (derived from NDBR101 Lot 4) and grown in modified Mueller-Hinton cation-adjusted (MHII) broth (Becton Dickinson) supplemented with sterile 0.1% glucose, sterile 0.025% ferric pyrophosphate, and 2% re-constituted IsoVitaleX (Becton Dickinston). Cultures were pelleted and washed three times with sterile PBS. To release chromosomal DNA cells were resuspended in TE buffer and mixed with 10% sodium dodecyl sulfate (SDS) and Proteinase K at 65°C overnight. DNA was isolated using phenol: chloroform and precipitated via ethanol precipitation (protocol adapted from elsewhere [39]). DNA pellets were resuspended in 50 L TE buffer and stored at -20°C.

Bacterial strains, *C. jejuni*, *P. mirabilis*, *C. metallidurans*, *M. luteus*, *B. dentium*, *E. aerogenes*, *B. fragilis*, and *P. aeruginosa* were grown overnight in 30 mL Brain Heart Infusion (BHI) media (BD) at 37°C. Bacterial cells were pelleted, washed two times with sterile 1X PBS, and resuspended in TE buffer. Proteinase K (1 mg/mL) (Sigma) and 0.5% SDS were added to the bacterial cells, which were then incubated overnight at 55°C on an

orbital shaker. The samples were then mixed with phenol-chloroform (Invitrogen) and centrifuged; supernatants were transferred to a fresh tube. This aqueous phase was then mixed with an equal volume of chloroform and centrifuged (and repeated). Finally, 1/10 volume of 2 M sodium chloride and an equal volume of isopropanol was added to precipitate the DNA. This mixture was incubated at -20°C for 30 min and centrifuged. The pellets were rinsed with 70% ethanol, air dried, and resuspended in TE buffer.

3.5.5 Random probe and bacterial DNA hybridization

Bacterial DNA was diluted to approximately 500 ng/μL using TE buffer and kept at -20°C until use. The random MB probes were diluted to 1 μM in MB buffer prior to use. 25 μL of 1 μM random probes (diluted in MB buffer) was added to 25 L of TE buffer control and *E. coli*, *F. tularensis*, *S. aureus*, *C. jejuni*, *P. mirabilis*, *C. metallidurans*, *M. luteus*, *B. dentium*, *E. aerogenes*, *B. fragilis*, or *P. aeruginosa* DNA. The DNA mixture was briefly centrifuged to collect DNA and then hybridized using a MyCycler Thermal Cycler (Bio-Rad) under the following conditions: 95°C for 5 minutes, 50°C for 2 minutes, 30°C for 1 minute, 20°C for 1 minute, and 4°C for 2 minutes. 45 L of each thermal cycled mixture was added to 155 μL 1 X MB buffer in a black flat bottom 96 well plate (Corning) and kept at 4°C overnight.

3.5.6 Measuring FRET ratio through fluorescence as indicator of random probe-bacteria hybridization

The FRET ratio for the genomic DNA samples following hybridization with each of the random probes was determined by reading the Cy3 and Cy5 fluorescence using a Fluorolog-3 spectrofluorometer (Jobin Yvon Horiba, Edison, NJ) coupled with a MicroMax 384 MicroWell Plate Reader and water-cooled PMT detector. Samples were excited at 545 nm, and single point fluorescence measurements were taken with at 562 nm and 677 nm emission

(optimal wavelengths determined through excitation-emission matrix analysis) to measure the Cy3 and Cy5 fluorescence, respectively. The FRET ratio was calculated as Cy5 / (Cy5 + Cy3).

3.5.7 Determining DNA hybridization affinity via SantaLucia thermodynamic models

A comprehensive thermodynamic model by SantaLucia et al. [6] was utilized to predict the hybridization of probes to bacterial genomes. The SantaLucia model incorporates thermodynamic parameters for mis-hybridizations between two DNA sequences. We utilized two software packages: ThermoBlast DE, which performs fast alignment of sequences against large genome databases to discover thermodynamically stable hybridizations, and Visual OMP DE, which simulates hybridization experiments with detailed solution conditions and generates results for melting temperature (Tm), Gibbs free energy (ΔG), and the percentage-based concentration of each resultant species post-experiment. The secondary structure of each monomer, homodimer, and heterodimer species formed from the constituent probes and target fragments can also be visualized.

To calculate the hybridization affinity of a genome to a probe, we first used the ThermoBlast package and thermodynamically aligned the sequence of the random probe to both complement strands of the target genome. We extracted all the sequence fragments of the genome (100-200 nucleotides) that aligned with the probe sequence with a predicted melting temperature within approximately 35°C of the melting temperature of the sequence genome. We then used Visual OMP DE to simulate the hybridization between the probe and the target genome, using the target fragments (Fig. 2). Every simulation contained information on the probe sequence, the target fragment sequences, and conditions for the experiment, including probe concentration (1 M), unit target concentration (500 ng/L for all

bacteria), assay temperature (4°C), hybridization buffer composition (4 mM Mg++, 50mM Na+, 0 M Glycerol, 0 M DMSO, 0 M Formamide, 0 M TMAC, 0 M Betaine), and pH (= 8). This procedure was repeated for each probe-target genome pair. We used the percentage of probe-target heterodimer structures formed, i.e., the percentage of probes that are bound to target fragments, as an estimate for the hybridization affinity of the probe to each target (Fig. 2B).

3.5.8 Linearity assumption considerations in UMD

In the UMD platform, the probe concentration (1×10^{-6} M) is in far excess of the target concentrations ($\sim 1 \times 10^{-10}$ M); therefore, we are able to linearly combine the hybridization affinity signatures that we measure for individual targets using the hybridization model. Due to the flooding of excessive number of probes, each target fragment has its choice of binding/not binding to the probes, and thus we can safely sum together multiple target interactions of the same probe, assuming them to be independent.

3.5.9 Receiver-operator curve (ROC) analysis

Receiver-operator curve (ROC) analysis was performed by plotting a ROC curve showing the sensitivity and (1-specificity) for 1000 threshold values ranging from -1 to 1. For each threshold value, the following procedure was performed on the data matrix of normalized inner products between the experimentally obtained hybridization affinity and predicted hybridization affinities (by thermodynamic model) for the nine independent bacterial DNA samples (Fig. 3C): Each entry in the inner product data matrix was compared with the threshold value to determine the number of true positives, false positives, true negatives, and false negatives. True positives were identified when values in the diagonal entries of the inner product data matrix were greater than the threshold value since diagonal entries

represent the correct classification of the bacterial sample with its corresponding genus in the database. False positives were identified as off-diagonal values that were greater than the threshold value. True negatives were identified as off diagonal values that were less than or equal to the threshold value. False negatives were identified as diagonal values that were less than or equal to the threshold value. For each threshold value, sensitivity was defined as (# true positives / (# true positives + # false negatives)) and specificity was defined as (# true negatives / (# true negatives + # false positives)).

3.5.10 Greedy probe selection

Given a set of P random probes, finding the set of M probes with the best detection performance in terms of sensitivity and specificity is an extremely challenging problem. A brute force search would require one to search among all possible combination of M probes to find the optimal probe set. This combinatorial search algorithm grows quadratic with P and thus becomes computationally intractable when the number of probes grows. We thus developed a rapid probe selection method that we dub Greedy Probe Selection (GPS). With a small sacrifice in sensitivity, GPS finds the best performance probe in a few seconds: exponentially faster than the nave search method. The algorithm in each iteration finds the probe that maximizes a detection performance criterion (here the maximum pairwise correlation of bacteria) and adds it to the list of probes picked from the previous iterations. GPS stops when the maximum desired number of probes is reached.

Chapter 4

Insense: Incoherent Sensor Selection for Sparse Signals*

4.1 Introduction

The accelerating demand for capturing signals at high resolution is driving acquisition systems to employ an increasingly large number of sensing units. However, factors like manufacturing costs, physical limitations, and energy constraints typically define a budget on the total number of sensors that can be implemented in a given system. This budget constraint motivates the design of *sensor selection* algorithms [40] that intelligently select a subset of sensors from a pool of available sensors in order to lower the sensing cost with only a small deterioration in acquisition performance.

In this paper, we extend the classical sensor selection setup, where D available sensors obtain linear measurements of a signal $x \in \mathbb{R}^N$ according to $y = \Phi x$ with each row of $\Phi \in \mathbb{R}^{D \times N}$ corresponding to one sensor. In this setup, the sensor selection problem is one of finding a subset Ω of sensors (i.e., rows of Φ) of size $|\Omega| = M$ such that the signal x can be recovered from its M linear measurements

$$y_\Omega = \Phi_\Omega x \tag{4.1}$$

with minimal reconstruction error. Here, $\Phi_\Omega \in \mathbb{R}^{M \times N}$ is called the *sensing matrix*; it contains the rows of Φ indexed by Ω.

The lion's share of current sensor selection algorithms [40, 41, 42] select sensors that

*This chapter is a revised version of an article, see ref. [2]

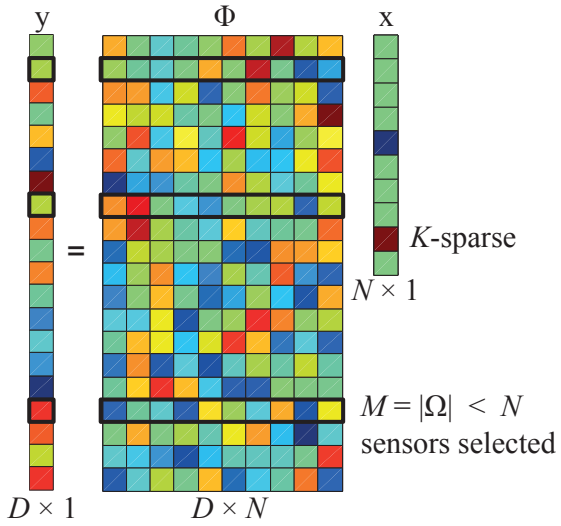

Figure 4.1 : Schematic of the sensor selection problem for sparse signals. Here, $M = 3$ sensors indexed by $\Omega = \{2, 8, 17\}$ are selected from $D = 20$ available sensors to recover a $K = 2$-sparse vector $x \in \mathbb{R}^N$, $N = 10$, from the linear system $y_\Omega = \Phi_\Omega x$.

best recover an arbitrary signal x from $M > N$ measurements. In this case, (4.1) is *overdetermined*. Given a subset of sensors Ω, the signal x is recovered simply by inverting the sensing matrix while computing $\Phi_\Omega^\dagger y_\Omega$, where $\Phi_\Omega^\dagger$ is the pseudoinverse of Φ_Ω.

Such approaches do not exploit the fact that many real-world signals are (near) *sparse* in some basis [43]. It is now well-known that (near) sparse signals can be accurately recovered from a number of linear measurements $M \ll N$ using sparse recovery/compressive sensing (CS) techniques [8, 9, 44]. Conventional sensor selection algorithms are not designed to exploit low-dimensional signal structure. Indeed, they typically fail to select the appropriate sensors for sparse signals in this *underdetermined* setting ($M < N$).

In this paper, we develop a new sensor selection framework that finds the optimal subset of sensors Ω that best recovers a (near) sparse signal x from $M < N$ linear measurements (see Fig. 4.1). In contrast to the conventional sensor selection setting, here the sensing equation (4.1) is underdetermined, and it can not be simply inverted in closed form.

A key challenge in sensor selection in the underdetermined setting is that we must replace the cost function that has been so useful in the classical, overdetermined setting, namely the estimation error $\|x - \hat{x}\|_2^2$ (or the covariance of the estimation error in the presence of noise). In the overdetermined setting, this error can be obtained in closed form simply by inverting (4.1). In the underdetermined setting, this error has no closed form expression. Indeed, recovery of a sparse vector x from y_Ω requires a computational scheme [45, 46].

Fortunately, the sparse recovery theory tells us that one can reliably recover a sufficiently sparse vector x from its linear measurements y_Ω when the columns of the sensing matrix Φ_Ω are sufficiently *incoherent* [47, 48, 49]. Define the *coherence* between the columns ϕ_i and ϕ_j in the sensing matrix Φ_Ω as $\mu_{ij}(\Phi_\Omega) = \frac{|\langle \phi_i, \phi_j \rangle|}{\|\phi_i\| \|\phi_j\|}$. If the values of $\mu_{ij}(\Phi_\Omega)$ for all pairs of columns (i, j) are bounded by a certain threshold, then sparse recovery algorithms such as Basis Pursuit (BP) [7, 50, 47] can recover the sparse signal x exactly. This theory suggests a new cost function for sensor selection. To select the sensors Ω that reliably recover a sparse vector, we can minimize the *average squared coherence*

$$\mu_{\text{avg}}^2(\Phi_\Omega) = \frac{1}{\binom{N}{2}} \sum_{1 \le i < j \le N} \mu_{ij}^2(\Phi_\Omega). \tag{4.2}$$

The challenge now becomes formulating an optimization algorithm that selects the subset of the *rows* of Φ (the sensors) whose *columns* have the smallest average squared coherence.

4.1.1 Contributions

We make three distinct contributions in this work. First, we are the first to propose and study the sparse-signal sensor selection problem. Below we demonstrate that the standard cost functions used in overdetermined sensor selection algorithms are not suitable for the underdetermined case.

Second, we develop a new sensor selection algorithm that optimizes the new cost function (4.2); call it the *Incoherent Sensor Selection (Insense)* algorithm. Insense employs an efficient optimization technique to find a subset of sensors with smallest average coherence among the columns of the selected sensing matrix Φ_Ω. The optimization technique – projection onto the convex set defined by a scaled-boxed simplex (SBS) constraint – is of independent interest. We have made the codes for the Insense algorithm available online at https://github.com/amirmohan/Insense.git.

Third, we demonstrate the superior performance of Insense over conventional sensor selection algorithms using an exhaustive set of experimental evaluations that include real-world datasets from microbial diagnostics and structural health monitoring and six performance metrics: average mutual coherence, maximum mutual coherence, sparse recovery performance, frame potential, condition number, and running time. We demonstrate that, for the kinds of redundant, coherent, or structured Φ that are common in real-world applications, Insense finds the best subset of sensors in terms of sparse recovery performance by a wide margin. Indeed, in these cases, many conventional sensor selection algorithms fail completely.

4.1.2 Paper organization

We first overview the state-of-the-art overdetermined sensor selection algorithms and elaborate on the unique properties of the undetermined problem and its relation to CS in Subsection 4.1.3. Section 4.2 states our coherence minimization formulation to solve the underdetermined sensor selection problem, and Section 4.3 details the Insense algorithm. Section 4.4 presents our simulation results on synthetic and real-world datasets, and Section 4.5 concludes the paper.

4.1.3 Related work

This section overviews the conventional sensor selection algorithms and their cost functions for the overdetermined case ($M > N$). We then detail the relationship of the underdetermined case to the field of CS.

4.1.4 Conventional sensor selection algorithms

Existing sensor selection algorithms mainly study the sensor selection problem in the overdetermined regime (when $M \geq N$) [40, 41, 51, 42].

In the overdetermined regime, robust signal recovery can be obtained using the solution to the least squares (LS) problem in the sensing model (4.1), which motivates as a cost function the mean squared error (MSE) [52, 53, 54] or a proxy of the MSE [55, 56, 57] of the LS solution.

For instance, Joshi, et al. [40] employ a convex optimization-based algorithm to minimize the log-volume of the *confidence ellipsoid* around the LS solution of x. Shamaiah et al. [41] develop a greedy algorithm that outperforms the convex approach in terms of MSE. FrameSense [42] minimize the frame potential (FP) of the selected matrix

$$\text{FP}(\Phi_\Omega) = \sum_{\forall (i,j) \in \Omega, i<j} |\langle \phi^i, \phi^j \rangle|^2, \tag{4.3}$$

where ϕ^i represents the i^{th} row of Φ. Several additional sensor selection algorithms that assume a non-linear observation model [58, 59] also operate only in the overdetermined regime.

4.1.5 Connections to compressive sensing

Our model for sensor selection has strong connections to, and enables powerful extensions of, the CS problem, in which a (near) sparse signal is recovered from a small num-

ber of randomized linear measurements [8, 9, 44]. First, note that CS theory typically employs *random* sensing matrices; for instance it has been shown that many ensembles of random matrices, including partial Fourier, Bernoulli, and Gaussian matrices, result in sensing matrices with guaranteed sparse recovery [60, 61]. Recently, there have been efforts to *design* sensing matrices that outperform random matrices for certain recovery tasks [62, 63, 64, 65, 66]. For instance, Grassmannian matrices [63] attain the smallest possible mutual coherence and hence can lead to better performance in some applications.

However, many real-world applications do not involve random or Grassmannian sensing matrices; rather the sensing matrix is dictated by physical constraints that are specific and unique to each application. For example, in the sparse microbial diagnostic problem [1], the entries of the sensing matrix Φ are determined by the hybridization affinity of random DNA probes to microbial genomes and do not necessarily follow a random distribution. Similarly, in the structural health monitoring problem [67, 68, 69], the sensing matrix is constrained by the solution to the wave equation. A key outcome of this paper is a new approach to construct practical and realizable sensing matrices using underdetermined sensor selection (via Insense).

4.2 Problem statement

Consider a set of D sensors taking nonadaptive, linear measurements of a K-sparse (i.e., with K non-zero elements) vector $x \in \mathbb{R}^N$ following the linear system $y = \Phi x$, where $\Phi \in \mathbb{R}^{D \times N}$ is a given sensing matrix. We aim to select a subset Ω of sensors of size $|\Omega| = M \ll D$, such that the sparse vector x can be recovered from the resulting $M < N$ linear measurements $y_\Omega = \Phi_\Omega x$ with minimal reconstruction error. Here, Φ_Ω contains the rows of Φ indexed by Ω, and y_Ω contains the entries of y indexed by Ω. This model for the sensor selection problem can be adapted to more general scenarios. For example, if the

signal is sparse in a basis Ψ, then we simply consider $\Phi = \Theta\Psi$ as the new sensing matrix, where Θ is the original sensing matrix.

In order to find a subset Ω of sensors (rows of Φ) that best recovers a sparse signal x from y_Ω,[†] we aim to select a submatrix $\Phi_\Omega \in \mathbb{R}^{M\times N}$ that attains the lowest average squared coherence

$$\mu^2_{\text{avg}}(\Phi_\Omega) = \frac{1}{\binom{N}{2}} \sum_{1\le i<j\le N} \frac{|\langle \phi_i, \phi_j\rangle|^2}{\|\phi_i\|^2\|\phi_j\|^2}, \tag{4.4}$$

where ϕ_i denotes the i^{th} column of Φ_Ω. The term μ_{avg} averages the off-diagonal entries of $\Phi_\Omega^T\Phi_\Omega$ (indexed by $1 \le i < j \le N$) after column normalization. Other measures of coherence (e.g., max coherence $\mu_{\text{max}}(\Phi_\Omega) = \max_{i<j}\mu_{ij}$) can also be employed by slightly modifying the optimization procedure developed below. We choose to work with average coherence due to its simplicity and the fact that our experiments show that its performance is comparable to max coherence.

Define the diagonal selector matrix $Z = \text{diag}(z)$ with $z = [z_1, z_2, z_3, \ldots, z_D]^T$ and $z_i \in \{0, 1\}$, where $z_i = 1$ indicates that the i^{th} row (sensor) in Φ is selected and $z_i = 0$ otherwise. This enables us to formulate the sensor selection problem as the following optimization problem

$$\begin{aligned} \underset{z\in\{0,1\}^D}{\text{minimize}} \sum_{1\le i<j\le N} \frac{{G_{ij}}^2}{G_{ii}\, G_{jj}}, \\ \text{s.t.} \quad G = \Phi^T Z \Phi,\ \mathbf{1}^T z = M, \end{aligned} \tag{4.5}$$

where $\mathbf{1}$ is the all-ones vector. This Boolean optimization problem is combinatorial, since one needs to search over $\binom{D}{M}$ combinations of sensors to find the optimal set Ω.

To overcome this complexity, we relax the Boolean constraint on z_i to the *box constraint*

[†]Or find one of the solutions if many solutions exists.

$z_i \in [0,1]$ to arrive at the following problem

$$\underset{z\in[0,1]^D}{\text{minimize}} \sum_{1\le i<j\le N} \frac{{G_{ij}}^2}{G_{ii}\, G_{jj}},$$
$$\text{s.t.} \quad G = \Phi^T Z \Phi,\ \mathbf{1}^T z = M, \tag{4.6}$$

which supports an efficient gradient–projection algorithm to find an approximate solution. We develop this algorithm next.

4.3 The Insense algorithm

We now outline the steps that Insense takes to solve the problem (4.6). We slightly modify the objective of (4.6) to

$$f_\epsilon(z) = \sum_{1\le i<j\le N} \frac{{G_{ij}}^2 + \epsilon_1}{G_{ii}\, G_{jj} + \epsilon_2} \quad \text{where} \quad G = \Phi^T Z \Phi, \tag{4.7}$$

where the small positive constants $\epsilon_2 < \epsilon_1 \ll 1$ make the objective well-defined and bounded over $z \in [0,1]^D$.

The objective function (4.7) is smooth and differentiable but non-convex; the box constraints on z are linear. We minimize the objective using the iterative gradient-projection algorithm outlined in Alg. 2. The gradient $\nabla_z f \in \mathbb{R}^D$ can be computed using the multidimensional chain rule of derivatives [70] as

$$(\nabla_z f)_i = \left(\Phi \nabla_G f \Phi^T\right)_{ii} \quad \text{for} \quad i = 1, \ldots, D. \tag{4.8}$$

The $N \times N$ upper triangular matrix $\nabla_G f$ is the gradient of f in terms of the (auxillary)

variable G at the point $G = \Phi^T Z \Phi$, given by

$$(\nabla_G f)_{ij} = \begin{cases} \dfrac{2G_{ij}}{G_{ii}\ G_{jj} + \epsilon_2}, & i < j \\ -\displaystyle\sum_{\forall \ell \neq i} \dfrac{G_{\ell\ell}\left(G_{i\ell}^2 + \epsilon_1\right)}{\left(G_{ii}\ G_{\ell\ell} + \epsilon_2\right)^2}, & i = j \\ 0, & \text{elsewhere.} \end{cases} \tag{4.9}$$

The Insense algorithm (Alg. 2) proceeds as follows. First, the variables G and Z are initialized. Next, we perform the following update in iteration k,

$$z^{k+1} = P_{\text{SBS}}\left(z^k - \gamma^k \nabla_z f(z^k)\right), \tag{4.10}$$

where P_{SBS} denotes the projection onto the convex set defined by the scaled boxed-simplex (SBS) constraints $\mathbf{1}^T z = M$ and $z \in [0,1]^D$. For certain bounded step size rules (e.g., $\gamma^k \leq 1/L$, where L is the Lipschitz constant of $\nabla_z f$), the sequence $\{z^k\}$ generated by (4.10) converges to a critical point of the non-convex problem [71, 72]. In our implementation, we use a backtracking line search to determine γ^k in each step [72].

4.3.1 The SBS projection

We now detail our approach to solving the SBS projection problem

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & \frac{1}{2}\|z - y\|_2^2, \\ \text{subject to} \quad & \sum_i z_i = M \ \text{ and } \ z_i \in [0,1] \ \ \forall i = 1, \ldots, D. \end{aligned} \tag{4.11}$$

We emphasize that, for $M > 1$, the SBS projection problem is significantly different from the (scaled-)simplex constraint ($\sum_i z_i = M$) projection problem that has been studied in the literature [73, 74, 75], due to the additional box constraints $z_i \in [0,1]$.

The Lagrangian of the problem (4.11) can be written as

$$f(z, \lambda, \alpha, \beta) = \frac{1}{2}\|z - y\|_2^2 + \lambda \left(\sum_i z_i - M \right) + \sum_i \alpha_i(-z_i) + \sum_i \beta_i(z_i - 1),$$

where λ, α, β are Lagrange multipliers for the equality and inequality constraints, respectively. The Karush-Kuhn-Tucker (KKT) conditions are given by

$$z_i - y_i + \lambda - \alpha_i + \beta_i = 0, \forall i,$$

$$\sum_i z_i - M = 0,$$

$$\alpha_i(-z_i) = 0,\ \beta_i(z_i - 1) = 0,\ \alpha_i, \beta_i \geq 0,\ 0 \leq z_i \leq 1,\ \forall i.$$

According to the complimentary slackness condition for the box constraint $z_i \in [0, 1]$, we have the following three cases for x_i:

(a) $z_i = 0$: $\beta_i = 0, \alpha_i = y_i + \lambda > 0$,

(b) $z_i = 1$: $\alpha_i = 0, \beta_i = 1 - y_i - \lambda > 0$,

(c) $z_i \in (0, 1)$: $\alpha_i = \beta_i = 0, z_i = y_i + \lambda$.

Therefore, the value of λ holds the key to the proximal problem (4.11). However, finding λ is not an easy task, since we do not know which entries of z will fall on the boundary of the box constraint (and are equal to either 0 or 1).

In order to find the entries z_i that are equal to 0, we assume without loss of generality that the values of y are sorted in ascending order: $y_1 \leq y_2 \leq \ldots y_D$. We note that, in all three cases above, $z_i = \max(\min(y_i + \lambda, 1), 0)$. Therefore, $\sum_i z_i$ is a non-decreasing function of λ. We evaluate $\sum_i z_i$ at the following values of λ:

- $\lambda = -y_1$: $z_1 = 0, z_i = \min(y_i - y_1, 1)$ for $i \geq 2$,

Algorithm 2: Insense

Input: Φ

Output: $Z = \text{diag}(z)$

Initialization:

$z \leftarrow z_0$;

$G \leftarrow \Phi^T Z \Phi$;

while *stopping criterion = false* **do**

1. $k \leftarrow k + 1$;
2. update $\nabla_z f(z^k)$ based on equation (4.9);
3. $\gamma_k \leftarrow$ line search$(f, \nabla_z f(z^k), z^k)$;
4. $z^k \leftarrow z^k - \gamma^k \nabla_z f(z^k)$ {gradient step};
5. $z^{k+1} \leftarrow P_{\text{SBS}}(z^k)$ {SBS projection step};

end

- $\lambda = -y_2$: $z_1 = z_2 = 0$, $z_i = \min(y_i - y_1, 1)$ for $i \geq 3$,

  ...

- $\lambda = -y_D$: $z_1 = z_2 = \ldots = z_D = 0$.

Thus, the entries in z that are equal to 0 correspond to the first K_0 entries in y, where K_0 is the largest integer k such that $\sum_i \max(\min(y_i - y_k, 1), 0) \geq M$.

Similarly, we can find the entries in z that are equal to 1 by negating z and y in (4.11). Let $p = -y$ and assume that its entries are sorted in ascending order; a procedure similar to that above shows that the entries in z that are equal to 1 correspond to the first K_1 entries in p, where K_1 is the largest integer K such that $\sum_i \max(\min(p_i - p_k - 1, 0), -1) \geq -M$.

Knowing which entries in z are equal to 0 and 1, we can solve for the value of λ by

working with the entries with values in $(0, 1)$. Using case (c) above and denoting the index set of these entries by ζ, we have

$$\lambda = \frac{M - K_1 - \sum_{i \in \zeta} y_i}{|\zeta|},$$

and the solution to (4.11) is given by $z_i = \max(\min(y_i + \lambda, 1), 0)$.

4.4 Experiments

In this section, we experimentally validate Insense (Alg. 2) using a range of synthetic and real-world datasets. In all experiments, we set $\epsilon_1 = 10^{-9}$ and $\epsilon_2 = 10^{-10}$ (anything in the range $\epsilon_2 < \epsilon_1 \ll 1$ can be utilized). We terminate Insense when the relative change of the cost function $\mu^2_{\text{avg}}(\Phi_\Omega)$ drops below 10^{-7}.

4.4.1 Baselines and performance metrics

We compare Insense with several leading sensor selection algorithms, including *Convex Sensor Selection* [40], *Greedy Sensor Selection* [41], *EigenMaps* [51], and *FrameSense* [42]. We also compare with four greedy sensor selection algorithms that were featured in [42]. The first three minimize different information theoretic measures of the selected sensing matrix as a proxy to the MSE: the determinant in *Determinant-G* [55], mutual information (MI) in *MI-G* [56], and entropy in *Entropy-G* [57]. The final greedy algorithm, *MSE-G* [52, 53, 54], directly minimizes the MSE of the LS reconstruction error. The code for these baseline greedy algorithms were obtained from https://github.com/jranieri/OptimalSensorPlacement. We also compare with *Random*, a simple baseline that selects sensors at random.

We compare the sensor selection algorithms using the following six metrics:

- *Average coherence* $\mu_{\text{avg}}(\Phi_\Omega)$.

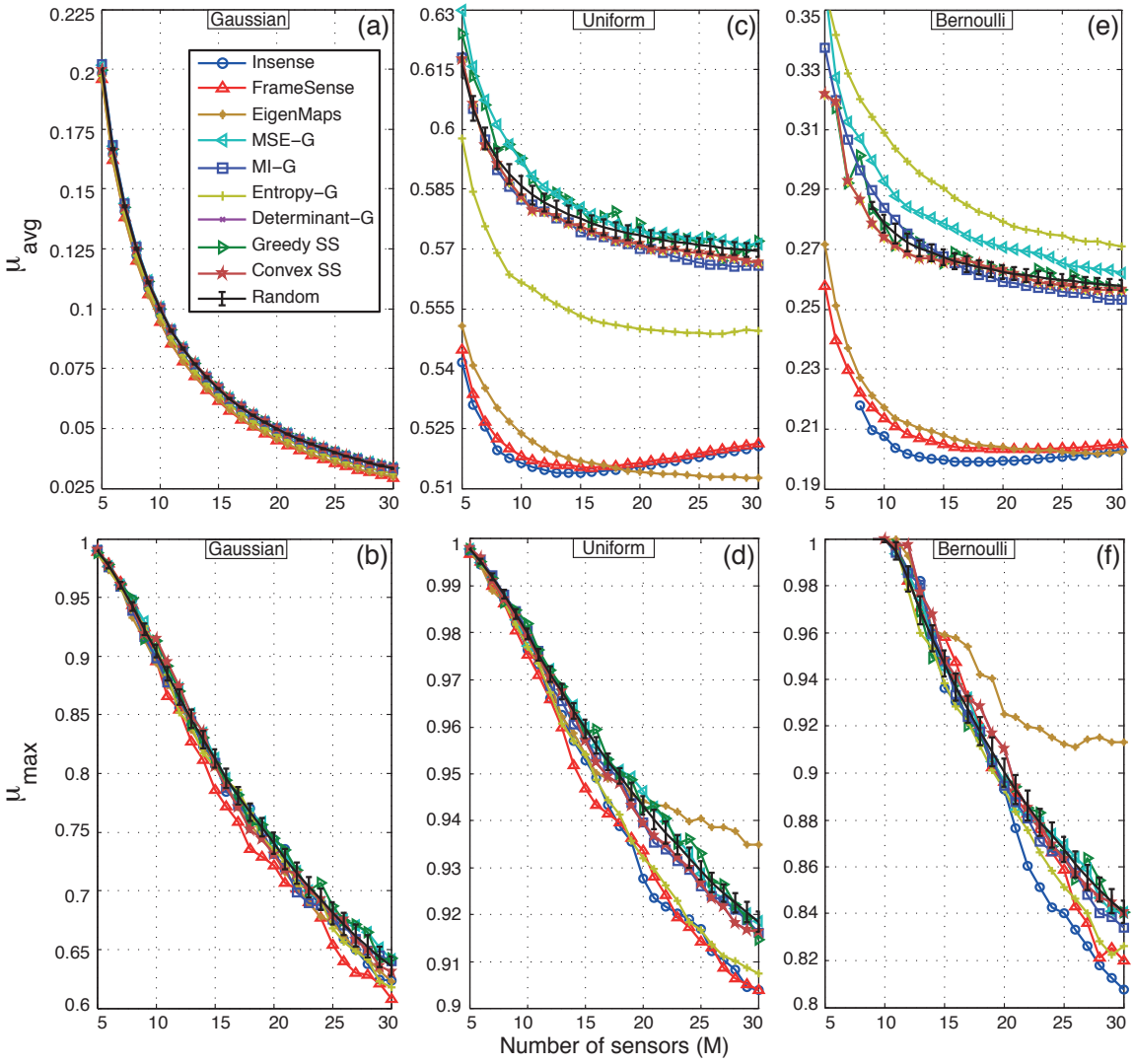

Figure 4.2 : Comparison of Insense against the baseline algorithms in minimizing the average coherence μ_{avg} and maximum coherence $\mu_{\max}$ of the selected sensing matrix Φ_Ω from random sensing matrices with independent Gaussian (a,b), Uniform (c,d), and Bernoulli (e,f) entries ($D = N = 100$). Results are averaged over 20 trials with different random matrices.

- *Maximum coherence* $\mu_{\max}(\Phi_\Omega)$.
- *Frame potential* FP(Φ_Ω) (see (4.3))
- *Condition number* CN(Φ_Ω).
- *BP recovery accuracy*.
- *Running time*

Depending on the task, in some experiments we only report a subset of the metrics. To compute BP recovery accuracy, we average the performance of the BP algorithm [76] over multiple trials. In each trial we first generate a K-sparse vector x whose non-zero entries are equal to one. We then use BP to recover x from linear, nonadaptive (noiseless) measurements $y = \Phi_\Omega x$.

We repeat the same experiment for all $\binom{N}{K}$ sparse vectors x with different support sets and report the percentage of trials that x has been exactly recovered. When $\binom{N}{K}$ is too large (here, greater than 10,000), we run the BP algorithm on a smaller random subset of all $\binom{N}{K}$ sparse vectors x.

4.4.2 Unstructured synthetic datasets

We first test the sensor section algorithms by applying them to random matrices. It is easy to show that asymptotically (i.e., when $N \to \infty$) random matrices do not favor certain rows (sensors) over others. In the non-asymptotic regime (i.e., when N is finite) the choice of sensors for sparse recovery might be critical, since the probability that certain sets of sensors significantly outperform others increases. In this case, the sensor selection algorithm aims to identify these high-performing sensors. We generate three types of random sensing

matrices Φ whose entries are drawn from Gaussian, Uniform, and Bernoulli distributions and compare the performance of Insense against the other baselines.

Random Gaussian matrix

We conduct this experiment for 20 random trials and generate 100×100 matrices Φ whose entries are independently drawn from a standard normal distribution. We use Insense and other baseline algorithms to select $M \in \{5, 6, 7, \ldots, 30\}$ sensors. In Fig. 4.2(a,b) we report μ_{avg} and $\mu_{\max}$ of the selected sub-matrices Φ_Ω, with $|\Omega| = M$. All of the sensor selection algorithms have comparable performance to the random sensor selection strategy (Random), illustrating that only small improvements to the maximum and average coherence can be achieved using these algorithms.

Random Uniform matrix

We repeat the previous experiment with a sensing matrix Φ whose entries are drawn uniformly at random from $[0, 1]$. Fig. 4.2(c) shows that Insense outperforms most of the baseline algorithms, including Random and Convex SS, in terms of μ_{avg}. Despite selecting completely different sensors, FrameSense and EigenMaps have comparable performance to Insense in minimizing μ_{avg}. Fig. 4.2(d) makes apparent the gap in maximum coherence $\mu_{\max}$ between that achieved by Insense and the other baselines.

Random Bernoulli matrix

We repeat the previous experiment with a sensing matrix Φ whose entries are 0 or 1 with equal probability.[‡] Fig. 4.2(e) shows that FrameSense, EigenMaps, and Insense have simi-

[‡]In our experiments, the coherence minimization performance of all of the sensor selection algorithms was similar for Bernoulli (1/-1) matrices and Gaussian matrices.

	$\mu_{\text{avg}(\Phi_\Omega)}$	FP(Φ_Ω)	CN(Φ_Ω)	BP accuracy %
Insense	**0.3061 ± 0.0047**	1019 ±313	1.93 ± 0.19	**92.27 ± 1.42**
FrameSense	–	**0.00 ± 0.00**	**1.00 ± 0.00**	4.00 ± 0.00
EigenMaps	–	**0.00 ± 0.00**	**1.00 ± 0.00**	4.00 ± 0.00
MSE-G	0.3872 ± 0.0305	1155 ± 374	11.51 ± 0.93	57.91 ± 1.09
MI-G	–	**0.00 ± 0.00**	**1.00 ± 0.00**	4.00 ± 0.00
Entropy-G	–	**0.00 ± 0.00**	**1.00 ± 0.00**	4.00 ± 0.00
Determinant-G	–	**0.00 ± 0.00**	**1.00 ± 0.00**	4.00 ± 0.00
Greedy SS	–	**0.00 ± 0.00**	**1.00 ± 0.00**	4.00 ± 0.00
Convex SS	0.3137 ± 0.0075	2279 ± 470	2.22 ± 0.25	88.64 ± 3.64

Table 4.1 : Comparison of Insense against the baseline algorithms on selecting $M = 10$ rows from a structured Identity/Gaussian Φ. Insense selects the set of sensors with the smallest μ_{avg} and achieves the best BP recovery performance.

lar performance and outperform the other algorithms by a large margin in terms of average coherence μ_{avg}.[§] Fig. 4.2(f) shows a clear gap between Insense and the other baselines in terms of the maximum coherence μ_{max}.

In summary, Insense selects reliable sensors that are consistently better than or comparable to the other baseline algorithms on random sensing matrices. This suggests that Insense could find application in designing sensing matrices that outperform random matrices for CS recovery tasks.

[§]When a selected matrices Φ_Ω contains one column with all zero entries, the average coherence μ_{avg} is not defined. The missing values in some curves correspond to these instances.

4.4.3 Highly structured synthetic datasets

In contrast to random matrices, the sensing matrices in real-world applications often have imposed structures or redundancies. In such cases, careful sensor selection can mean the difference between low and high performance. We now explore sensor selection with structured over-complete matrices by constructing two synthetic datasets that resembles the redundancies and structures in real-world datasets. Similar over-complete basis has been explored in [76].

Identity/Gaussian matrix

We construct our first highly structured dataset by concatenating two 50×50 matrices: An identity matrix and a random matrix with i.i.d. Gaussian entries. Such matrices feature prominently in certain real-world CS problems.

For instance, in the universal DNA-based microbial diagnostics platform studied in [1], the identity and Gaussian matrices symbolize two different types of sensors: the identity matrix corresponds to a set of sensors that are designed to be specific to a single microbial target (column) in the dictionary Φ, while the Gaussian matrix corresponds to a set of sensors that are universal for microbial targets in the dictionary. (We extend this simulation example with real experimental data below in Subsection 4.4.4.) As in [1], consider a bacterial detection scenario where the solution to the sparse recovery problem both detects and identifies the bacterial targets in a sample (through the support of the sparse vector x); here the goal is to maximize the average sparse recovery (detection) performance. On the one hand, if all of the sensors are selected from the identity submatrix, then nearly all of the selected sensors will lie dormant when detecting a particular bacterial target. On the other hand, if the sensors are selected from the Gaussian submatrix, then the selected sensors will work jointly to detect all bacterial targets, which provides both universality and better

average sparse recovery performance [1]. To achieve a better sparse recovery performance, the sensor selection algorithm should select rows (sensors) from the Gaussian submatrix rather than the identity submatrix.

Table 4.1 compares the performance of Insense to the baseline algorithms for the problem of selecting $M = 10$ rows from the structured Identity/Gaussian Φ. We repeat the same experiment 10 times with different random Gaussian matrices.[¶] In particular, Insense, Convex SS, and MSE-G are the only algorithms that select rows of the Gaussian sub-matrix. While achieving the minimum FP(Φ_Ω) ($= 0$), the other algorithms perform poorly on BP recovery. The greedy algorithms select rows from the identity matrix that result in columns with all-zero entries and thus fail to recover most of the entries in x. Digging deeper, Insense selects rows with smaller column coherence than Convex SS and MSE-G. As a result, Insense achieves the best BP recovery performance (Table 4.1) among these three algorithms.

In summary, this example demonstrates that minimizing a similarity metric imposed on the rows of the sensing matrix (such as frame potential, etc.) will not maximize the recovery performance of sparse signals. Our results also provide reassurance that the coherence among the columns of the sensing matrix is a useful performance objective.

Uniform/Gaussian matrix

To study the quality of the box constraint relaxation in (4.6), we compare Insense against the baseline algorithms for a matrix Φ where we know the globally optimal index set of rows (sensors) Ω. (For arbitrary Φ, global combinatorial optimization is computationally intractable when $D, N > 200$ or so.)

[¶]Dashes correspond to instances where the selected matrices Φ_Ω contain columns with all zero entries; here the average coherence μ_{avg} is undefined.

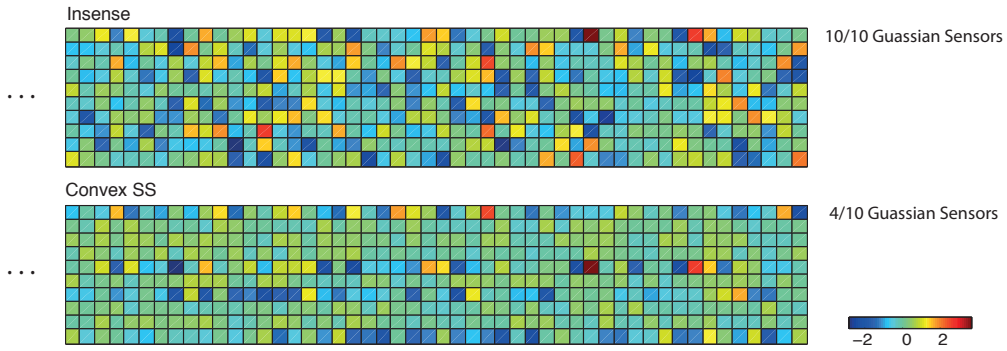

Figure 4.3 : Visualizations of the $M = 10$ sensing matrices Φ_Ω selected by Insense and Convex SS from a structured Uniform/Gaussian matrix. Insense selects 10/10 Gaussian rows (sensors), while Convex SS selects only 4/10 Gaussian rows.

We concatenate a 10×200 matrix with i.i.d. Gaussian entries and a 190×200 matrix with i.i.d. $[0, 1]$ uniform distribution entries. In this case, one would expect that the Gaussian submatrix has the lowest μ_{avg} when we set $M = 10$. Fig. 4.3 visualizes the results of running Insense and the Convex SS baseline algorithm on such a Φ. In all 10 random trials, Insense successfully selects all Gaussian rows and hence find the globally optimal set of sensors. FrameSense and EigenMaps miss, on average, 10–20% of the Gaussian sensors. The other baselines algorithms, including Convex SS, select only a small portion ($<$20%) of the Gaussian rows (sensors). Table 4.2 also indicates that Insense achieves better BP recovery performance, since it selects exclusively Gaussian rows, resulting in the minimum average coherence μ_{avg} of the resulting sensing matrix.

4.4.4 Real-world datasets

Finally, we assess the performance of Insense on two real-world datasets from microbial diagnostics and structural health monitoring.

	$\mu_{\text{avg}}(\Phi_\Omega)$	FP(Φ_Ω)	CN(Φ_Ω)	Gaussian sensor ratio %	BP accuracy %
Insense	**0.3165 ± 0.0023**	9320 ±3292	**1.46 ± 0.07**	**100 ± 0**	**58.55 ± 2.64**
FrameSense	0.3273 ± 0.0059	**6095 ± 1708**	3.19 ± 0.92	84 ± 5	58.15 ± 2.26
EigenMaps	0.3215 ± 0.0021	7230 ± 2319	2.07 ± 0.12	90 ± 0	57.60 ± 3.72
MSE-G	0.5805 ± 0.0440	78530 ± 12450	5.99 ± 0.31	17 ± 4	49.90 ± 3.54
MI-G	0.6814 ± 0.0556	93260 ± 109250	6.26 ± 0.77	7 ± 4	51.60 ± 5.21
Entropy-G	0.7007 ± 0.0804	98950 ± 16216	6.61 ± 0.48	5 ± 7	53.70 ± 5.21
Determinant-G	0.7303 ± 0.0545	105700 ± 11228	6.57 ± 0.31	3 ± 4	55.50 ± 4.50
Greedy SS	0.7303 ± 0.0545	105700 ± 11228	5.57 ± 0.31	3 ± 4	55.50 ± 4.50
Convex SS	0.5788 ± 0.1140	75270 ± 27383	5.97 ± 0.77	20 ± 15	54.40 ± 4.20

Table 4.2 : Comparison of Insense against the baseline algorithms on selecting $M = 10$ rows from a structured Uniform/Gaussian Φ. Insense selects the set of sensors with the smallest μ_{avg} and achieves the best BP recovery performance.

Microbial diagnostics

Microbial diagnostics seek to detect and identify microbial organisms in a sample. Next-generation systems detect and classify organisms using DNA probes that bind (hybridize) to the target sequence and emit some kind of signal (e.g., fluorescence). Designing DNA probes for microbial diagnostics is an important application of sensor selection in the underdetermined sensing regime. For example, in the universal microbial sensing (UMD) framework [1], DNA probes acquire linear measurements from a microbial sample (e.g., bacterial, viral, etc.) in the form of a fluorescence resonance energy transfer (FRET) signal that indicates the hybridization affinity of a particular DNA probe to the organisms present in the sample. Given a matrix Φ that relates the hybridization affinities of DNA probes to microbial species, the objective is to recover a sparse vector x comprising the concentrations of the organisms in the sample from as few linear measurements as possible.

	BP accuracy in detecting organisms %											
Number of organisms	$K = 2$				$K = 3$				$K = 5$			
Number of probes (M)	5	8	12	15	8	12	15	20	12	15	20	25
Insense	25.52	**68.33**	**94.78**	**99.65**	**26.46**	**71.74**	**93.95**	**99.53**	**16.78**	**51.95**	**92.71**	**99.10**
FrameSense	27.73	61.83	88.40	95.71	22.70	62.32	82.29	98.36	10.79	35.16	81.92	96.50
EigenMaps	14.97	49.65	84.69	94.66	13.17	54.68	78.09	96.25	6.69	27.47	72.13	95.30
MSE-G	27.26	60.79	91.53	97.91	22.01	67.16	89.15	98.40	14.69	43.26	83.52	97.40
MI-G	26.22	59.98	89.68	96.40	20.96	65.69	84.10	97.39	13.49	37.96	79.72	96.00
Entropy-G	**27.96**	61.25	91.53	98.61	21.51	66.35	88.96	99.19	14.19	42.86	89.61	97.50
Determinant-G	14.85	46.75	82.13	94.55	12.49	48.97	76.13	96.03	6.29	24.48	72.73	92.81
Greedy SS	25.52	57.54	87.70	96.87	19.72	59.65	84.64	97.34	10.99	36.16	80.22	94.11
Convex SS	15.89	53.36	87.94	98.94	14.29	57.58	87.59	98.89	7.69	38.46	83.52	98.40
Random	25.57	61.53	88.79	96.66	22.37	62.29	86.15	97.72	12.79	38.88	82.94	86.44

Table 4.3 : Comparison of Insense against the baseline algorithms on selecting M DNA probes to identify pathogenic samples containing K bacterial organisms. Insense selects the sets of DNA probes that achieve the best pathogen identification performance.

We run Insense and the baseline sensor selection algorithms on a large sensing matrix comprising the hybridization affinity of $D = 100$ random DNA probes to $N = 42$ bacterial species (as described in [1]). For each algorithm, after selecting M probes and constructing a sensing matrix Φ_Ω with $|\Omega| = M$, we perform BP recovery for multiple sparse vectors x with random support (corresponding to the presence of a random subset of bacterial organisms). We repeat the same experiment for all $\binom{N}{K}$ sparse vectors x with $K = \{2, 3, 5\}$ non-zero elements (i.e., bacteria present) and report the average BP recovery performance on identifying the composition of the samples in Table 4.3. (To report the BP recovery for $K = 5$ organisms, we randomly generate 1000 realizations of the sparse vector x with 5 active elements and average the BP recovery performance on selected samples.)

The DNA probes selected by Insense outperform all of the baseline algorithms in identifying the bacterial organisms present. Specifically, Insense requires a smaller number of DNA probes than the other algorithm to achieve almost perfect detection performance (BP accuracy $> 99\%$), suggesting that Insense is the most cost-efficient algorithm to select DNA probes for this application. Moreover, the performance gap between Insense and the other algorithms grows as the number of bacterial species present in the sample K increases, indicating that Insense has better recovery performance in complex biological samples.

Structural health monitoring

Structural health monitoring (SHM) studies the problem of detecting, localizing, and characterizing damage (e.g., cracks, holes, etc.) occurring on or inside structures such as buildings, bridges, pipes, etc. [67]. Typical SHM systems continuously monitor a target structure by transmitting signals that propagate throughout the structure and then analyzing the reflected signals measured using an array of sensors. Classical methods for damage

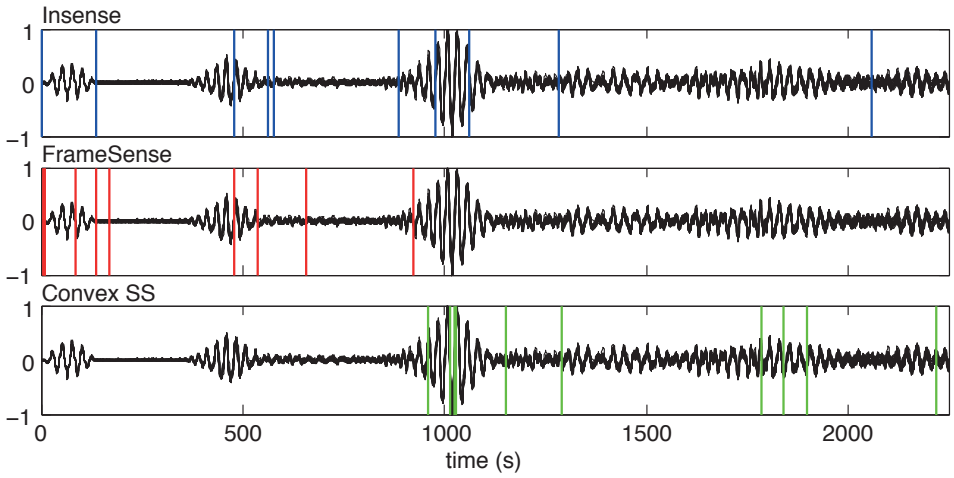

Figure 4.4 : Visualizations of the $M = 10$ time instances selected by Insense (blue lines), FrameSense (red lines), and Convex SS (red lines) from a matrix for the crack localization problem. Insense selects time instances that are more distributed along the time axis and variant along the intensity axis.

	$\mu_{\text{avg}}(\Phi_\Omega)$						BP accuracy in localizing cracks %						Running time (s)
Number of samples (M)	5	6	7	8	9	10	5	6	7	8	9	10	10
Insense	**0.2491**	**0.2251**	**0.2185**	**0.2127**	**0.2075**	**0.2030**	**40.99**	**60.49**	**76.49**	**86.09**	**91.39**	**94.64**	4.12
FrameSense	–	0.4051	0.3843	0.3179	0.2717	0.2640	24.41	51.58	63.73	81.92	89.04	91.14	0.18
EigenMaps	0.4341	0.4102	0.4028	0.3950	0.3841	0.3699	25.13	41.16	40.55	51.38	64.52	80.45	0.11
MSE-G	0.9832	0.9779	0.9752	0.9748	0.9707	0.9637	39.50	53.50	63.73	79.18	86.28	93.50	0.97
Entropy-G	0.9694	0.9649	0.9615	0.9568	0.9525	0.9507	40.89	58.95	71.35	80.98	86.50	92.36	0.16
Greedy SS	–	–	–	–	–	–	9.32	14.96	19.55	28.36	45.77	57.43	1.65
Convex SS	0.9702	0.9685	0.9611	0.9571	0.9519	0.9451	38.92	59.56	72.42	82.83	89.98	94.56	4.52
Random	0.7980	0.7976	0.8083	0.8147	0.8111	0.8132	33.84	52.20	67.51	79.60	88.19	93.75	$< 10^{-4}$

Table 4.4 : Comparison of Insense against the baseline algorithms on sampling M time instances that best localize cracks on a plane in a structural health monitoring problem. Insense selects the set of sensors with the smallest μ_{avg} and achieves the best damage localization performance.

localization calculate the time-of-flight (TOF) and estimate the location of a crack using triangulation, which is computationally intensive.

Dictionary-based SHM methods sidestep the computational complexity of TOF methods by constructing a matrix of signal profiles, where each column corresponds to the impulse response of the structure with a crack at a certain location on a predefined grid [68, 69]. The locations of a small number of cracks in the structure are then determined by solving a sparse recovery problem, where the support of the sparse signal vector x in equation (4.1) corresponds to the damage locations [69]. In dictionary-based SHM methods, each row of the sensing matrix Φ corresponds to a measured time instance of a reflected signals (see Fig. 4.4). Applying a sensor selection algorithms to the SHM matrix will reduce the number of measurements required while preserving the damage localization performance.

To compare Insense against the other baseline algorithms, we use real SHM data from a crack localization experiment on a metal plate with 25 potential cracks located on a 5×5 square grid. The experimentally measured signals of size $D = 2250$ from all cracks are obtained and stored in a matrix $\Phi \in \mathbb{R}^{2250 \times 25}$. The crack locations are then identified by solving the sensor selection problem with the experimentally measured sensing matrix Φ. Fig. 4.4 visualizes the $M = 10$ selected time instances for Insense and two other baseline algorithms, and Table 4.4 showcases their performance on BP recovery in localizing $K = 2$ cracks using $M \in \{5, 6, 7, 8, 9, 10\}$ time instances. MI-G and Determinant-G does not run on our desktop PC with 12 GB of RAM due to high memory requirements. Insense ran faster than the optimization-based algorithm Convex SS and only slightly slower than the greedy algorithms.

In summary, in this SHM application, Insense consistently outperforms the baseline algorithms in terms of crack localization accuracy and reduces the number of time mea-

surement instances required for accurate localization. We note that the measurements are not statistically independent in this dataset, since they correspond to measurements of the same reflected signal at different time instances, highlighting the superior performance of Insense in applications where the independence assumption in the sensing model is slightly violated.

4.5 Conclusions

In this paper, we developed the Incoherent Sensor Selection (Insense) algorithm for the underdetermined sensor selection problem that optimizes the average squared coherence of the columns of the selected sensors (rows) via a computationally efficient relaxation. Our synthetic and real-world data results have both verified the utility of the average squared coherence metric and the performance of the Insense algorithm. In particular, Insense provides superior performance than existing state-of-the-art sensor selection algorithms, especially in the real-world problems of microbial diagnostics and structural health monitoring.

There are a number of avenues for future work. One is finding computational shortcuts for operating on large-scale sensing matrices. Another is exploring different variants of the Insense algorithm that select sensors under generalized sparse models, e.g., group-sparse, tree-sparse and low-rank models.

Chapter 5

RHash: Robust Hashing via ℓ_∞-norm Distortion*

5.1 Introduction

Hashing, one of the primitive operations in large-scale machine learning systems, seeks a low-dimensional binary embedding of a high-dimensional dataset. Such a binary embedding can increase the computational efficiency of a variety of tasks, including nearest neighbor (NN) search.

The embedding of high-dimensional data into short binary codes naturally distorts the set of top NNs of each data point. To minimize this distortion, a range of hashing schemes have been developed. The simplest, *random projection*, projects the data onto a lower-dimensional random subspace and then quantizes to binary values. Random projection belongs to the family of probabilistic, locality sensitive hashing (LSH) algorithms that guarantee the preservation of the set of NN points with small distortion [77]. The cost, however, is that LSH requires an impractically high number of bits for indexing real-world, large-scale datasets [78]. In response, *data-dependent* binary hashing algorithms have been developed that require significantly fewer bits; see [79] for a survey.

Data-dependent hashing algorithms learn a hash function by minimizing a measure of distortion between certain pairs of data points [80, 81]. For the task of NN search, it is sufficient to control the pairwise distortion among the NNs of each point plus a few additional points [82]. Given a pair of data points $(\mathbf{x}_i, \mathbf{x}_j)$ in the ambient space and their

*This chapter is a revised version of an under review article [3]

corresponding embedded binary codes $(\mathbf{h}_i, \mathbf{h}_j)$ in the Hamming space, define the *pairwise distortion function* $\delta_{ij} = L(\mathbf{x}_i, \mathbf{x}_j; \mathbf{h}_i, \mathbf{h}_j)$, where $L(.)$ is a chosen *loss function*. The lion's share of data-dependent hashing algorithms use an ℓ_2-norm loss function which minimizes the *average* of the square of pairwise distortions $\|\boldsymbol{\delta}\|_2^2 = \sum_{(i,j)\in\Omega} \delta_{ij}^2$ over a set of selected pairs of points Ω [83, 84, 85, 82, 86, 80, 81].

Unfortunately, the loss functions employed by the hashing community to date, including the ℓ_2-norm loss, are not *robust* to perturbations of the data points. Indeed, the ℓ_2-norm average distortion is well-known to be sensitive to small disturbances and noise [87]. In this paper, we argue that alternative norms, in particular the ℓ_∞-norm [88, 89, 90], are more robust to small perturbations and more appropriate for designing data-dependent hash functions.

To elucidate the strong effect of the loss function on the robustness of a hashing function, we compare the hash functions learned using the ℓ_2- and ℓ_∞-norm losses on a subset of the MNIST dataset of handwritten digit images [91]. For visualization purposes, we project the data points onto the first two principal components and compute the optimal 1-dimensional binary embeddings generated by minimizing the ℓ_2-norm distortion and the ℓ_∞-norm distortion $\|\boldsymbol{\delta}\|_\infty = \max_{(i,j)\in\Omega} |\delta_{ij}|$ with $|\Omega| = 50$. Figure 5.1(a) shows the ℓ_2- and ℓ_∞-optimal embeddings that we learned from a grid search. Figure 5.1(b) summarizes an experiment where we add a small amount of white Gaussian noise to each data point before computing the optimal embeddings. We see that the ℓ_2-optimal embedding is increasingly less robust than the ℓ_∞-optimal embedding as the amount of noise grows.

The robustness of the ℓ_∞-optimal embedding translates directly to improved NN preservation. Continuing the MNIST example from above but with no added noise, Figure 5.1(c) illustrates that the top 5 NNs of the query point q are severely damaged by the ℓ_2-optimal embedding but not by the ℓ_∞-optimal embedding. Figure 5.1(d) provides some insight into

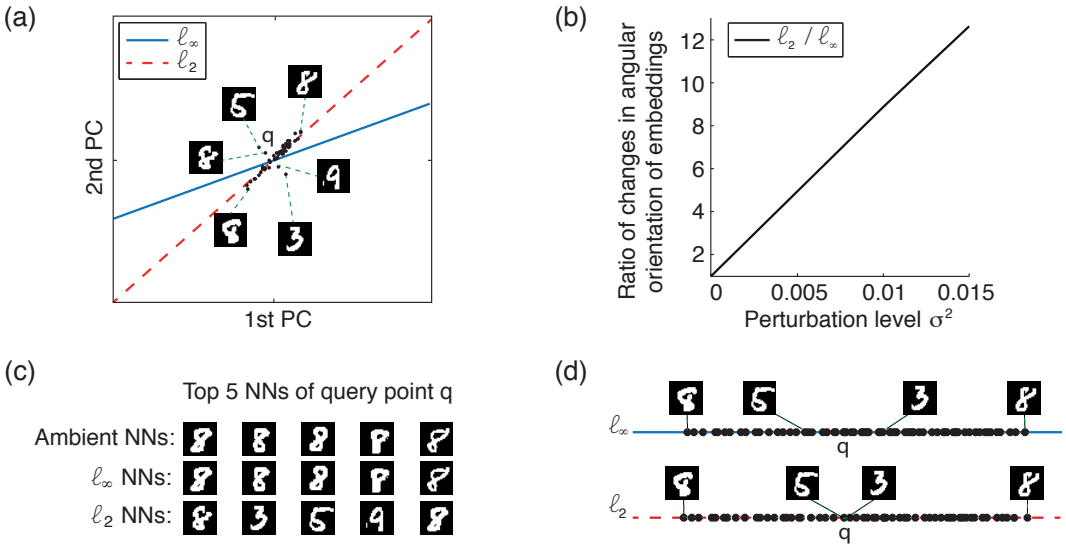

Figure 5.1 : Comparison of the robustness and nearest neighbor (NN) preservation performance of embeddings based on minimizing the ℓ_2-norm (average distortion) vs. the ℓ_∞-norm (worst-case distortion) on a subset of the MNIST handwritten digit dataset projected onto its first two principal components. The subset consists of the 50 NN (in the 2-dimensional ambient space) of the centroid q of the cluster of "8" digits. (a) Optimal embeddings for both distortion measures computed using a grid search over the orientation of the line representing the embedding. (b) Robustness of the embedding orientations to the addition of a small amount of white Gaussian noise to each data point. This plot of the mean square error of the orientation of the ℓ_2-optimal embedding divided by the mean square error of the orientation of the ℓ_∞-optimal embedding indicates that the latter embedding is significantly more robust to perturbations in the data points. (c) Comparison of the top 5 NNs of the query point q obtained in the ambient space using the ℓ_∞- and ℓ_2-optimal embeddings (no added noise). (d) Projections of the data points onto the ℓ_∞- and ℓ_2-optimal embeddings (no added noise).

the reason. The ℓ_2-norm ("average" distortion) optimal embedding does not preserve some of the distances between the digits "3," "5", and "9" and the digit "8" and thus projects some of them into the same neighborhood close to q. In contrast, the ℓ_∞-norm ("worst-case" distortion) optimal embedding preserves these *harder* distances and thus preserves the set of original NNs of q with minimal distortion.

Inspired by the robustness of the ℓ_∞-norm distortion [88, 89, 90], we develop *Robust Hashing* (RHash), the first hashing algorithm that learns robust binary hash codes by minimizing the ℓ_∞-norm of the pairwise distances vector $\|\boldsymbol{\delta}\|_\infty = \max_{(i,j)\in\Omega} |d_H(\mathbf{h}_i, \mathbf{h}_j) - d(\mathbf{x}_i, \mathbf{x}_j)|$, where d and d_H denote the Euclidean and Hamming distances, respectively. The robustness of ℓ_∞-norm distortion stems from its dependence on only the pairwise distortions that are hardest to preserve. As long as the hardest pairwise distortions are not affected, small perturbations do not alter the outcome of the embedding. In other words, in the NN retrieval task, where we only care about the distortions of NN pairs, the hardest pair to preserve is the deal-breaker. We emphasize that the worst-case distorted *NN pair* (under an embedding) is the hardest and not the farthest or an outlier pair.

5.1.1 Contributions

We make four distinct contributions in this paper. First, conceptually, we advocate minimizing the more robust worst-case ℓ_∞-norm distortion measure rather than the more prosaic ℓ_2-norm distortion.

Second, algorithmically, since ℓ_∞-norm minimization problems are computationally challenging, especially for large datasets, we develop two accelerated and scalable algorithms to find the optimal robust embedding. The first, *Robust Hashing* (RHash), is based on the alternating direction method of multipliers (ADMM) framework [92]. The second, *RHash-CG*, is based on an accelerated greedy extension of the RHash algorithm using the

concept of *column generation* [93]. RHash-CG can rapidly learn hashing functions from large-scale data sets that require the preservation of *billions* of pairwise distances.

Third, theoretically, since current data-dependent hashing algorithms do not offer any probabilistic guarantees in terms of preserving NNs, we develop new theory to prove that, under natural assumptions regarding the data distribution and with a notion of hardness of NN search (dubbed as k*-order gap*), RHash preserves the top NNs with high probability.

Fourth, experimentally, we demonstrate the superior performance of RHash over ten state-of-the-art binary hashing algorithms using an exhaustive set of experimental evaluations involving six diverse datasets and three different performance metrics (distance preservation, Hamming distance ranking, and Kendall τ ranking performance). In particular, we show that RHash achieves the same retrieval performance as state-of-the-art algorithms in terms of average precision *while using up to* 60% *fewer bits.* Our experiments clearly show the advantages of the ℓ_∞-norm formulation compared to the more classical ℓ_2-norm formulation that underlies many hashing algorithms [85, 82]. Our formulation also outperforms recently developed techniques that assume more structure in their hash functions [94, 95].

5.2 Robust hashing via ℓ_∞-norm distortion

In this section, we formulate the RHash algorithm using ℓ_∞-norm distortion. We pick a simple formulation for data dependent binary hash function which embeds a data point $\mathbf{x} \in \mathbb{R}^N$ into the low-dimensional Hamming space $\mathcal{H} = \{0,1\}^M$ by first multiplying it by an *embedding matrix* $\mathbf{W} \in \mathbb{R}^{M \times N}$ and then quantizing the entries of the product $\mathbf{Wx}$ to binary values: $h(\mathbf{Wx}) = (1 + \text{sgn}(\mathbf{Wx}))/2$. The function $\text{sgn}(\cdot)$ operates element-wise on the entries of $\mathbf{Wx}$, transforming the real-valued vector $\mathbf{Wx}$ into a set of binary codes depending on the sign of the entries in $\mathbf{Wx}$.

5.2.1 Problem formulation

Consider the design of a binary embedding f that maps Q high-dimensional data vectors $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_Q\}$ in the ambient space $\mathbb{R}^N$ into low-dimensional binary codes $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_Q\}$ in the Hamming space with $\mathbf{h}_i \in \{0,1\}^M$, where $\mathbf{h}_i = f(\mathbf{x}_i)$, $i = 1, \ldots, Q$, and $M \ll N$. Define the distortion of the embedding by

$$\delta = \inf_{\lambda>0} \sup_{(i,j)\in\Omega} |\lambda d_H(\mathbf{h}_i, \mathbf{h}_j) - d(\mathbf{x}_i, \mathbf{x}_j)|, \tag{5.1}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ denotes the Euclidean distance between the data points $\mathbf{x}_i$, $\mathbf{x}_j$, $d_H(\mathbf{h}_i, \mathbf{h}_j)$ denotes the Hamming distance between the binary codes $\mathbf{h}_i$ and $\mathbf{h}_j$, Ω indexes the set containing a selected pair of points (more details in section 5.3), and λ is a positive scaling variable. The distortion δ measures the worst-case deviation from perfect distance preservation [96]. Define the *secant set* $\mathcal{S}(\mathcal{X})$ as $\mathcal{S}(\mathcal{X}) = \{\mathbf{x}_i - \mathbf{x}_j : (i,j) \in \Omega\}$, i.e., the set of pairwise difference vectors indexed by Ω. Let $|\mathcal{S}(\mathcal{X})| = |\Omega|$ denote the size of the secant set. We formulate our robust hashing algorithm as minimizing the worst-case distortion parameter δ using the following optimization problem:

$$\underset{\mathbf{W},\lambda>0}{\text{minimize}} \sup_{(i,j)\in\Omega} \left| \lambda \|h(\mathbf{W}\mathbf{x}_i) - h(\mathbf{W}\mathbf{x}_j)\|_2^2 - \|\mathbf{x}_i - \mathbf{x}_j\|_2 \right|. \tag{5.2}$$

Note that the squared ℓ_2-distance between a pair of binary codes is equivalent to their Hamming distance up to a scaling factor that can be absorbed into λ. We can rewrite the above optimization problem as

$$(\mathbf{P}^*) \quad \underset{\mathbf{W},\lambda>0}{\text{minimize}} \ \|\lambda \mathbf{v}'(\mathbf{W}) - \mathbf{c}\|_\infty, \tag{5.3}$$

where $\mathbf{v}' \in \mathbb{N}^{|\Omega|}$ is a vector containing the pairwise Hamming distances between the embedded data vectors $\|h(\mathbf{x}_i) - h(\mathbf{x}_j)\|_2^2$, and $\mathbf{c}$ is a vector containing the pairwise ℓ_2-distances between the original data vectors. Intuitively, the ℓ_∞-norm objective optimizes the *worst-case* distortion among NN's distances.

The problem (P*) is a combinatorial problem with complexity $\mathcal{O}(Q^{2M})$. Following the standard approach [82], we approximate the hash function $h(\cdot)$ by the sigmoid function $\sigma(x) = (1 + e^{-x})^{-1}$. This enables us to approximate (P*) by the following optimization problem with smooth objective:

$$\text{(P)} \quad \underset{\mathbf{W},\lambda>0}{\text{minimize}} \ \|\lambda \mathbf{v}(\mathbf{W}) - \mathbf{c}\|_\infty, \tag{5.4}$$

where $\mathbf{v} \in \mathbb{R}_+^{|\Omega|}$ is a vector containing the pairwise ℓ_2 distances between the embedded data vectors after sigmoid relaxation $\left\|(1 + e^{-\mathbf{W}\mathbf{x}_i})^{-1} - (1 + e^{-\mathbf{W}\mathbf{x}_j})^{-1}\right\|_2^2$. The sigmoid function operates element-wise on $\mathbf{W}\mathbf{x}_i$. In practice we use a more general definition of the sigmoid function, defined as $\sigma_\alpha(x) = (1 + e^{-\alpha x})^{-1}$, where α is the *rate* parameter controlling how closely it approximates the non-smooth function $h(\cdot)$. We formally characterize the quality of such an approximation in Lemma 1 of the Supplementary Materials.

5.2.2 The RHash algorithm

We now outline the steps that RHash algorithm takes to solve the optimization problem (P). We apply the alternating direction method of multipliers (ADMM) framework [92] to construct an efficient algorithm to solve (P). We introduce an auxiliary variable $\mathbf{u}$ to arrive at the equivalent problem:

$$\underset{\mathbf{W},\mathbf{u},\lambda>0}{\text{minimize}} \ \|\mathbf{u}\|_\infty \quad \text{subject to} \quad \mathbf{u} = \lambda \mathbf{v}(\mathbf{W}) - \mathbf{c}. \tag{5.5}$$

The augmented Lagrangian form of this problem can be written as $\underset{\mathbf{W},\mathbf{u},\lambda>0}{\text{minimize}} \ \|\mathbf{u}\|_\infty + \frac{\rho}{2}\|\mathbf{u} - \lambda \mathbf{v}(\mathbf{W}) + \mathbf{c} + \mathbf{y}\|_2^2$, where ρ is the scaling parameter in ADMM and $\mathbf{y} \in \mathbb{R}^{|\Omega|}$ is the Lagrange multiplier vector. The RHash algorithm proceeds as follows. First, the variables $\mathbf{W}$, λ, $\mathbf{u}$, and Lagrange multipliers $\mathbf{y}$ are initialized randomly. Then, at each iteration, we optimize over each of the variables $\mathbf{W}, \mathbf{u}$, and λ while holding the other variables fixed. More specifically, in iteration ℓ, we perform the following four steps until convergence:

- *Optimize over* $\mathbf{W}$ via $\mathbf{W}^{(\ell+1)} \leftarrow \arg\min_{\mathbf{W}} \frac{1}{2} \sum_{(i,j)\in\Omega} (u_{ij}^{(\ell)} - \lambda^{(\ell)} \| \frac{1}{1+e^{-\mathbf{W}\mathbf{x}_i}} - \frac{1}{1+e^{-\mathbf{W}\mathbf{x}_j}} \|_2^2 + \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + y_{ij}^{(\ell)})^2$, where $\lambda^{(\ell)}$ denotes the value of λ in the ℓ^{th} iteration. We also use $u_{ij}^{(\ell)}$ and $y_{ij}^{(\ell)}$ to denote the entries in $\mathbf{u}^{(\ell)}$ and $\mathbf{y}^{(\ell)}$ that correspond to the pair $\mathbf{x}_i$ and $\mathbf{x}_j$ in the dataset $\mathcal{X}$. We show in our experiments below that using the accelerated first-order gradient descent algorithm [97] to solve this subproblem results in good convergence performance.

- *Optimize over* $\mathbf{u}$ while holding the other variables fixed; it corresponds to solving the proximal problem of the ℓ_∞-norm $\mathbf{u}^{(\ell+1)} \leftarrow \arg\min_{\mathbf{u}} \|\mathbf{u}\|_\infty + \frac{\rho}{2} \|\mathbf{u} - \lambda^{(\ell)} \mathbf{v}^{(\ell+1)} + \mathbf{c} + \mathbf{y}^{(\ell)}\|_2^2$. We use the low-cost algorithm described in [98] to perform the proximal operator update.

- *Optimize over* λ while holding the other variables fixed; it corresponds to a positive least squares problem, where λ is updated as $\lambda^{(\ell+1)} \leftarrow \arg\min_{\lambda>0} \frac{1}{2} \|\mathbf{u}^{(\ell+1)} - \lambda \mathbf{v}^{(\ell+1)} + \mathbf{c} + \mathbf{y}^{(\ell)}\|_2^2$. We perform this update using the non-negative least squares algorithm [99].

- *Update* $\mathbf{y}$ via $\mathbf{y}^{(\ell+1)} \leftarrow \mathbf{y}^{(\ell)} + \eta(\mathbf{u}^{(\ell+1)} - \lambda^{(\ell+1)} \mathbf{v}^{(\ell+1)} + \mathbf{c})$, where the parameter η controls the dual update step size.

5.2.3 Empirical convergence of the RHash algorithm

Figure 5.3 shows the empirical loss and the actual distortion parameter δ as a function of the iteration count ℓ in the RHash algorithm. The behavior of the empirical loss function closely matches that of δ as they gradually converge. The curve empirically confirms that minimizing the loss function in each iteration of RHash (using the ADMM framework) directly penalizes the non-convex loss function (distortion parameter δ). After initializing the RHash algorithm with random entries for $\mathbf{W}$, the value of the distortion parameter δ significantly drops in the first few iterations, empirically conforming that RHash learns an embedding with significantly lower distortion parameter after a few iterations.

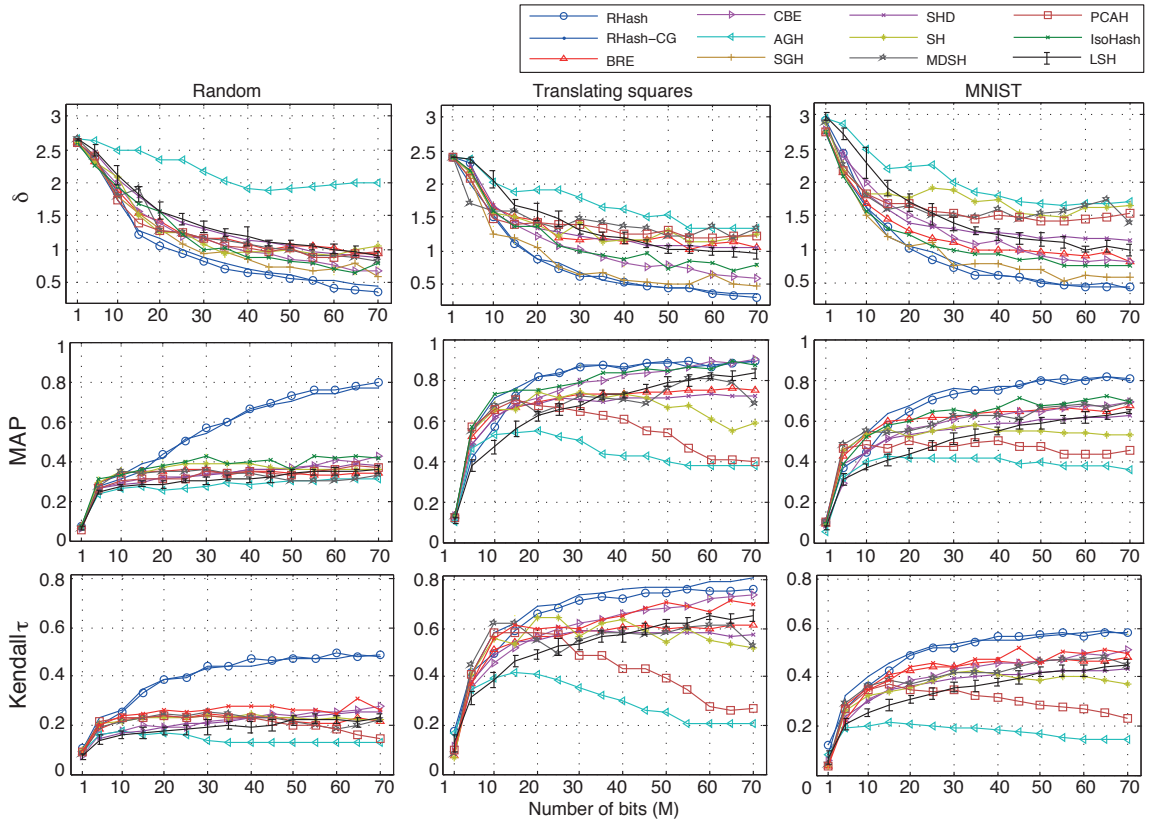

Figure 5.2 : Comparison of the RHash and RHash-CG algorithms against several baseline binary hashing algorithms using three small-scale datasets (Q = 100). The performance of RHash-CG closely follows that of RHash, and both outperform all of the other algorithms in terms of the maximum distortion δ (superior NN distance preservation), mean average precision MAP of training samples (superior retrieval performance), and Kendall τ rank correlation coefficient (superior ranking preservation).

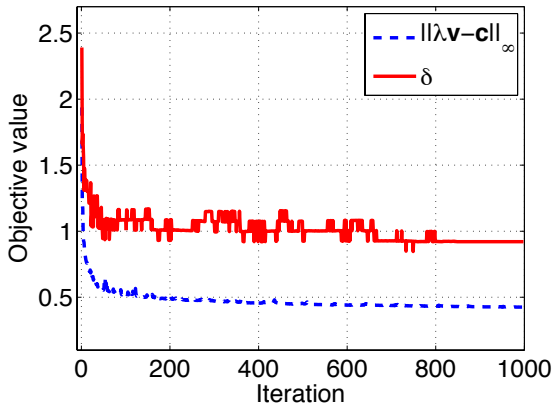

Figure 5.3 : Empirical convergence behavior of the RHash algorithm. Both the maximum distortion parameter δ and the loss function $||\lambda \mathbf{v} - \mathbf{c}||_\infty$ that approximates δ gradually decrease and converge as the number of iterations increases. We see that the loss function of RHash closely matches the behavior of the actual distortion parameter in each iteration.

5.2.4 Accelerated RHash for large-scale datasets

The ADMM-based RHash algorithm is efficient for small-scale datasets (e.g., for secant sets of size $|\mathcal{S}(\mathcal{X})| < 5000$ or so). However, the memory requirement of RHash is quadratic in $|\mathcal{X}|$, which would be problematic for applications involving large-scale numbers of data points and secants. In response, we develop an algorithm that approximately solves (P) while scaling very well to large-scale problems. The key idea comes from classical results in optimization theory related to *column generation* (CG) [93, 100].

The optimization problem (5.5) is an ℓ_∞-norm minimization problem with an equality constraint on each secant. The Karush-Kuhn-Tucker (KKT) condition for this problem states that, if strong duality holds, then the optimal solution is entirely specified by a (typically very small) portion of the constraints. Intuitively, the secants corresponding to these constraints are the pairwise distances of kNNs that are harder to preserve in the low-dimensional Hamming space. We call the set of such secants the *active* set. In order to solve (P), it suffices to find the active secants and solve RHash with a much smaller number

of active constraints. To leverage the idea of the active set, we iteratively run RHash on a small subset of all the secants that violate the distance preservation, as detailed below:

- Solve (P) with a small random subset $\mathcal{S}_0$ of all the secants $\mathcal{S}(\mathcal{X})$ using RHash to obtain $\widehat{\mathbf{W}}$, $\hat{\delta}$, and $\hat{\lambda}$, initial estimates of the parameters. Identify the active set $\mathcal{S}_{\mathrm{a}}$. Fix $\lambda = \hat{\lambda}$ for the rest of the algorithm.

- Randomly select a new subset $\mathcal{S}_{\mathrm{v}} \subset \mathcal{S}$ of secants that violate the distance preservation condition using the current estimates of $\widehat{\mathbf{W}}$, $\hat{\delta}$, and $\hat{\lambda}$. Then, form an augmented secant set $\mathcal{S}_{\mathrm{aug}} = \mathcal{S}_{\mathrm{a}} \cup \mathcal{S}_{\mathrm{v}}$.

- Solve (P) with the secants in the set $\mathcal{S}_{\mathrm{aug}}$ using the RHash algorithm.

We dub this approach *RHash-CG*. RHash-CG iterates over the above steps until no new violating secants are added to the active set. Since the algorithm searches over all the secants for violating secants in each iteration before terminating, RHash-CG ensures that all of the constraints are satisfied when it terminates.

A key benefit of RHash-CG is that only the set of active secants (and not all the secants) needs to be stored in memory. This benefit leads to significant improvements in terms of memory complexity over competing algorithms, since the set of all secants quickly becomes large-scale and can exceed the system memory capacity in large-scale applications. We note that the notion of active secants can not be defined in hashing algorithms based on ℓ_2-norm distortion. In fact the ℓ_∞-norm distortion function has enabled the RHash algorithm to leverage the simple CG approach mentioned above in order to scale to larger datasets, which can be listed as another advantage of ℓ_∞-norm in large-scale hashing.

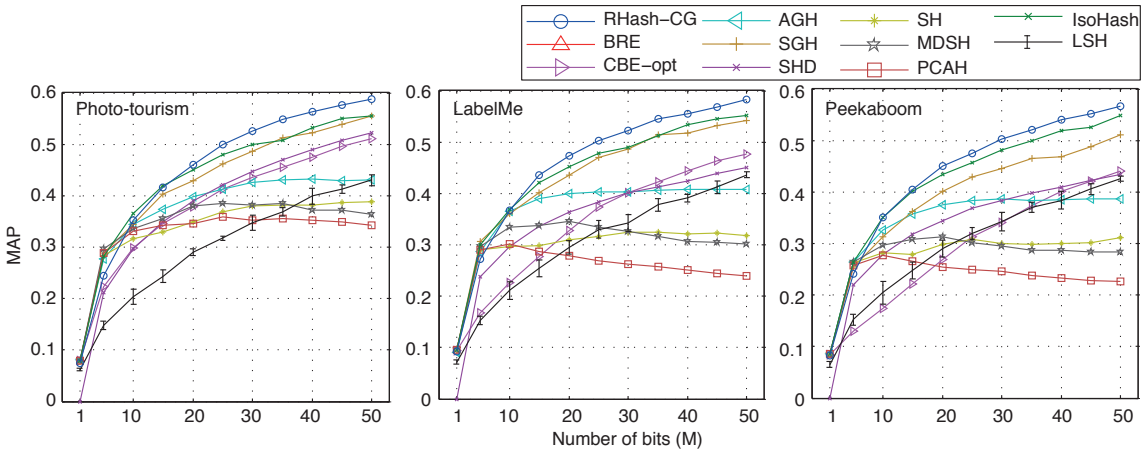

Figure 5.4 : Hamming ranking performance comparison on three medium-scale datasets ($Q = 1000$). The top-50 neighbors are used to report MAP over a test data of same size.

5.2.5 RHash and NN preservation guarantee

Now that we have fully described the RHash algorithm, we develop a theory that bounds the probability of preserving all kNNs as a function of the ℓ_∞-norm distortion measure δ minimized in the RHash algorithm. Inspired by the definition of *relative contrast* [101], we define a more generalized measure of data separability to preserve kNN that we call the *k-order gap* $\Delta_k := d(\mathbf{x}_0, \mathbf{x}_{k+1}) - d(\mathbf{x}_0, \mathbf{x}_k)$, where $\mathbf{x}_0$ is a query point and $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are its k^{th} and $k+1^{\text{th}}$ NNs, respectively. We formally show that if the data is sufficiently separable (Δ_k is large), then RHash preserves all kNNs with high probability under natural assumption in NN search literature [101].

Theorem 1 Assume that all the data points are independently generated from a mixture of Gaussians distribution i.e., $\mathbf{x}_i \sim \sum_{p=1}^{P} \pi_p \mathcal{N}(\mu_p, \Sigma_p)$. *Let* $\mathbf{x}_0 \in \mathbb{R}^N$ *denote a query data point in the ambient space, and let the other data points* $\mathbf{x}_i$ *be ordered such that* $d(\mathbf{x}_0, \mathbf{x}_1) < d(\mathbf{x}_0, \mathbf{x}_2) < \ldots < d(\mathbf{x}_0, \mathbf{x}_Q)$. *Let* δ *denote the* ℓ_∞*-norm distortion parameter in RHash, and let* c *denote a positive constant. Then, if* $\mathbb{E}_x[\Delta_k] \geq 2\delta + \sqrt{\frac{1}{c}\log\frac{Qk}{\epsilon}}$, *RHash preserves all* k*NNs of the query point* $\mathbf{x}_0$ *with probability at least* $1 - \epsilon$.

Before outlying the proof, we need to prove two lemmas.

Proof of Lem. 2

Lemma 2 Let x be a Gaussian random variable as $x \sim \mathcal{N}(\mu, \sigma^2)$. Define the distortion of the sigmoid approximation at x as $|h(x) - \sigma_\alpha(x)|$. Then, the expected distortion is bounded as

$$\mathbb{E}_x[|h(x) - \sigma_\alpha(x)|] \leq \frac{1}{\sigma\sqrt{2\pi\alpha}} + 2e^{-(\sqrt{\alpha}+c/\alpha\sigma^2)},$$

where c is a positive constant. As α goes to infinity, the expected distortion goes to 0.

Proof. It is easy to see that the distortion $|h(x) - \sigma_\alpha(x)|$ occurs at $x = 0$. Therefore, among different values of μ, $\mu = 0$ gives the largest distortion since the density of x peaks at $x = 0$. Therefore, we bound the distortion at setting $\mu = 0$, which is an upper bound of the distortion when $\mu \neq 0$. By definition (1) in the main text, $h(x)$ can be written as

$$h(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

When $x \sim \mathcal{N}(0, \sigma^2)$, we have

$$\begin{aligned}
&\mathbb{E}_x[|h(x) - \sigma_\alpha(x)|] = \int_{-\infty}^{\infty} |h(x) - \sigma_\alpha(x)|\, \mathcal{N}(x; 0, \sigma^2)\mathrm{d}x \\
&= 2\int_0^{\infty} (h(x) - \sigma_\alpha(x))\mathcal{N}(x; 0, \sigma^2)\mathrm{d}x \qquad (5.6) \\
&= 2\int_0^{x_0} (h(x) - \sigma_\alpha(x))\mathcal{N}(x; 0, \sigma^2)\mathrm{d}x \\
&\quad + 2\int_{x_0}^{\infty} (h(x) - \sigma_\alpha(x))\mathcal{N}(x; 0, \sigma^2)\mathrm{d}x \\
&\leq 2\int_0^{x_0} \frac{1}{2}\mathcal{N}(x; 0, \sigma^2)\mathrm{d}x + 2\int_{x_0}^{\infty} \frac{1}{1 + e^{\alpha x_0}}\mathcal{N}(x; 0, \sigma^2)\mathrm{d}x \qquad (5.7) \\
&\leq \frac{x_0}{\sqrt{2\pi}\sigma} + 2\frac{e^{-cx_0^2/\sigma^2}}{1 + e^{\alpha x_0}} \qquad (5.8) \\
&\leq \frac{1}{\sigma\sqrt{2\pi\alpha}} + 2e^{-(\sqrt{\alpha}+c/\alpha\sigma^2)}, \qquad (5.9)
\end{aligned}$$

when we set $x_0 = \frac{1}{\sqrt{\alpha}}$ and c is a positive constant. In (5.6), we used the fact that $\sigma_\alpha(x)$ and $h(x)$ are symmetric with respect to the point $(0, \frac{1}{2})$. (5.7) is given by the properties of the sigmoid function, (5.8) is given by the Gaussian concentration inequality [102], and (5.9) is given by the inequality $1/(1 + e^{\alpha x_0}) \leq e^{-\alpha x_0}$. The fact that $\mathbb{E}_x[|h(x) - \sigma_\alpha(x)|] \to 0$ as $\alpha \to \infty$ is obvious from the bound above.

Proof of Thm. 3

Theorem 3 Assume that all the data points are independently generated from a mixture of Gaussian distribution, i.e., $\mathbf{x}_i \sim \sum_{p=1}^{P} \pi_p \mathcal{N}(\mu_p, \Sigma_p)$. *Let* $\mathbf{x}_0 \in \mathbb{R}^N$ *denote a query data point in the ambient space, and the other data points* $\mathbf{x}_i$ *be ordered so that* $d(\mathbf{x}_0, \mathbf{x}_1) < d(\mathbf{x}_0, \mathbf{x}_2) < \ldots < d(\mathbf{x}_0, \mathbf{x}_Q)$. *Let* δ *denote the final value of the* ℓ_∞*-norm distortion parameter defined in RHash, and let* c *denote a positive constant. Then, if* $\mathbb{E}_x[\Delta_k] \geq 2\delta + \sqrt{\frac{1}{c} \log \frac{Qk}{\epsilon}}$, *then RHash preserves the* k*-nearest neighbors of* $\mathbf{x}_0$ *with probability at least* $1 - \epsilon$.

In order to prove this theorem, we need the following Lemma:

Lemma 4 Let $\mathbf{x}_0, \ldots, \mathbf{x}_N$ *and* Δ_k *be defined as in Thm. 3. Then, there exist a constant* c *such that* $P(\Delta_k - \mathbb{E}_x[\Delta_k] < t) \leq e^{-ct^2}$ *for* $t > 0$.

Proof. Since the data points $\mathbf{x}_0$, $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are independently generated from a finite mixture of Gaussian distributions, the random variable of their concatenation

$\mathbf{y} = [\mathbf{x}_0^T, \mathbf{x}_k^T, \mathbf{x}_{k+1}^T]^T \in \mathbb{R}^{3N}$ is sub-Gaussian [103]. Then, we have

$$\begin{aligned}
\Delta_k(\mathbf{y}) &= \|\mathbf{x}_0 - \mathbf{x}_{k+1}\|_2 - \|\mathbf{x}_0 - \mathbf{x}_k\|_2 \\
&= \|\begin{pmatrix} \mathbf{I} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\mathbf{I} \end{pmatrix}\mathbf{y}\|_2 - \|\begin{pmatrix} \mathbf{I} & 0 & 0 \\ 0 & -\mathbf{I} & 0 \\ 0 & 0 & 0 \end{pmatrix}\mathbf{y}\|_2 \\
&\leq \|\begin{pmatrix} 2\mathbf{I} & 0 & 0 \\ 0 & -\mathbf{I} & 0 \\ 0 & 0 & -\mathbf{I} \end{pmatrix}\mathbf{y}\|_2 \\
&\leq 2\|\mathbf{y}\|_2,
\end{aligned}$$

where we have used the triangular inequality in the second to last step, the Rayleigh-Ritz theorem [104], and the fact that the maximum singular value of the matrix in the step before is 2. This result means that $\Delta_k(\mathbf{y})$ is a Lipschitz function of $\mathbf{y}$. Thus, by Talagrand's inequality [102], we have that $P(\Delta_k - \mathbb{E}_x[\Delta_k] < -t) \leq e^{-ct^2}$ for some positive constant c and $t > 0$, since $\mathbf{y}$ is sub-Gaussian.

Now we are ready to prove Thm. 3.

Proof. Let E denote the event that the set of top-k nearest neighbors is not preserved in the Hamming space. Then, we have $E = \cup e_{m,n}$, where $e_{m,n}$ denote the event that $d_H(\mathbf{x}_0, \mathbf{x}_m) > d_H(\mathbf{x}_0, \mathbf{x}_n)$ with $m \in \{1, \ldots, k\}$ and $n \in \{k+1, \ldots, Q\}$. Then, using the union bound [105], we have

$$\begin{aligned}
P(E) &\leq \sum_{m,n} P(e_{m,n}) \leq k(Q-k)P(e_{k,k+1}) \\
&= k(Q-k)P(d_H(\mathbf{x}_0, \mathbf{x}_k) > d_H(\mathbf{x}_0, \mathbf{x}_{k+1})) \\
&= k(Q-k)P(d_H(\mathbf{x}_0, \mathbf{x}_{k+1}) < d_H(\mathbf{x}_0, \mathbf{x}_k)),
\end{aligned}$$

where we have used the fact that the most possible event among all $e_{m,n}$ events is the

one corresponding to the order mismatch between the k^{th} and $k+1^{\text{th}}$ nearest neighbor. Now, note that the RHash output δ satisfies $\max_{i,j} |d_H(\mathbf{x}_i, \mathbf{x}_j) - d(\mathbf{x}_i, \mathbf{x}_j)| \leq \delta$[†]. Observe that $\Delta_k = d(\mathbf{x}_0, \mathbf{x}_{k+1}) - d(\mathbf{x}_0, \mathbf{x}_k) \geq 2\delta$ is a sufficient condition for $d_H(\mathbf{x}_0, \mathbf{x}_{k+1}) \geq d_H(\mathbf{x}_0, \mathbf{x}_k)$, since

$$\begin{aligned} & d_H(\mathbf{x}_0, \mathbf{x}_{k+1}) - d_H(\mathbf{x}_0, \mathbf{x}_k) \\ & \quad \geq d(\mathbf{x}_0, \mathbf{x}_{k+1}) - \delta - d(\mathbf{x}_0, \mathbf{x}_k) - \delta \\ & \quad \geq 2\delta - 2\delta = 0, \end{aligned}$$

by the triangular inequality. This leads to

$$\begin{aligned} & P(d_H(\mathbf{x}_0, \mathbf{x}_{k+1}) < d_H(\mathbf{x}_0, \mathbf{x}_k)) \\ & \quad = 1 - P(d_H(\mathbf{x}_0, \mathbf{x}_{k+1}) \geq d_H(\mathbf{x}_0, \mathbf{x}_k)) \\ & \quad \leq 1 - P(\Delta_k \geq 2\delta) = P(\Delta_k < 2\delta). \end{aligned}$$

Therefore, combining all the above and Lem. 4, the probability that the k-nearest neighbor is not preserved is bounded by

$$\begin{aligned} P(E) & \leq k(Q-k)P(d_H(\mathbf{x}_0, \mathbf{x}_{k+1}) < d_H(\mathbf{x}_0, \mathbf{x}_k)) \\ & \leq k(Q-k)P(\Delta_k < 2\delta) \\ & = k(Q-k)P(\Delta_k - \mathbb{E}_x[\Delta_k] < -(\mathbb{E}_x[\Delta_k] - 2\delta)) \\ & \leq k(Q-k)e^{-c(\mathbb{E}_x[\Delta_k]-2\delta)^2} \\ & \leq kQe^{-c(\mathbb{E}_x[\Delta_k]-2\delta)^2}. \end{aligned}$$

Now, let $kQe^{-c(\mathbb{E}_x[\Delta_k]-2\delta)^2} \leq \epsilon$, we have that the requirement for the k-nearest neighbors to be exactly preserved with probability at least $1-\epsilon$ is

$$\mathbb{E}_x[\Delta_k] \geq 2\delta + \sqrt{\frac{1}{c}\log\frac{Qk}{\epsilon}}.$$

[†]Here we assume $\lambda = 1$ without loss of generality.

Remark Note that our bound on the number of nearest neighbor preserved k depends on the final outcome of the RHash algorithm in the value of δ. In order to relate our result to the number of binary hash functions M required, we can make use of [96]. For a bounded set $K \subset \mathbb{R}^N$ with diameter 1, let $M \geq C\delta^{-6}w(K)^2$ where $w(K) = \mathbb{E}_x[\sup_{\mathbf{x}\in K}\langle g, \mathbf{x}\rangle]$ denotes the Gaussian width of K and some constant C. Then, Thm. 1.10 in (Plan, Yaniv, and Roman Vershynin. Dimension reduction by random hyperplane tessellations.) states that, with high probability, $h(x)$ as defined in (1) with a random matrix $\mathbf{W}$ whose entries are independently generated from $\mathcal{N}(0, 1)$ is a δ_0-isometric embedding. Therefore, if we initialize RHash with such a random $\mathbf{W}$ which is likely to be δ_0-isometric, then empirically (see Figure 5.3), the RHash algorithm will learn a better embedding that is δ-isometric with $\delta < \delta_0$. Therefore, we have that the number of hash functions M required for k-nearest neighbor preservation is at least $M \sim (\mathbb{E}_x[\Delta_k] - \sqrt{\log(kQ/\epsilon)})^{-6}w(\mathbf{X})^2$ assuming that the training dataset $\mathbf{X}$ is properly normalized.

5.3 Experiments

In this section, we validate the RHash and RHash-CG algorithms via experiments using a range of synthetic and real-world datasets, including three small-scale, three medium-scale, and one large-scale datasets with respect to three metrics. We compare RHash against ten state-of-the-art binary hashing algorithms, including binary reconstructive embedding (BRE) [82], spectral hashing (SH) [81], anchor graph hashing (AGH) [86], multidimensional spectral hashing (MDSH) [80], scalable graph hashing (SGH) [83], PCA hashing (PCAH) [84], isotropic hashing (IsoHash) [85], spherical Hamming distance hashing (SHD) [94], circulant binary embedding (CBE) [95], and locality-sensitive hashing (LSH) [106].

5.3.1 Performance metrics and datasets

We compare the algorithms using the following three metrics:

Maximum distortion $\delta = \inf_{\lambda>0} ||\lambda\hat{\mathbf{v}} - \mathbf{c}||_\infty$, where the vector $\hat{\mathbf{v}}$ contains the pairwise Hamming distances between the learned binary codes. This metric quantifies the distance preservation after projecting the training data in the ambient space into binary codes. We also define the maximum distortion for unseen test data δ_{test}, which measures the distance preservation on a hold-out test dataset using the hash function learned from the training dataset.

Mean average precision (MAP) for NN preservation in the Hamming space. MAP is computed by first finding the set of kNNs for each query point on a hold-out test data in the ambient space $\mathcal{L}^k$ and the corresponding set $\mathcal{L}^k_{\text{H}}$ in the Hamming space and then calculating the average precision AP $= |\mathcal{L}^k \cap \mathcal{L}^k_{\text{H}}|/k$. We then report MAP by calculating the mean value of AP across all data points.

Kendall τ ranking correlation coefficient. We first rank the set of kNNs for each data point by increasing distance in the ambient space as $\mathcal{T}(\mathcal{L}^k)$ and in the Hamming space as $\mathcal{T}(\mathcal{L}^k_{\text{H}})$. The Kendall τ correlation coefficient is a scalar $\tau \in [-1, 1]$ that measures the similarity between the two ranked sets $\mathcal{T}(\mathcal{L}^k)$ and $\mathcal{T}(\mathcal{L}^k_{\text{H}})$ [107]. The value of τ increases as the similarity between the two rankings increases and reaches the maximum value of $\tau = 1$ when they are identical. We report the average value of τ across all data points in the training dataset.

To compare the algorithms, we use the following standard datasets from computer vision: *Random* consists of independently drawn random vectors in $\mathbb{R}^{100}$ from a multivariate Gaussian distribution with zero mean and identity covariance matrix. *Translating squares* is a synthetic dataset consisting of 10×10 images that are translations of a 3×3 white square on black background [100]. *MNIST* is a collection of 60,000 28×28 greyscale im-

ages of handwritten digits [91]. *Photo-Tourism* is a corpus of approximately 300,000 image patches, represented using scale-invariant feature transform (SIFT) features [108] in $\mathbb{R}^{128}$ [109]. *LabelMe* is a collection of over 20,000 images represented using GIST descriptors in $\mathbb{R}^{512}$ [110]. *Peekaboom* is a collection of 60,000 images represented using GIST descriptors in $\mathbb{R}^{512}$ [110]. Following the experimental approaches of the hashing literature [82, 111], we pre-process the data by subtracting the mean and then normalizing all points to lie on the unit sphere.

5.3.2 Small- and medium-scale experiments

We start by evaluating the performance of RHash and RHash-CG using a subset of the first three datasets. Small-scale datasets enable us to compare the performance of RHash and RHash-CG against other hashing algorithms and assess the asymptotic behavior of algorithms in preserving distances in a regime where number of bits is high enough (compared to total number of NNs) to preserve a big portion of NN distances.

Experimental setup. We randomly select $Q = 100$ data points from the *Random*, *Translating squares*, and *MNIST* datasets. We then apply the RHash, RHash-CG, and all the baseline algorithms on each dataset for different target binary code word lengths M from 1 to 70 bits. We set the RHash and RHash-CG algorithm parameters to the common choice of $\rho = 1$ and $\eta = 1.6$. To generate hash function of length M for LSH, we draw M random vectors from a Gaussian distribution with zero mean and an identity covariance matrix. We use the same random vectors to initialize RHash and other baseline algorithms. In the NN preservation experiments, to show the direct advantage of minimizing ℓ_∞-norm over ℓ_2-norm, we followed the exact procedure described in BRE [82] to select the training secants, i.e., we apply the RHash algorithm on the lowest 5% of the pairwise distances

	MAP / Top-500				Training time
Training size Q	$1K$	$10K$	$100K$	$240K$	$240K$
Secant size $\lvert\mathcal{S}(\mathcal{X})\rvert$	$100K$	$10M$	$1B$	$5B$	$5B$
RHash-CG	**52.79** ($\pm$0.15)	**54.69** ($\pm$0.18)	**54.93** ($\pm$0.23)	**55.52** ($\pm$0.11)	541.43
CBE	38.70 ($\pm$1.18)	38.12 ($\pm$1.34)	38.50 ($\pm$2.05)	38.53 ($\pm$0.83)	68.94
SPH	44.33 ($\pm$0.74)	44.24 ($\pm$0.61)	44.37 ($\pm$0.71)	44.32 ($\pm$0.63)	184.46
SH	40.12 ($\pm$0.00)	39.37 ($\pm$0.00)	38.79 ($\pm$0.00)	38.26 ($\pm$0.00)	3.05
MDSH	41.06 ($\pm$0.00)	41.23 ($\pm$0.00)	40.80 ($\pm$0.00)	40.39 ($\pm$0.00)	15.00
AGH	45.81 ($\pm$0.34)	47.78 ($\pm$ 0.38)	47.69 ($\pm$0.41)	47.38 ($\pm$0.32)	4.49
SGH	51.32 ($\pm$0.07)	51.33 ($\pm$0.20)	51.01 ($\pm$0.23)	50.66 ($\pm$0.76)	5.89
BRE	48.33 ($\pm$0.65)	50.67($\pm$0.33)	–	–	18685.51
PCAH	39.90 ($\pm$0.00)	38.53 ($\pm$0.00)	38.81 ($\pm$0.00)	37.50 ($\pm$0.00)	0.08
IsoHash	50.91 ($\pm$0.00)	50.90 ($\pm$0.00)	50.72 ($\pm$0.00)	50.55 ($\pm$0.00)	2.82
LSH	33.69 ($\pm$0.94)	33.69 ($\pm$0.94)	33.69 ($\pm$0.94)	33.69 ($\pm$0.94)	2.29×10^{-4}

Table 5.1 : Comparison of RHash-CG against several baseline binary hashing algorithms on large-scale *MNIST+rotation* dataset with over 5 billion secants $|\mathcal{S}(\mathcal{X})|$. We tabulate the Hamming ranking performance in terms of mean average precision MAP for different sizes of the dataset. All training times are in seconds.

(which are set to zero as in BRE) combined with the highest 2% of the pairwise distances.

We follow the *continuation* approach [112] to set the value of α. We start with a small value of α, (e.g., $\alpha = 1$) to avoid becoming stuck in bad local minima, and then gradually increase α as the algorithm proceeds. As the algorithm gets closer to convergence and has obtained a reasonably good estimate of the parameters $\mathbf{W}$ and λ, we set $\alpha = 10$, which enforces a good approximation of the sign function (see Lemma 1 in the Supplementary Materials).

Results. The plots in the top row of Figure 6.4 illustrate the value of the distortion parameter δ as a function of the number of projections (bits) M. The performance of RHash and RHash-CG closely follow each other, indicating that RHash-CG is a good approximation to RHash. Both RHash and RHash-CG outperform the other baseline algorithms in terms of the distortion parameter δ. Among these baselines, LSH has the lowest distance preservation performance since random projections are oblivious to the intrinsic geometry of the training dataset. To achieve $\delta = 1$, RHash(-CG) requires 60% fewer bits M than CBE and BRE. RHash(-CG) also achieves better distance preservation performance asymptotically, i.e., up to $\delta \approx 0.5$, given a sufficient number of bits ($M \geq 70$), while for most of the other algorithms the performance plateaus after $\delta = 1$. RHash's superior distance preservation performance extends well to unseen data thanks to the simple structure of its hash function and the ℓ_∞-norm minimization. To demonstrate this property, we test the hash function learned from the small *MNIST* dataset on an unseen test dataset of the same size. Figure 5.5 (left) shows the superior performance of RHash and RHash-CG in minimizing the distortion parameter δ_{test} on the test dataset over eight other hashing baselines, suggesting that they generalize very well to unseen data. Figure 5.5 (right) shows the tradeoff between the adaptivity of different binary hashing schemes to a given training set (achieving a lower

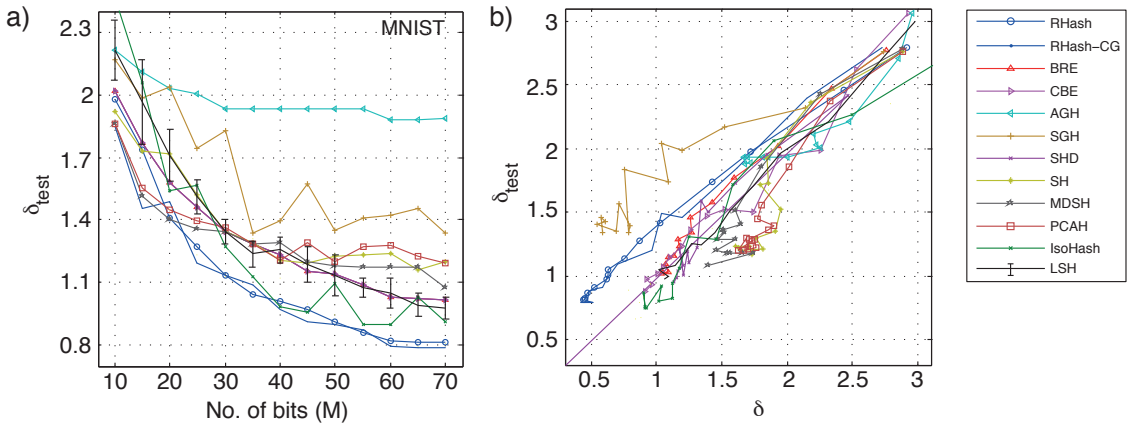

Figure 5.5 : Comparison of RHash and RHash-CG against several state-of-the-art binary hashing algorithms in preserving distances on MNIST data. (a) RHash-CG outperforms the other algorithms in minimizes the distance distortion parameter on unseen data δ_{test}. (b) RHash and RHash-CG provide better distance preservation guarantee with a small sacrifice to universality.

value of δ_{test}) vs. universality (deviation of the δ_{test}-δ curve from $\delta_{\text{test}} = \delta$). On the one hand, random projection (LSH) is completely oblivious to the data at hand and therefore attains similar values of the distortion parameter on both test and training sets ($\delta_{\text{test}} \approx \delta$). On the other hand, RHash adapts very well to the training dataset and provides the best isometry guarantees on both the training data (δ) and test data (δ_{test}) while sacrificing universality.

The plots in the middle and bottom row of Figure 6.4 shows the average precision for retrieving training data and the Kendall τ correlation coefficient respectively, as a function of the number of bits M. We see that RHash preserves a higher percentage of NNs compared to other baseline algorithms as M increases with better average ranking among $k = 10$-NNs.

Now we showcase the performance of RHash-CG on three medium-scale, real-world datasets used in [82, 111], including *Photo-tourism*, *LabelMe*, and *Peekaboom* for the task of *data retrieval*. From each dataset we randomly select $Q = 1000$ training points, following the setup in BRE [82], and use them to train RHash-CG and the other baseline algo-

rithms. We then randomly select a separate set of $Q = 1000$ data points and use it to test the performance of RHash-CG and other baseline algorithms in terms MAP with $k = 50$. Figure 5.4 illustrates the performance of RHash-CG on these datasets. RHash-CG outperforms all the baseline algorithms with large margins in Hamming ranking performance in term of MAP with top-50 NNs.

5.3.3 Large-scale experiments

We demonstrate how RHash-CG scales to large-scale datasets. We use the full *MNIST* dataset with 60,000 training images and augment it with three rotated versions of each image (rotations of 90°, 180°, and 270°) to create a larger dataset with $Q = 240{,}000$ data points. Next, we construct 4 training sets with 1,000, 10,000, 100,000, and 240,000 images out of this large set. We train all algorithms with $M = 30$ bits and compare their performance on a test set of 10,000 images. The results for all algorithms are given in Table 5.1; we tabulate their performance in terms of MAP for the top-500 NNs. [‡] The performance of RHash-CG is significantly better than the baseline algorithms and, moreover, improves as the size of the training set grows.

5.3.4 Discussion

We have demonstrated that our worst-case, ℓ_∞-norm-based, Robust Hashing (RHash) algorithm is superior to a wide range of algorithms based on the more traditional average-case, ℓ_2-norm. Despite its non-convexity and non-smoothness, RHash admits an efficient optimization algorithm that converges to a high-performing local minimum. Moreover, RHash-CG, the accelerated version of RHash, provides significant memory advantages

[‡]BRE fails to execute on a standard desktop PC with 12 GB of RAM due to the size of the secant set.

over existing algorithms; the memory requirement of RHash-CG is linear only in the number of *active* secants rather than the total number of secants, a significant advantage over the existing reconstructive based hashing methods. Our experiments with six datasets, three metrics, and ten algorithms have shown that RHash outperforms the state-of-the-art data-dependent hashing algorithms. The results in this paper provide a strong motivation for exploring ℓ_∞-norm formulations and other related algorithms for robust binary hashing.

Chapter 6

Biologically Relevant Representations of DNA Sequences Using The *K*mer2Vec Platform

6.1 Introduction

In the past few years we have been observing a revolution in the field of genomics. The DNA sequencing cost has dropped a considerable amount in only 15 years. As a consequence, huge volumes of sequencing data has been extracted from humans, plants, bacteria, viruses, and other organisms. This genomic data is a valuable resource to extract hidden biological knowledge and improve our perception of the real world.

The traditional approach to solve problems in biology involves coming up with a *biological model* of the universe. The biological model reflects the way we *see* the world through our observations of the biological phenomena. While *hand-carafted* biological models have been successful in solving many important problems in computational biology, such as sequence assembly or alignment, the availability of sequencing data is changing this paradigm by suggesting alternative ways to develop biological models for more complex problems; the biological data such as sequencing data can now be used to learn biological models and representations.

In this paper, we focus on learning the concept of drug-resistance in microbiology by parsing the DNA sequences of hundreds of microbial genomes and then transferring the knowledge learned from those organisms to multiple pairs of susceptible and drug-resistant strains of human-pathogens with no supervision or prior knowledge of biology.

A common method for representing and featurizing DNA sequence in classical computational biology literature is by segmenting the DNA sequence into short DNA sequences of length K – known as Kmers. Kmers provide a rich representation of the DNA sequence. Due to their flexibility and computational advantages, Kmers are a key part in many important algorithms in bioinformatics such as sequence alignment and assembly. Representation of DNA sequences using Kmers, however, loses the semantic information buried in genomic sequences; the information related to the ordering of Kmers will be lost in repressing the DNA sequence by its Kmers frequency. The ordering of Kmers, however, has important information about he biology of the sequence. Consider two Kmers with high Hamming distance that co-occur in the same gene and show up frequently in a microbial genome. These two Kmers are expected to have close distance in a biologically relevant representation while being far in the hamming space. In meta-genomic binning application, as an example, the occurrence of one Kmer can indirectly indicate the presence of the other Kmer.

To answer this shortcomings, we propose a new platform to embed DNA sequences into a lower dimensional space that respects the semantic of Kmers in genomic datasets. Inspired by the Word2Vec algorithm in natural language processing, we developed a new platform called Kmer2Vec to learn biologically relevant representation of DNA such as drug-resistance.

We have three distinct distribution in this work. One, our Kmer2Vec platform uses a continuous bag-of-words model for Kmers and learn the biological semantics using a two-layer neural network, which other implementations train on overlapping sequences and use skip-gram models. Two, our Kmer2Vec platform enables a universal and biologically relevant representation of DNA sequences of arbitrary length in a single low-dimensional space. Three, we propose the first machine-learning approach to learn drug-resistance from

raw DNA sequence with no prior biology knowledge. We show that embeddings of *K*mers learned on a separate set of microbial genomes, improves the drug-resistance representation a new set of bacterial strains.

6.1.1 Background

The idea of extracting *K*mers from DNA sequences, treating them as words in a text corpora, and analyzing them using natural language processing techniques has recently gained attention in machine learning and bioinformatics literature. BioVec [[113]] and Seq2Vec [[114]] use a two-layer neural network, similar to the one in Word2Vec algorithm, to find a new representation for *proteins* that better classify them into protein families. Dna2Vec [[115]] learns the representation of *K*mers with the ability to generalize to variable-length *K*. Dna2Vec further shows that vector addition operator in the subspace of *K*mer embeddings is equivalent to sequence concatenation in the space of *K*mers. In other words, if you concatenate two *K*mers, the resulting *K*mer will have a low-dimensional representation that is almost equal to the sum of the vector representation of original *K*mers. However, these methods do not focus on extracting biological concepts from larger DNA sequences such as genomic sequences with millions of basepairs, instead they limit their analysis to relationship between *K*mers which is not practically useful per se; they will only have practical relevance when the *K*mer representation connects to the biology of the original DNA sequence.

6.2 *K*mer2Vec

Our *K*mer2Vec platform is comprised of two layers: 1- *K*mer2vec embedding, which is an algorithm, inspired by the Word2Vec algorithm [[116]], that learns a low-dimensional embedding of *K*mers based on their semantics, and 2- *K*mer2Vec representation, which

embeds a DNA sequence of arbitrary length (e.g., the sequence of microbial genomes) into a low-dimensional space using the embedding of individual *K*mers. In this section we describe these two parts in detail.

6.2.1 The *K*mer2Vec embedding

In order to learn the *K*mer2vec embedding, we start off by collecting a set of DNA sequences that we aim to train our unsupervised algorithm. It is important to note that no information other than *raw DNA sequence* is required to train *K*mer2Vec. We segment the DNA sequences into shorter subsequences of length *K*, known as *K*mers. Next, we encode the *K*mers into one-hot encoding vectors, and feed the result into a two-layer neural network (Fig. 6.2.1**A** and **B**). Assume there are N distinct *K*mers in DNA sequence database and call them the vocabulary of our DNA sequences. We encode each *K*mer as a 1-sparse vector of length N prior to feeding to the neural network. The task of the neural network is to predict the occurrence of a *K*mer given its surrounding *K*mers using a single hidden layer of dimension $M \ll N$. Drawing an analogy to text processing, this task is similar to completing a sentence with a missing word. The network is trained to fill in, as an example, the following sentence "We See ... We Want." with no prior knowledge of English grammar with the word "What".

In order to find the input to the neural network, we loop over all the *K*mers extracted from the database and for each *K*mer we collect C one-hot encoded vectors as inputs of the network. Take the bold $K = 5$mer "**CTGGG**" in Fig. 6.2.1**A**, as an example. This *K*mer is surrounded by $C = 4$ other *K*mers. Half of these vectors correspond to *K*mers occurring before and half of the vectors correspond to *K*mers occurring after the *K*mer we are looping over. The output of the neural network is the one-hot encoded vector corresponding to the *K*mer we are looping over (the bold *K*mer in Fig. 6.2.1**A**).

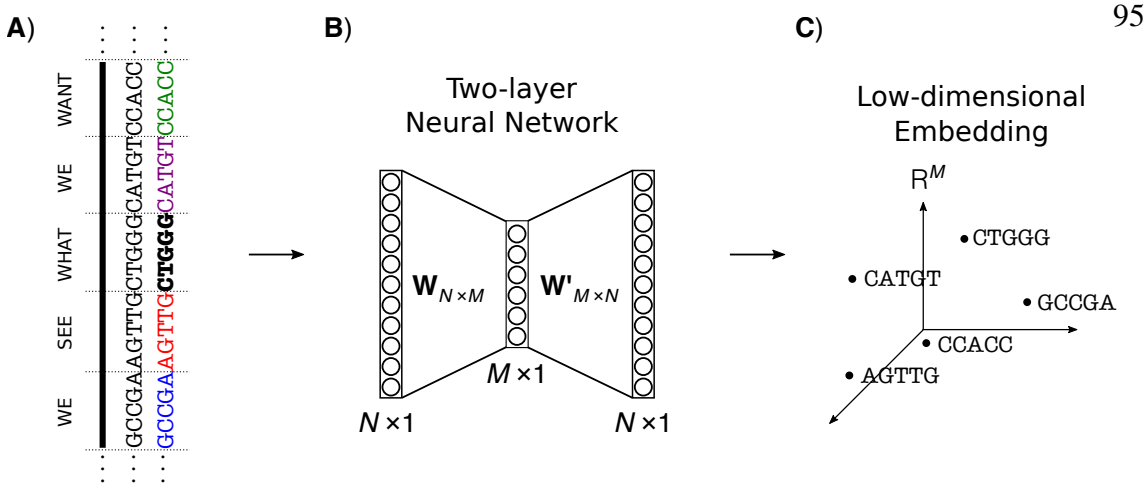

Figure 6.1 : **Illustration of the *K*mer2vec embedding pipeline.** **A**) We start by segmenting the sequence of a large dictionary of different organisms's genome into short non-overlapping subsequences of length K called Kmers. We produce a *bag-of-Kmers* representation of the N extracted Kmers from DNA sequences. **B**) Next, we train a two layer neural network which predicts the occurrence of a Kmer given its surrounding Kmers. The weight matrix $\mathbf{W}$ in the trained neural network act as a embedding operator into a lower ($M \ll N$) dimensional space. **C**) We use the weight matrix $\mathbf{W}$ to embed the Kmers into a low-dimensional space in $\mathbb{R}^M$. The new so-called Kmer2Vec embedding of Kmers preserve the semantics of the original DNA sequences.

Once the network is trained for the task of predicting the occurrence of Kmers given it's surrounding ones, we use the trained network as an embedding into a low (M)-dimensional space; we use the network to embed each Kmer from a subspace in $\mathbb{R}^N$ to a lower-dimensional subspace in $\mathbb{R}^M$, where $M \ll N$ (Fig. 6.2.1 **C**). Let matrix $\mathbf{G}$ denote a $N \times M$ matrix which comprises the Kmer2Vec embedding of all N Kmer in its rows. In contrast to the original representation of Kmers, where the distances between Kmers is defined by the Hamming distance between their DNA sequence, in our new representation distances between Kmers is defined the semantics of Kmers, i.e, how similar are the surrounding Kmers in the training sequence data.

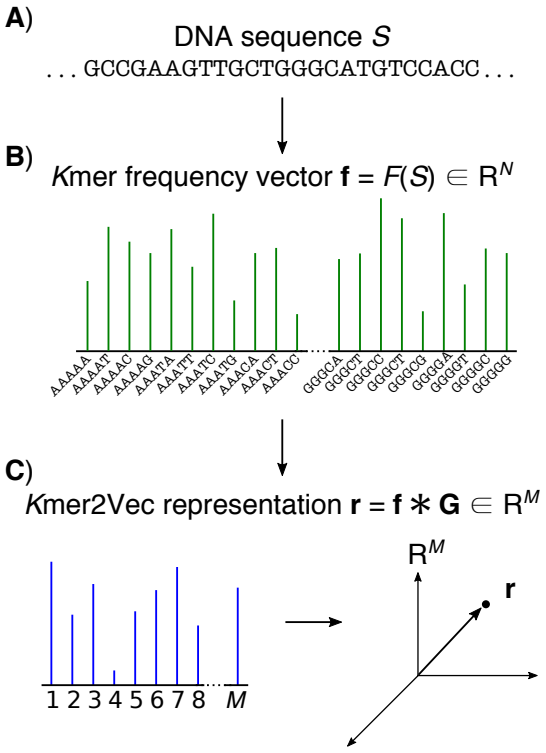

Figure 6.2 : **Illustration of the *K*mer2Vec representation of DNA sequences**. **A**, **B**) To find the *K*mer2Vec representation of a DNA sequence S, we first find the occurrence frequency of *K*mers ($K = 5$ in the illustration) and store them in a *K*mer frequency vector $\mathbf{f} \in \mathbb{R}^N$, where N counts the number of distinct *K*mers in S. **C**) We obtain the *K*mer2Vec representation of S by adding the *K*mer2Vec embedding of the *K*mers in $\mathbf{f}$ as given by $\mathbf{r} = \mathbf{f} * \mathbf{G} = \sum_{i=1}^{N} f_i \mathbf{G}_i \in \mathbb{R}^M$, where $\mathbf{G}_i$ is the *K*mer2Vec embedding of the i^{th} *K*mer.

6.2.2 *K*mer2Vec representation

We propose a new method to embed a DNA sequence of arbitrary length into a lower dimensional space using the *K*mer2Vec embedding. As we will see in the next chapters, our *K*mer2Vec representation respects the biological concepts of DNA sequences such as drug-resistance.

In order to find the *K*mer2Vec representation of a DNA sequence S (see Fig. 6.2.1), we first find the *K*mer occurrence frequency vector $f = \mathcal{F}(S) : \mathcal{A}^L \mapsto \mathbb{R}^N$, where $\mathcal{A} = \{\mathrm{A}, \mathrm{T}, \mathrm{C}, \mathrm{G}\}$ is the set of four alphabets in DNA sequences, L is the length of the DNA

sequence S, and N is number of distinct *K*mer is S. Given the frequency of occurrence of *K*mers in sequence S and the *K*mer2Vec embedding of *K*mers, we now describe how to find the representation of sequence S. We find the *K*mer2Vec representation of S by adding the *K*mer2Vec embedding vectors weighted by *K*mer frequencies in the sequence S,

$$\mathbf{r} = \mathcal{R}(S) = \sum_{i=1}^{N} f_i \mathbf{G}_i, \tag{6.1}$$

where $\mathbf{G}_i$ denotes the i^{th} row of $\mathbf{G}$, the Kmer2Vec representation of i^{th} *K*mer. Using this representation a DNA sequence S of arbitrary length is mapped into a point $\mathbf{r}$ in the low-dimensional space in $\mathbb{R}^M$. This enable us extracting biological concepts from reference sequences or a collection of sequencing reads from a more complex sample.

6.3 Results

We showcase the performance of our algorithm in learning new biological concepts using a real-world problem. We asses our *K*mer2Vec platform in learning the biological concept of drug-resistance from DNA sequences of microbial genomes without any prior microbiology information.

6.3.1 Learning drug-resistance concept

Drug-resistance is a growing crisis in the modern world; more and more number of bacteria are becoming resistance to current antibiotics. It is estimated that at least 2 million new illnesses and 23 thousand deaths is caused by drug-resistant pathogens every year. [*] Among the drug-resistant bacteria, *Enterococcus faecium*, *Staphylococcus aureus*, *Klebsiella pneumoniae*, *Acinetobacter baumannii*, *Pseudo monas aeruginosa*, and *Enterobacter* strains, a group of pathogens known as ESKAPE, are notified by the Infectious Diseases

[*] https://www.cdc.gov/drugresistance/about.html

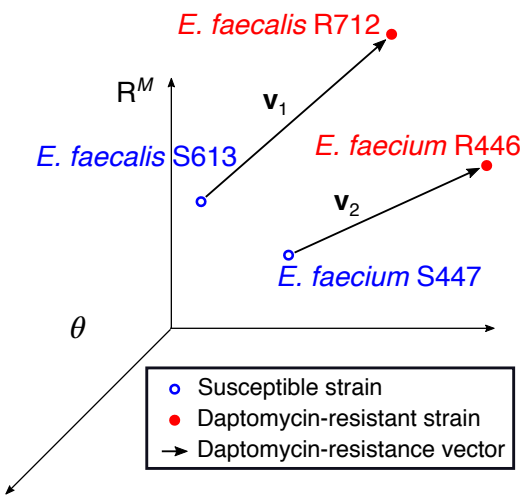

Figure 6.3 : **Learning drug-resistance vector**. *K*mer2Vec representation of susceptible bacterial strains (S613 and S447), their drug-resistant strains (R712 and R446), and drug-resistance vectors ($\mathbf{v}_1$ and $\mathbf{v}_2$) correspondingly in low dimensional space $\mathbb{R}^M$ ($M = 200$ in our simulations). Biological relevance of the low dimensional representation is measured by the angle between drug-resistance vectors $\theta = \measuredangle(\mathbf{v}_1, \mathbf{v}_2)$ and the norm of difference vector $\rho = \|\mathbf{v}_1 - \mathbf{v}_2\|_2/(\|\mathbf{v}_1\|_2 + \|\mathbf{v}_2\|_2)$. Smaller values of θ and ρ corresponds to more biologically relevant representations in unsupervised learning of drug-resistance from raw genome sequence.

Society of America as problem pathogens requiring new therapies [[117]]. Unfortunately there are only few types of antibiotics that can be used to cure infections caused by ESKAPE pathogens. Insights into the mechanism of resistance in these pathogens can help produce new antibiotics.

Here we showcase the performance of *K*mer2Vec platform in learning the representation of Daptomycin-resistance in *Enterococcus faecalis* and *Enterococcus faecium* strains. The representation can help us discover resistance mechanism in new species.

To achieve this goal, we obtain two pair of strains (one susceptible and one drug-resistant) of *Enterococcus faecalis* and *Enterococcus faecium* species from NCBI website, which have acquired resistance to Daptomycin, through the same resistance mechanisms. Since both strains have acquired resistance through the same mechanism, we expect that in a biologically-relevant representation of microbial organisms, the same vector take us

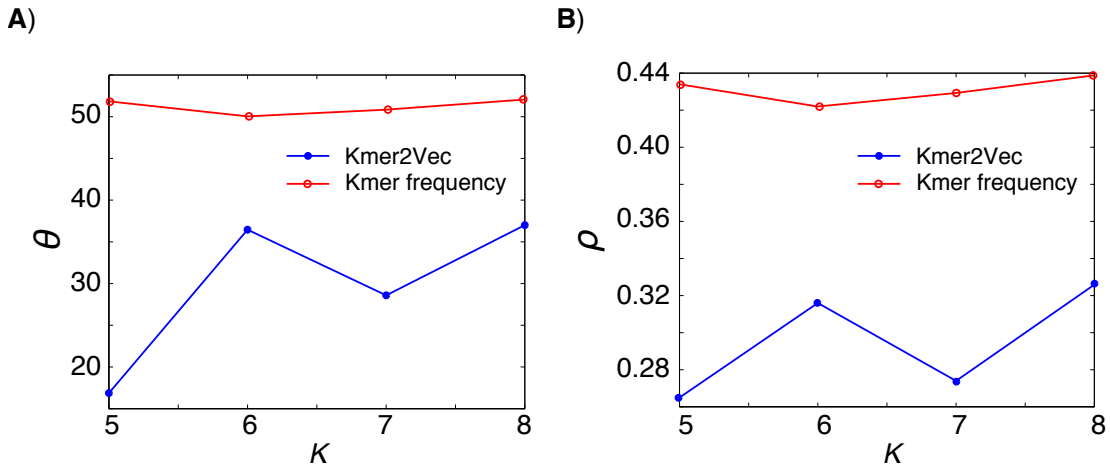

Figure 6.4 : **Comparison of Kmer2Vec and Kmer frequency representations in learning Daptomycin-resistance cencept.** *K*mer2Vec learns a representation with smaller angle θ and magnitude ρ (see Fig. 6.3 for exact definition) between the Daptomycin-resistance vectors of *E. faecalis* and *E. faecium* strains. *K*mer2Vec enables a better biologically relevant representation of microbial genomic sequences in terms of drug-resistance by leveraging the semantics of *K*mer in genomic data in an supervised fashion.

from a susceptible strain to the drug-resistant strain in the low-dimensional representation. The vectors associated to Daptomycin-resistant in these species are illustrated in Fig. 6.3 and labeled by $\mathbf{v}_1$ and $\mathbf{v}_2$. We define the measures to evaluate the biological relevance of embeddings based on how these two vectors align. Our first metric measures the angle between the drug-resistant vectors $\theta = \measuredangle(\mathbf{v}_1, \mathbf{v}_2)$ and our second metric finds the norm of the drug-resistant distance vector $\rho = \|\mathbf{v}_1 - \mathbf{v}_2\|_2/(\|\mathbf{v}_1\|_2 + \|\mathbf{v}_2\|_2)$ after normalization. Smaller values of θ and ρ corresponds to cases where the representation better aligns the drug-resistant vectors and thus learns better biologically relevant representations of drug-resistance.

Fig. 6.3.1 illustrates that Kmer2Vec learns representations that associates *E. faecalis* and *E. faecium* strains with vectors that better captures the drug-resistance concept com-

pared to using the original *K*mer representation of strains while embedding into a much lower dimensional space ($M \ll N$). In particular, *K*mer2Vec reduces the drug-resistance angle θ by a factor of 2.5 and distance ρ by a factor of 1.5 while embedding into vectors with dimension 5 times smaller than dimension of *K*mer embedding. Our results suggest that *K*mer2Vec is a more efficient and biologically more relevant alternative for *K*mer representaion.

6.4 Conclusion

We developed a new platform, dubbed as *K*mer2Vec, to embed DNA sequences of arbitrary length into a single lower-dimensional space. We demonstrate that our *K*mer2Vec representation captures important biological concepts, such as drug-resistance in microbiology, by parsing raw DNA sequence with no prior knowledge. There are many unexplored directions: Finding biological applications other than drug-resistance in genomics where a lower dimensional biologically-relevant representation is of interest, finding other neural network architectures beyond the one used in traditional Word2Vec, and incorporating side information besides raw DNA sequence are among interesting directions to explore.

6.5 Materials and Methods

6.5.1 *K*mer2Vec implementaion

To train the network we create *non-overlapping* *K*mers off the genome sequences into one large corpus. We used 43 genome sequences and none of them were from the sequences that we used to perform the drug-resistance analysis. We used the publicly available Word2Vec code available at https://code.google.com/archive/p/word2vec/. We set the dimensionality of embedded space to $N = 200$. For every *K*mer, we used a context window of the 5 *K*mers

before it and the 5 *K*mers after it ($C = 10$) to understand the association or semantics of what other *K*mers are next to it. For negative sampling we set it as 5 while we set sampling as 10^{-4} in order to to counter the imbalance between rare and frequent *K*mers. This model was trained as a continuous bag-of-words while skip-gram model could also be used to embed the *K*mers.

6.5.2 Drug-resistant strains

We obtained four strains, *Enterococcus faecalis* S613 , *Enterococcus faecalis* R712, *Enterococcus faecium* S447, and *Enterococcus faecium* R446, from the NCBI Genome Database with accession numbers NZ-GG739820.1, NZ-GG739925.1, NZ-JH806851.1, and NZ-JH808003.1.

Chapter 7

Conclusion and Future Directions

In this thesis, I developed several machine learning tools and algorithms for and inspired by recent advancements in the field of genomics. I categorize these developments into two categories: 1- Computational sensing, and 2- Large-scale processing and inference.

7.1 Computational sensing

From the sensing side, we developed a universal microbial diagnostics (UMD) platform to rapidly detect bacteria in resource-limited environments. Enabled by the compressive sensing theory, UMD identifies the composition of a bacterial sample by decoding the hybridization patterns of human-pathogens across a set of probes whose sequence is designed by random ordering of nucleotides. The UMD can be implemented as a low-cost, rapid, and reprogrammable diagnostic chip to universally identify bacteria that are problematic now and in future. This project opens up many interesting direction.

A. Binary Sensing: Random binary sensing matrices reduce the computational cost in sparse recovery problems, as the zero entries in the binary sensing matrix do not add to the computational cost of the recovery algorithm. Besides it's computational advantages, random binary sensing matrices help improve the detection performance in our universal microbial diagnostics platform, both in terms of sensitivity and specificity. We thus propose a new method, as a future work, that uses random binary measurements in combination with the UMD platform. Here we describe how the detection accuracy of UMD can be improved

using binary sensing matrices. The detection performance of the UMD system depends on the accuracy of the thermodynamic modeling of stable hybridizations of random probes. To model the hybridization affinity of random probes, UMD adds the hybridization levels of numerous partially stable probe-bacteria bindings to get an aggregate hybridization affinity. The measured hybridization affinities enable UMD to differentiate between two bacterial genomes. When the noise level in the system is high differentiating between the hybridization affinities become difficult since it becomes harder to predict the hybridization affinity of partially stable bindings. To improve the detection accuracy of the system, we propose a new platform where random probes either completely bind or do not bind to the microbial genomes. In our new approach, random probes produce binary-valued (0/1) sensing matrices when hybridized to the reference bacteria. A zero entry in the sensing matrix means the probe does not bind to the genome, while a one entry in the sensing matrix means the probe binds at least once to the bacterial genome. This approach will potentially improve the prediction accuracy of hybridization affinities, since the simulation now only needs to model thermodynamically stable bindings of probes to bacteria. Further, given that the system is now differentiating between bound and unbound conditions, any noise resulting from variabilities in hybridization are mitigated. Coupled with a simple fluoresce coloring scheme of DNA probes, pathogens can be easily identified by the color codes induces on the probes. In addition to its improved prediction accuracy, our preliminary simulations indicate that the resulting binary matrices improve the coherence properties of the sensing matrices for a large subset of pathogens. Improved coherence reduces the number of DNA probes required to identify human pathogens.

B. Data-aware Random Probes: In the UMD platform, the sequence of DNA probes are drawn uniformly random from the set of four alphabets $\{$A,T,C,G$\}$. The randomness allows the probes to universally identify new pathogenic strains – even the pathogens that will be

sequenced in future. While randomness is a key feature of UMD system, the distribution that random probes are drawn can be improved in a way that favors probes that better differentiate among bacteria. For example, if the sequences of target pathogens have high GC content, ratio of G+C nucleotides to all the total number of nucleotides, then we expect to draw the sequence of random probes from a distribution that favors high GC content probes in contrast to a uniform random DNA probes. The new set of data-aware random DNA probes will drastically improve the sensitivity and specificity of the UMD system.

As a future direction, we propose a new method that seeks to find a distribution to draw the random probes from which is tailored to the training microbial genome. To this end, an algorithm will be designed that parse the dataset of bacterial genome sequences in the database, and finds the statistics of the DNA probes that best differentiate the bacterial organisms. The statistics of the dataset can be learned by finding the differentiating subsequences of length K – known as Kmers. It is easy to see that the first order statistics of the sequences is the same as GC content and can be easily extracted by counting the number of alphabets A, T, C, and G which is the histogram of K=1-mers. Finding the differencing Kmers for higher values of K however is computationally intensive since it requires the alignment of multiple genomes together. To tackle this problem, we propose to first segment the DNA sequences into Kmers of any K and then construct a sparse regression problem with Kmers as features and bacteria as atoms of the dictionary. With this setup, we propose solving a sparse regression problem that seeks to find few (sparse) number of Kmers (features) that contributes the most in classifying the DNA sequences into different species. For larger values of K this sparse regression problem becomes intractably since the dimensionality of features (4^K) explodes. We aim to use techniques from large-scale machine learning to mitigate this problem.

In Chapter 4 of this thesis, we developed a sensor selection algorithm for sparse signals

dubbed as Insense. Insense is an optimization algorithm that finds the set of rows in a large sensing matrix with smallest coherence across its columns. We showed that Insense selects sensors An interesting direction to explore is to come up with scalable version of Insense. In many applications the sensing matrix is too big that non of the greedy or optimization algorithms can operate both in terms of memory requirement and speed. Theoretical guarantees for sensor selection in the sparse regime is another important direction to be explored.

7.2 Large-scale processing and inference

In Chapter 5, I described Rhash, an efficient method to hash data point from a high-dimensional space into compact binary codes that preserve the distances between neighboring points up to a distortion parameter. We showed that our robust hashing scheme outperforms various hashing baselines in preserving top nearest neighbors. The potential interplay between our hashing algorithm and neural network in among interesting potential direction to be explored in future. Each hash function has a linear operator followed by a sign operator, very similar to a single layer of neural network. A potential question is how the norm-preserving properties of embedding will be morphed while getting cascaded.

In the last chapter of this thesis, we developed a platform, dubbed as *K*mer2Vec, to embed DNA sequence of arbitrary length into points in a low-dimensional space based on the ordering of *K*mers in a training database of real-world DNA sequences. We demonstrated that such embedding learns concepts such as drug-resistance significantly better than classical *K*mer frequency representation. Our *K*mer2Vec platform opens many conceptual directions. The idea of learning biological models and representations from data is going to revolutionize many disciplines such as computational biology and bioinformatics. Similar to the trends in artificial intelligence that the availability of data helps learning

models such as classifiers, various fields in biology such as genomics is going to undergo the same trend. *K*mer2Vec is a first step into that direction and many more are yet to come.

Appendix A

Mathematical formulation of the compressive sensing (CS) detection and estimation algorithms

First we set up our mathematical model. Recall that there are N target bacteria of interest. Each microbial sample is characterized by a concentration vector $\mathbf{x} = [x_1, x_2, x_3, \ldots, x_N]^T$ containing the concentration x_i of bacterium i. While the total number of target species N might be large, a typical contaminated or infected sample will contain only a few target species K with significant concentration. When $K \ll N$, we say that the vector x is *sparse*. Experimental quantification of the amount of hybridization between the DNA of the microbial sample and M random probes produces the probe-binding vector $\mathbf{y} = [y_1, y_2, y_3, \ldots, y_M]^T$ containing the hybridization binding level y_j of the sample to probe j. By predicting the hybridization binding level ϕ_{ij} of probe i to bacterium j from the database of targets of interest, we can form the $M \times N$ hybridization affinity matrix $\mathbf{\Phi}$ (Fig. 1B and Fig. 2B). We refer to column i of the matrix $\mathbf{\Phi}$ by the column vector $\boldsymbol{\phi}_i$.

As we describe in Methods section, the probe-binding vector **y** will approximately form as a linear combination of the predicted hybridization affinities of the species in the reference genome database (the columns of the matrix $\mathbf{\Phi}$) weighted by their concentrations **x**, i.e.,

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{n}. \tag{A.1}$$

Here the vector $\mathbf{n}$ accounts for noise and modeling errors.

Two key goals of UMD are to detect the presence and estimate the concentrations $\mathbf{x}$ of a potentially large number N of reference microbial genomes in a sample given only a small

number $M \ll N$ of probe-binding vector measurements $\mathbf{y}$. Simply inverting the matrix $\mathbf{\Phi}$ is impossible in this case, since it has many more columns than rows. Fortunately, it is reasonable to assume that only a small number K of microbial genomes will be present in a given sample, in which case the concentration vector $\mathbf{x}$ is sparse with K nonzero and $N - K$ zero (or close to zero) entries; when $K < M$, one can hope to invert $\mathbf{\Phi}$ to estimate the K nonzero concentrations.

In order to apply the standard compressive sensing theory, the columns of $\mathbf{\Phi}$ should satisfy the so-called Restricted Isometry Property (RIP). It has been shown that a matrix satisfies the RIP if its columns are sufficiently *incoherent*, i.e., when the largest normalized inner product between any two columns of $\mathbf{\Phi}$, $\mu = \max_{\forall i,j:i\neq j} \frac{\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j}{||\boldsymbol{\phi}_i||_2 ||\boldsymbol{\phi}_j||_2}$, known as the coherence, is bounded above by a small constant. More specifically, it has been shown that $\mu < \frac{1}{2K-1}$ is a sufficient condition to exactly recover a K-sparse signal with only $M = cK\log(N/K)$, where c is a small constant, measurements in the noise-free scenario when $\mathbf{n} = 0$.

In the presence of noise, the same sufficiency bound holds if the magnitudes of the non-zero elements of x are sufficiently large compared to the noise variance. When these conditions hold, we can both detect the presence of bacteria and estimate their concentration using standard CS signal recovery algorithms. In the UMD platform, both the hybridization matrix $\mathbf{\Phi}$ and the sparse concentration vector $\mathbf{x}$ are non-negative. A more optimistic recovery bound has been proven to hold in this regime. In particular an alternative notion of incoherence is defined that improves the recovery guarantee. For an arbitrary matrix $\mathbf{\Phi}$, the one-sided coherence is defined as $\rho(\mathbf{\Phi}) = \max_{\forall i,j:i\neq j} \frac{\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j}{||\boldsymbol{\phi}_i||_2^2}$. It is shown that, for a non-negative matrix $\mathbf{\Phi}$, if a non-negative K-sparse solution exists such that $\rho(\mathbf{PD}) < \frac{1}{2K-1}$, then the solution is unique and CS recovery algorithms can find it. Here $\mathbf{D}$ is defined as the column ℓ_1 (sum) normalized matrix of $\mathbf{\Phi}$, and $\mathbf{P}$ is the column mean subtraction op-

erator. While the one-sided coherence recovery bound is based on the better-conditioned matrix $\mathbf{PD}$, it remains pessimistic, and better recovery performance is typically achieved in practice. The concentration vector $\mathbf{x}$ is recovered from the measurement vector $\mathbf{y}$ via a sparsity-penalized optimization of the form

$$\min_{\mathbf{x}} ||\mathbf{x}||_0, \text{ subject to } ||\mathbf{y} - \mathbf{\Phi x}||_2 < \sigma. \tag{A.2}$$

Here $||\mathbf{x}||_0$, known as the ℓ_0-norm, counts the number of non-zero values in the vector $\mathbf{x}$, and σ estimates the noise standard deviation. While this optimization problem has exponential complexity, a variety of different greedy algorithms have been developed to solve it approximately. Orthogonal Matching Pursuit (OMP) is an iterative greedy algorithm that, at each step, selects the column of $\mathbf{\Phi}$ that is most correlated with a residual vector from the previous iteration. The primary advantages of OMP are its simplicity and fast convergence. Moreover, if the sparsity level K (the number of bacteria) is known, then the algorithm can use it as the stopping criterion. We can leverage the fact that both the hybridization affinity matrix $\mathbf{\Phi}$ and the sparse concentration vector $\mathbf{x}$ are non-negative to improve the performance of OMP. We utilize the variant of OMP that is adapted to recover non-negative sparse solutions from non-negative sensing matrix $\mathbf{\Phi}$. Instead of directly working on $\mathbf{\Phi}$, this algorithm operates on the canonical matrix $\mathbf{PD}$, where $\mathbf{D}$ is the column ℓ_1-norm normalized version of matrix $\mathbf{\Phi}$ defined as: $\mathbf{D} = \mathbf{\Phi W}^{-1}$, where $\mathbf{W}$ is an $M \times M$ diagonal matrix containing the column sums of the hybridization matrix $\mathbf{\Phi}$ and $\mathbf{P}$ is the pre-conditioner matrix and can be chosen as any invertible $N \times N$ matrix. In the case of a positive matrix $\mathbf{D}$, an efficient preconditioning can be obtained by subtracting the weighted mean of each column of $\mathbf{D}$: $\mathbf{PD} = (I - \frac{(1-\epsilon)}{N}\mathbf{E})\mathbf{D}$, where $\mathbf{E}$ is $N \times N$ matrix of ones, $\mathbf{I}$ is the identity matrix, and $0 < \epsilon \ll 1$ is a weighing constant to make the $\mathbf{P}$ matrix invertible. Working with the preconditioned matrix $\mathbf{PD}$ does not change the solution of the problem; however, it significantly improves the OMP algorithm behavior and performance

guarantees. This OMP algorithm variant is given in Alg. 1:

Algorithm 3: Non-negative OMP

Data: $\mathbf{D}$, $\mathbf{y}$

Result: $\mathbf{x}^i$

Initialization;

The temporary solution: $\mathbf{x}_i = \mathbf{0}$; The temporary residual: $\mathbf{r}^i = \mathbf{y}$; The temporary solution support: $\mathcal{S}^i = support\{\mathbf{x}^i\} = \emptyset$

while $||\mathbf{r}||_2^2 < T$ **do**

Sweep: $\forall j \in [1, 2, \ldots, N]$: find $\epsilon(j) = \min_{x \geq 0} ||\mathbf{D}_j \mathbf{x}_j - \mathbf{r}||_2^2$;

Update support: find j_0 such that $\forall j \in \mathcal{S}^i$, $\epsilon(j_0) \leq \epsilon(j)$, and update $\mathcal{S}^i = \mathcal{S}^{i-1} \bigcup \{j_0\}$;

Update solution: compute $x^i : x^i = \min_{\mathbf{x}} ||\mathbf{D}\mathbf{x} - \mathbf{y}||_2^2$ subject to $support\{\mathbf{x}^i\} = \mathcal{S}^i$ and $\mathbf{x} \geq 0$;

Update residual: compute $\mathbf{r}^i = \mathbf{y} - \mathbf{D}\mathbf{x}^i$;

end

We set the stopping criterion to $T = 2 \times 10^{-1}$ for recovering the experimentally obtained hybridization affinity vectors in Fig. 3 and the numerically simulated hybridization affinity vector corrupted by modeling noise level in Fig. 4. $T = 7 \times 10^{-3}$ was selected to recover the simulated hybridization affinity vector corrupted by the experimental noise level simulations in Fig. 4. In reporting the similarity of the experimentally measured hybridization affinity vectors in Fig. 3D, we have reported the similarity of measured hybridization affinity vector $\mathbf{y}$ in each experiment to the bacteria i in the dictionary as the inner product of the normalized affinity vector $\frac{\mathbf{y}}{||\mathbf{y}||_2}$ and the i^{th} column of the normalized preconditioned matrix $\mathbf{PD}$. The inner product is a unit-less number in the range $[-1, 1]$, where 1 indicates the highest similarity and -1 the lowest similarity.

While OMP attempts to solve the exact ℓ_0-norm penalized problem, another important class of CS recovery algorithms relaxes the problem by replacing the ℓ_0-norm with the more tractable ℓ_1-norm

$$\min_{\mathbf{x}} ||\mathbf{x}||_1, \text{ subject to } ||\mathbf{y} - \mathbf{\Phi x}||_2 < \sigma. \tag{A.3}$$

Here $||\mathbf{x}||_1$ sums the absolute value of the entries in the vector $\mathbf{x}$. The ℓ_1-norm penalized optimization above is convex and reduces to a linear program known as Basis Pursuit Denoising (BPDN). Surprisingly the ℓ_1-norm penalized problem, under the same sufficiency conditions on mutual coherence stated above, can exactly recover a K-sparse signal from the noisy measurements $\mathbf{y}$ with $M = cK\log(N/K)$ measurements. We can leverage the fact that both the hybridization affinity matrix $\mathbf{\Phi}$ and the sparse concentration vector $\mathbf{x}$ are non-negative to improve the performance of BPDN by adding a positivity constraint

$$\min_{\mathbf{x}} ||\mathbf{x}||_1, \text{ subject to } ||\mathbf{y} - \mathbf{\Phi x}||_2 < \sigma \text{ and } \mathbf{x} \geq 0. \tag{A.4}$$

By replacing the objective function with the squared norm $||\mathbf{x}||_1^2$ and bringing the constraint into the cost function using a Lagrangian multiplier $\lambda > 0$, we have

$$\min_{\mathbf{x}} ||\mathbf{x}||_1 + \lambda||\mathbf{y} - \mathbf{\Phi x}||_2, \text{ subject to } \mathbf{x} \geq 0. \tag{A.5}$$

Since the ℓ_1-norm of a non-negative valued vector $\mathbf{x}$ is $||\mathbf{x}||_1 = \sum_{i=1}^{N} x_i$, we can reformulate the above problems as a non-negative least squares problem

$$\min_{\mathbf{x}} ||\hat{\mathbf{y}} - \hat{\mathbf{\Phi}}\mathbf{x}||_2^2, \text{ subject to } \mathbf{x} \geq 0. \tag{A.6}$$

that can be solved using using standard methods (e.g., the lsqnonneg command in Matlab). Here $\hat{\mathbf{y}} = \begin{bmatrix} \sqrt{\lambda}\mathbf{y} \\ 0 \end{bmatrix}$, and $\hat{\mathbf{\Phi}} = \begin{bmatrix} \sqrt{\lambda}\mathbf{\Phi} \\ 1 \ldots 1 \end{bmatrix}$, where the last column penalizes the ℓ_1-norm. The parameter λ strikes the balance between the sparsity level and our trust in the measurements (signal to noise ratio). In all of our experiments, we used $\lambda = 400$.

Appendix B

Human pathogenic bacterial species dataset

To evaluate the UMD platform for genus level bacterial detection, we selected 40 species from 40 different genera that are listed among most commonly pathogenic to humans by the Center for Disease Control and Prevention (CDC). The genome sequence of the following strains were obtained from the NCBI website:

Acinetobacter baumannii ATCC 17978,

Aeromonas salmonicida subsp. salmonicida A449,

Bacteroides fragilis 638R,

Bacillus cereus ATCC 14579,

Bartonella henselae str. Houston-1,

Bifidobacterium dentium Bd1,

Bordetella pertussis Tohama,

Borrelia burgdorferi B31,

Brucella abortus S19,

Campylobacter jejuni subsp. *jejuni* 81116,

Clostridium botulinum B1 str. *Okra*,

Corynebacterium jeikeium K411,

Coxiella burnetii RSA 331,

Cupriavidus metallidurans CH34,

Enterobacter aerogenes EA1509E,

Enterococcus faecalis V583,

Escherichia coli str. K-12 substr. MG1655,

Francisella tularensis subsp. *holarctica* LVS,

Fusobacterium nucleatum subsp. *nucleatum* ATCC 25586,

Haemophilus influenzae F3047, *Helicobacter pylori* B38,

Klebsiella pneumoniae 342,

Legionella pneumophila str. *Corby*,

Leptospira interrogans serovar Copenhageni str. *Fiocruz* L1-130,

Listeria monocytogenes 08-5578,

Micrococcus luteus NCTC 2665, *Mycobacterium leprae* TN,

Mycoplasma pneumoniae M129,

Neisseria meningitidis MC58,

Prevotella melaninogenica ATCC 25845,

Propionibacterium acnes KPA171202,

Proteus mirabilis HI4320,

Pseudomonas aeruginosa LESB58,

Rickettsia rickettsii str. *Iowa*,

Salmonella enterica subsp. *enterica serovar Paratyphi* A str. ATCC 9150,

Serratia proteamaculans 568, *Shigella sonnei* Ss046,

Staphylococcus aureus subsp. *aureus* USA300 FPR3757,

and *Vibrio cholerae* MJ-1236, *Yersinia pestis* CO92.

For species-level bacterial detection, the following 24 different strains from the *Staphylococcus* genus were selected to perform the simulaitons:

Staphylococcus arlettae CVD059 SARL-c1,

Staphylococcus aureus subsp. *aureus* NCTC 8325,

Staphylococcus auricularis strain DSM 20609,

Staphylococcus capitis subsp. *capitis* strain AYP1020,

Staphylococcus carnosus subsp. *carnosus* TM300,

Staphylococcus cohnii subsp.*cohnii* strain 532,

Staphylococcus epidermidis ATCC 12228,

Staphylococcus equorum strain KS1039,

Staphylococcus gallinarum strain DSM 20610,

Staphylococcus haemolyticus JCSC1435,

Staphylococcus hominis subsp. *hominis* C80,

Staphylococcus hyicus strain ATCC 11249,

Staphylococcus lentus F1142 s6-trimmed-contig-1,

Staphylococcus lugdunensis HKU09-01,

Staphylococcus massiliensis CCUG 55927,

Staphylococcus pettenkoferi VCU012,

Staphylococcus pseudintermedius HKU10-03,

Staphylococcus saprophyticus subsp. *saprophyticus* ATCC 15305,

Staphylococcus schleiferi strain 2317-03,

Staphylococcus sciuri subsp. *sciuri* strain DSM 20345,

Staphylococcus simiae CCM 7213 contig00565,

Staphylococcus simulans ACS-120-V-Sch1,

Staphylococcus vitulinus F1028 S-vitulinus-F1028-0001,

and *Staphylococcus warneri* SG1.

For species-level bacterial detection, the following 23 different strains from the *Vibrio* genus were selected to perform the simulaitons:

Vibrio anguillarum 775,

Vibrio brasiliensis LMG 20546 VIBR0546-99,

Vibriocholerae O1 biovar El Tor str. N16961,

Vibrio cyclitrophicus FF75 Ctg1,

Vibrio diazotrophicus NBRC 103148,

Vibrio ezurae NBRC 102218,

Vibrio furnissii NCTC 11218,

Vibrio genomosp. F6 str. FF-238,

Vibrio genomosp. F10 str. ZF-129,

Vibrio harveyi ATCC BAA-1116,

Vibrio litoralis DSM 17657,

Vibrio maritimus strain: JCM 19235,

Vibrio metschnikovii CIP 69.14 VIB.Contig153,

Vibrio nigripulchritudo str. SFn1,

Vibrio orientalis CIP 102891 ATCC 33934 strain *CIP* 102891,

Vibrio pacinii DSM 19139 BS19DRAFT-scaffold00001.1-C,

Vibrio parahaemolyticus RIMD 2210633,

Vibrio proteolyticus NBRC 13287,

Vibrio rhizosphaerae DSM 18581,

Vibrio scophthalmi LMG 19158 VIS19158-99,

Vibrio splendidus LGP32,

Vibrio tubiashii ATCC 19109,

and *Vibrio vulnificus* YJ016.

Appendix C

UMD supplementary figures

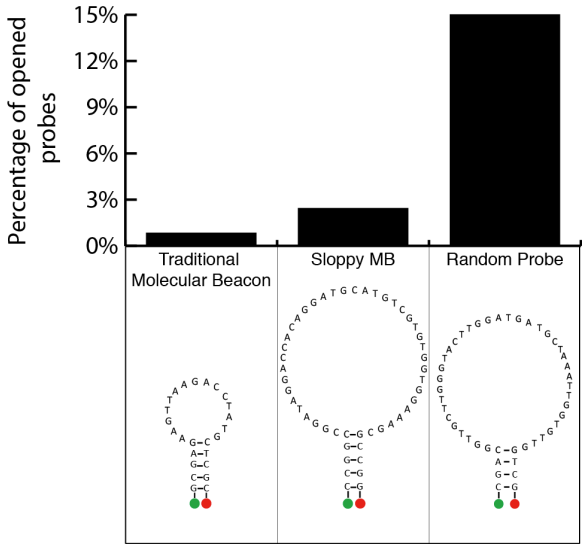

Figure C.1 : Comparison of the sloppiness of random probes to other molecular beacons. The predicted number of opened probes when the *E. coli* genome is exposed to a specific (in the same thermodynamic condition as the experiment) traditional molecular beacon (MB) (*20*), sloppy MB (*19*), and an MB created according to our random design rules. While the nucleotides in the loop were determined randomly, the percentage of opened beacons (hybridization affinity to *E. coli*) is substantially increased for the random probe MB, thanks to the choice of stem/loop length.

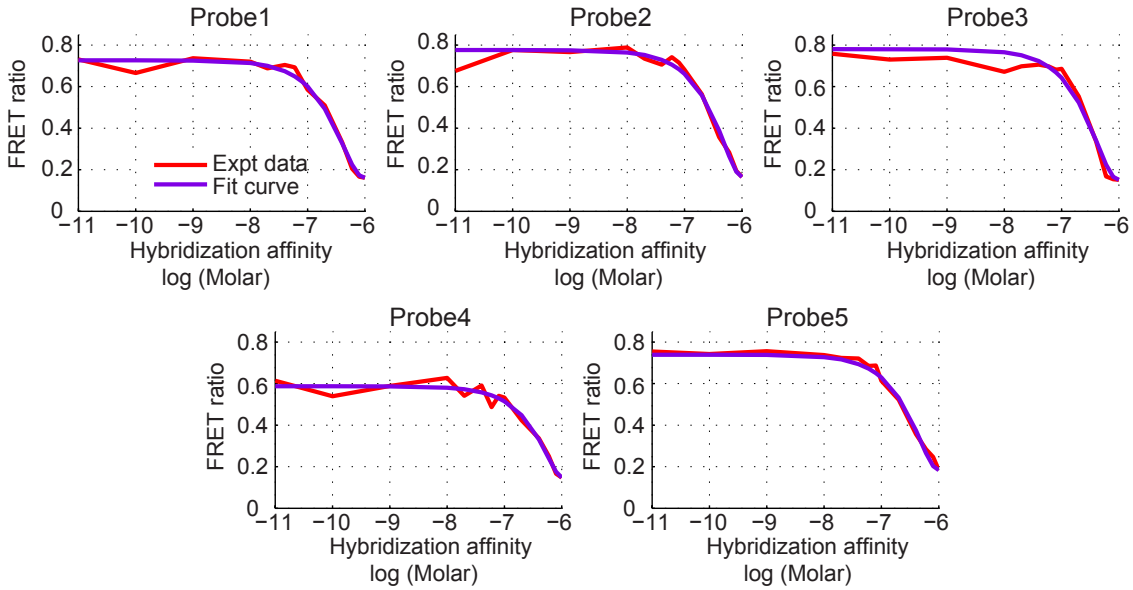

Figure C.2 : Random probes' characteristic curves. Probe characteristic curves for random probes 1–5 for determining the experimental relationship between FRET ratio and open MB concentration (hybridization affinity). The characteristic curve is fitted to the measured FRET ratio as a function of the concentration of the probe's exact complement for five random probes.

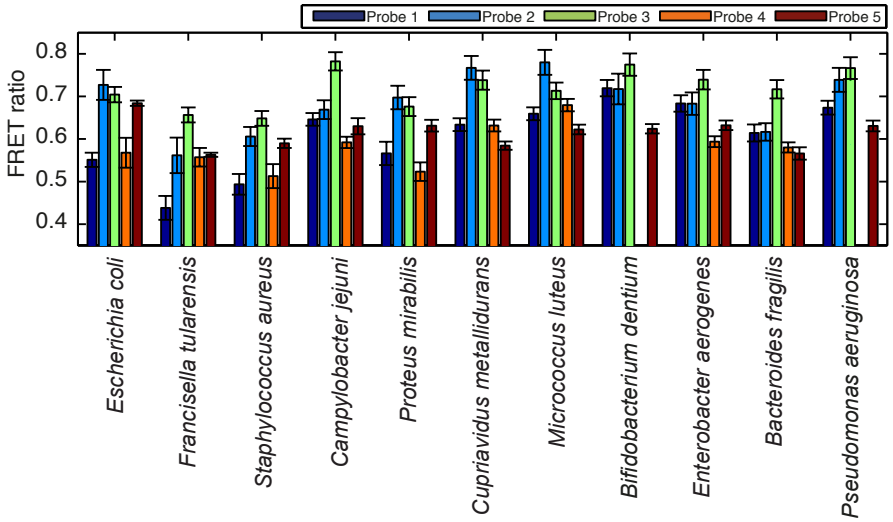

Figure C.3 : Experimentally measured FRET ratios to quantify hybridization between 11 bacterial DNA samples and random probes 1–5. The error bars represent the standard error of the mean (s.e.m.).

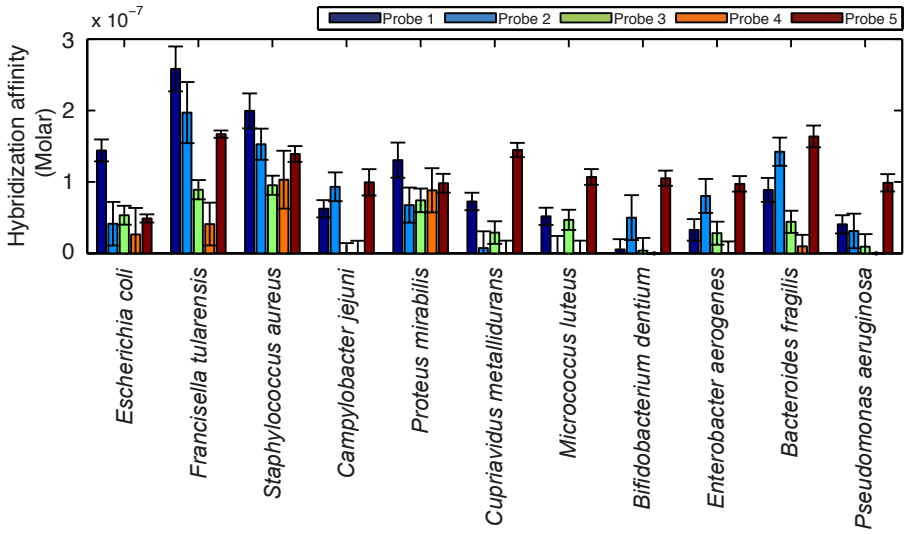

Figure C.4 : Hybridization affinity between 11 bacterial DNA samples and random probes 1–5. converted from FRET ratio through the probe characteristic curve fit equations. The error bars represent the standard error of the mean (s.e.m.).

	Probe 1	Probe 2	Probe 3	Probe 4	Probe 5
$\log_{10}(a)$	14.14	11.56	13.60	9.97	12.45
b	1.13	0.90	0.51	1.19	1.47
n	2.39	1.97	2.35	1.71	2.09
FRET_0	0.161	0.165	0.152	0.148	0.183
R^2	0.9877	0.9770	0.9688	0.9666	0.9899
RMSE	2.37E-2	3.33E-2	4.04E-2	2.90E-2	2.05E-2

Table C.1 : The fitted parameters to the probes' characteristic curves. Characteristic curve parameters a, b, and n for five test random probes, and the associated curve fit performance criteria, R^2 and root mean square error (RMSE) are tabulated. Parameters were fit to the following curve: $\text{FRET}(c) = \text{FRET}_0 + \frac{a}{(1+b(10^{-6}-c)^{-n})}$ (*22*).

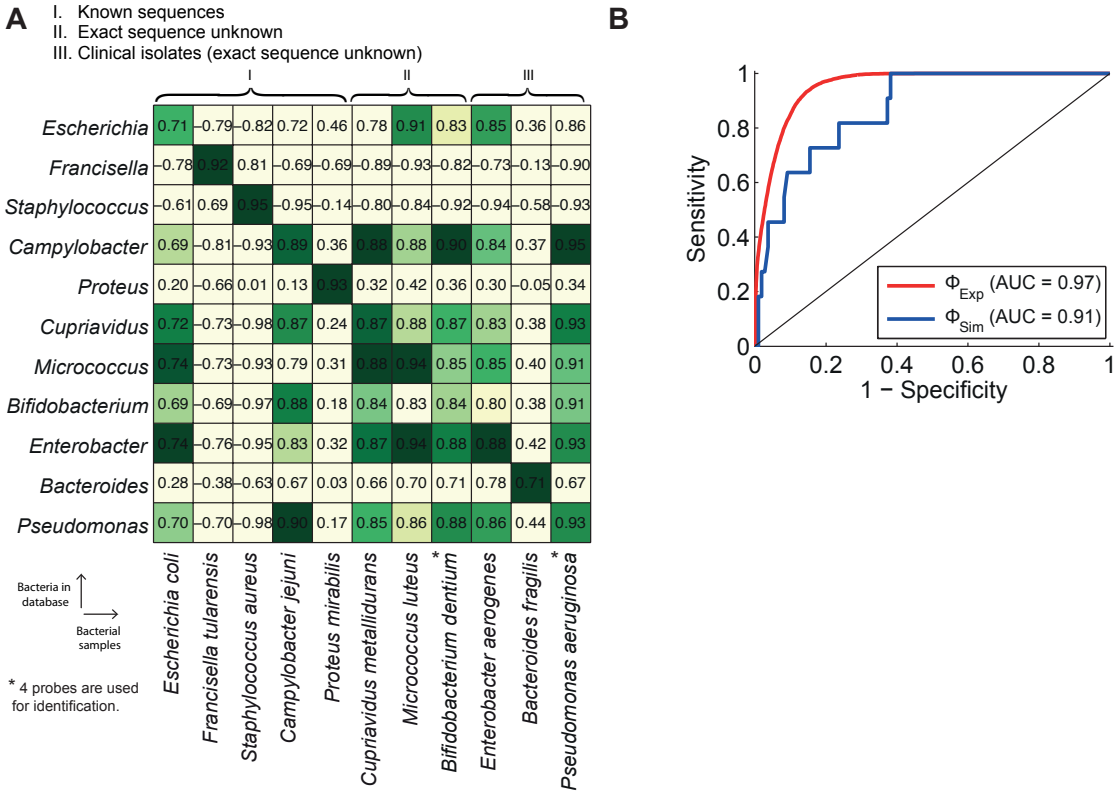

Figure C.5 : Detection performance of 11 bacterial samples using five random probes. (**A**) The heat map and raw values of inner product of the experimentally obtained hybridization affinity and predicted hybridization affinity (by thermodynamic model) for eleven DNA samples as a measure of the similarity of the probe measurements. For detection of the two strains marked with asterisk (*), only four random probes (Probes 1, 2, 3, and 5) were utilized. (**B**) The average receiver operating characteristic (ROC) curve of UMD in detecting eleven bacteria, assuming the independence of different test trials. Each point on the curve corresponds to a threshold value between $[-1, 1]$. UMD achieves high values of the area under the curve using experimentally obtained $\mathbf{\Phi}_{\text{exp}}$ (red line) and simulated $\mathbf{\Phi}_{\text{sim}}$ using the DNA software.

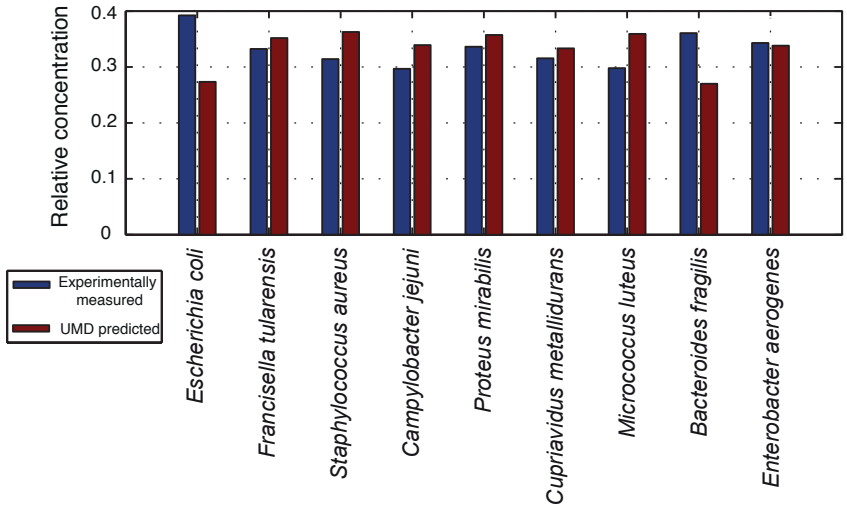

Figure C.6 : Comparison of the predicted concentrations of bacterial DNA with the experimentally measured values. Concentration of bacterial samples are estimated by compressive sensing recovery algorithm.

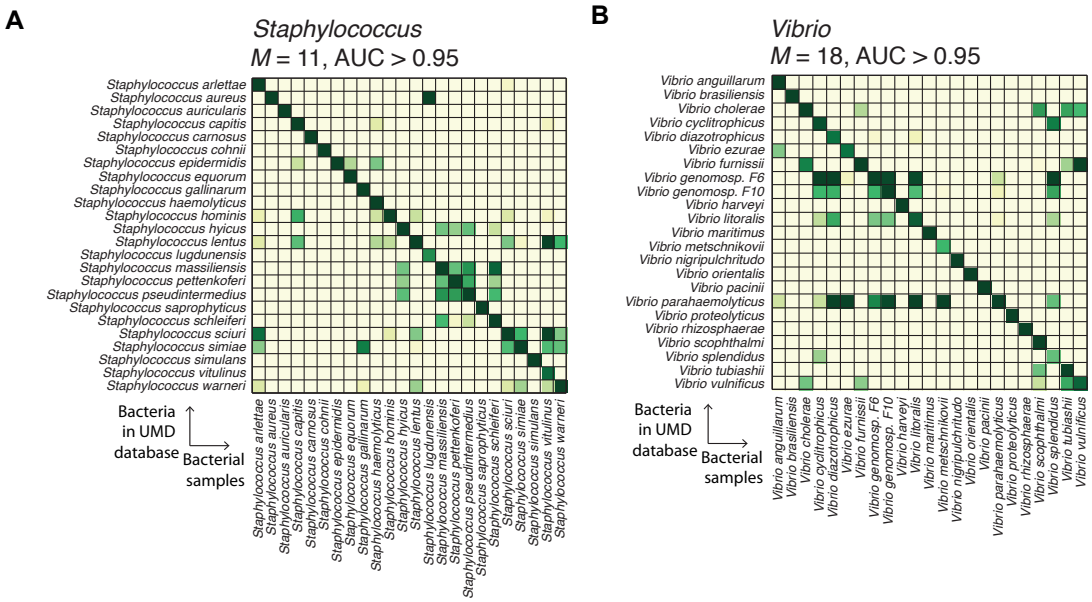

Figure C.7 : Performance of UMD in species-level recovery of 24 strains of Staphylococcus and 23 strains of Vibrio. (**A**) UMD's confusion matrix in identifying 24 strains of *Staphylococcus*. With $M = 11$ random probes, UMD identifies all the species of *Staphylococcus* genera with AUC > 0.95. (**B**) UMD's confusion matrix in identifying 23 strains of *Vibrio*. With $M = 18$ random probes, UMD identifies all the species of *Vibrio* with AUC > 0.95.

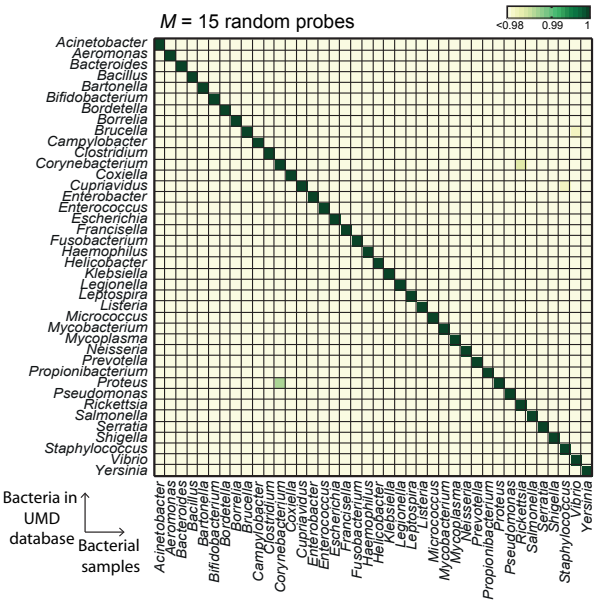

Figure C.8 : Performance of UMD at identifying pathogens in genus-level using 15 GPS probes. The UMD detection performance increases with a large number of GPS probes. With M = 15 probes, UMD alleviates all the potential false positives and provides a more noise-robust identification of pathogens at the genus-level.

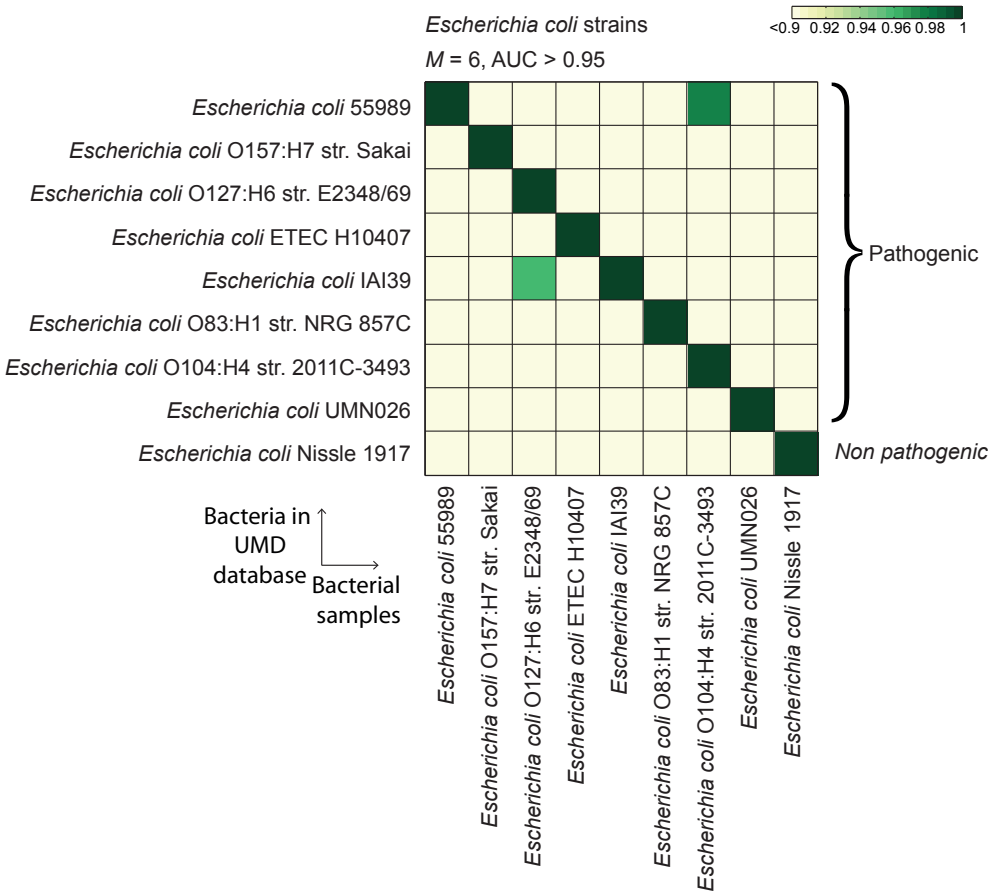

Figure C.9 : Performance of UMD in identifying eight pathogenic and one nonpathogenic E.coli strains using GPS probes. GPS selects 6 UMD probes that differentiate between eight pathogenic and one nonpathogenic *E.coli* strains *in silico*.

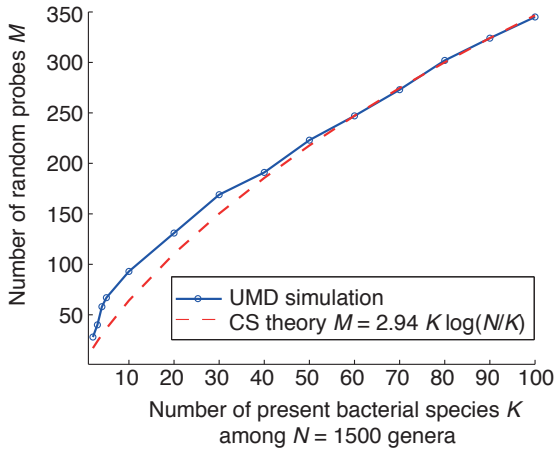

Figure C.10 : Performance of UMD in identifying the composition of several complex samples. We simulate the number of required random probes M to identify the composition of complex bacterial samples with 100% accuracy using UMD. The complexity of each sample is measured by the number of present bacterial species K among a total of N = 1500 bacterial genera. To obtain each data point in the curve, we randomly selected K bacterial genera from the database and created a sample by mixing K sample species from each genera with equal concentrations. We used UMD to recover the composition of the mixture using random probes. We repeated this same experiment for 1000 trials and reported the minimum number of probes that recovered all 1000 mixtures accurately. UMD requires orders of magnitudes less number of random probes than the size of the database N (i.e., number of probes typically required by conventional methods) to recover complex samples containing mixtures of K species (with equal concentrations) among N = 1500 genera. Furthermore, number of required UMD probes closely ($R^2 = 0.98$) follows the number of probes predicted by the compressive sensing theory $M = ck\log(N/K)$ with $c = 2.94$.

Bibliography

[1] A. Aghazadeh, A. Y. Lin, M. A. Sheikh, A. L. Chen, L. M. Atkins, C. L. Johnson, J. F. Petrosino, R. A. Drezek, and R. G. Baraniuk, "Universal microbial diagnostics using random DNA probes," *Sci. Adv.*, vol. 2, p. e1600025, Sep. 2016.

[2] A. Aghazadeh, M. Golbabaee, A. S. Lan, and R. G. Baraniuk, "Insense: Incoherent sensor selection for sparse signals," *arXiv preprint arXiv:1702.07670*, 2017.

[3] A. Aghazadeh, A. S. Lan, S. A., and R. G. Baraniuk, "Rhash: Robust hashing via ℓ_∞-norm distortion," *IJCAI-2017*, 2017.

[4] S. Tyagi, S. A. Marras, and F. R. Kramer, "Wavelength-shifting molecular beacons," *Nature biotechnology*, vol. 18, no. 11, pp. 1191–1196, 2000.

[5] M. M. Mhlanga and S. Tyagi, "Using trna-linked molecular beacons to image cytoplasmic mrnas in live cells," *Nature protocols*, vol. 1, no. 3, pp. 1392–1398, 2006.

[6] J. SantaLucia Jr and D. Hicks, "The thermodynamics of dna structural motifs," *Annu. Rev. Biophys. Biomol. Struct.*, vol. 33, pp. 415–440, 2004.

[7] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pur. Appl. Math.*, vol. 59, pp. 1207–1223, Aug. 2006.

[8] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[9] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, pp. 118–124, Jul. 2007.

[10] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *Information Theory, IEEE Transactions on*, vol. 47, no. 7, pp. 2845–2862, 2001.

[11] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *Information Theory, IEEE Transactions on*, vol. 57, no. 7, pp. 4680–4688, 2011.

[12] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.

[13] J. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4655–4666, Dec. 2007.

[14] S. Tyagi, F. R. Kramer, *et al.*, "Molecular beacons: probes that fluoresce upon hybridization," *Nature Biotechnol.*, vol. 14, no. 3, pp. 303–308, 1996.

[15] Y. Kim, D. Sohn, and W. Tan, "Molecular beacons in biomedical detection and clinical diagnosis," *International journal of clinical and experimental pathology*, vol. 1, no. 2, p. 105, 2008.

[16] S. Chakravorty, B. Aladegbami, M. Burday, M. Levi, S. A. Marras, D. Shah, H. H. El-Hajj, F. R. Kramer, and D. Alland, "Rapid universal identification of bacterial pathogens from clinical cultures by using a novel sloppy molecular beacon melting temperature signature technique," *J. Clin. Microbiol.*, vol. 48, no. 1, pp. 258–267, 2010.

[17] M. Klompas, D. S. Yokoe, and R. A. Weinstein, "Automated surveillance of health care–associated infections," *Clin. Infect. Dis.*, vol. 48, no. 9, pp. 1268–1275, 2009.

[18] V. N. Pinto, "Bioterrorism: Health sector alertness," *J. Nat. Sc. Biol. Med.*, vol. 4, no. 1, p. 24, 2013.

[19] B. Van Dorst, J. Mehta, K. Bekaert, E. Rouah-Martin, W. De Coen, P. Dubruel, R. Blust, and J. Robbens, "Recent advances in recognition elements of food and environmental biosensors: A review," *Biosensors Bioelectronics*, vol. 26, no. 4, pp. 1178–1194, 2010.

[20] G. S. Martin, D. M. Mannino, S. Eaton, and M. Moss, "The epidemiology of sepsis in the united states from 1979 through 2000," *New England Journal of Medicine*, vol. 348, no. 16, pp. 1546–1554, 2003.

[21] D. L. Hoyert, J. Xu, *et al.*, "Deaths: preliminary data for 2011," *Natl Vital Stat Rep*, vol. 61, no. 6, pp. 1–65, 2012.

[22] A. Pechorsky, Y. Nitzan, and T. Lazarovitch, "Identification of pathogenic bacteria in blood cultures: comparison between conventional and pcr methods," *J. Microbiol. Methods*, vol. 78, no. 3, pp. 325–330, 2009.

[23] P. M. Dark, P. Dean, G. Warhurst, *et al.*, "Bench-to-bedside review: the promise of rapid infection diagnosis during sepsis using polymerase chain reaction-based pathogen detection," *Crit. Care*, vol. 13, no. 4, pp. 217–217, 2009.

[24] S. Sontakke, M. B. Cadenas, R. G. Maggi, P. P. V. Diniz, and E. B. Breitschwerdt, "Use of broad range16s rdna pcr in clinical microbiology," *J. Microbiol. Methods*, vol. 76, no. 3, pp. 217–225, 2009.

[25] C. D. Sibley, G. Peirano, and D. L. Church, "Molecular methods for pathogen and microbial community detection and characterization: current and potential application in diagnostic microbiology," *Infect. Genet. Evol.*, vol. 12, no. 3, pp. 505–521, 2012.

[26] S. Riedel and K. C. Carroll, "Blood cultures: key elements for best practices and future directions," *Journal of infection and chemotherapy*, vol. 16, no. 5, pp. 301–316, 2010.

[27] M. Paolucci, M. P. Landini, and V. Sambri, "Conventional and molecular techniques for the early diagnosis of bacteremia," *International Journal of Antimicrobial Agents*, vol. 36, pp. S6–S16, 2010.

[28] M. Bauer and K. Reinhart, "Molecular diagnostics of sepsiswhere are we today?," *International Journal of Medical Microbiology*, vol. 300, no. 6, pp. 411–413, 2010.

[29] R. P. Peters, M. A. van Agtmael, S. A. Danner, P. H. Savelkoul, and C. M. Vandenbroucke-Grauls, "New developments in the diagnosis of bloodstream infections," *The Lancet infectious diseases*, vol. 4, no. 12, pp. 751–760, 2004.

[30] Centers for Disease Control and Prevention, 2012.

[31] M. A. Sheikh, *Genomic Detection Using Sparsity-inspired Tools*. PhD thesis, Rice University, 2010.

[32] Ž. Jeričević and Ž. Kušter, "Non-linear optimization of parameters in michaelis-menten kinetics," *Croat. Chem. Acta*, vol. 78, no. 4, pp. 519–523, 2005.

[33] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations," *Information Theory,*

IEEE Transactions on, vol. 54, no. 11, pp. 4813–4820, 2008.

[34] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 445–460, 2010.

[35] R. G. Baraniuk, "More is less: signal processing and the data deluge," *Science(Washington)*, vol. 331, no. 6018, pp. 717–719, 2011.

[36] M. A. Sheikh, O. Milenkovic, and R. G. Baraniuk, "Designing compressive sensing dna microarrays," *IEEE Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pp. 141–144, 2007.

[37] M. Mohtashemi, D. K. Walburger, M. W. Peterson, F. N. Sutton, H. B. Skaer, and J. C. Diggans, "Open-target sparse sensing of biological agents using dna microarray," *BMC bioinformatics*, vol. 12, no. 1, p. 314, 2011.

[38] H. J. Chung, C. M. Castro, H. Im, H. Lee, and R. Weissleder, "A magneto-DNA nanoparticle system for rapid detection and phenotyping of bacteria," *Nat. Nanotechnol.*, vol. 8, pp. 369–375, Nov. 2013.

[39] J. Sambrook, E. F. Fritsch, and T. Maniatis, *Molecular cloning: a laboratory manual.* No. Ed. 2, Cold spring harbor laboratory press, 1989.

[40] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, pp. 451–462, Feb. 2009.

[41] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *Proc. IEEE. Conf. Dec. Ctrl. (CDC'10)*, pp. 2572–2577, Feb. 2010.

[42] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Trans. Signal Process.*, vol. 62, pp. 1135–1146, Mar. 2014.

[43] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, pp. 21–30, Mar. 2008.

[44] E. J. Candès, "Compressive sampling," in *Proc. Intl. Cong. Math,*, vol. 3, pp. 1433–1452, Aug. 2006.

[45] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Stat. Soc. B Met.*, pp. 267–288, Jan. 1996.

[46] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Natl. Acad. Sci.*, vol. 106, no. 45, pp. 18914–18919, 2009.

[47] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, pp. 2231–2242, Sep. 2004.

[48] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[49] C. Herzet, C. Soussen, J. Idier, and R. Gribonval, "Exact recovery conditions for sparse representations with partial support information," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7509–7524, 2013.

[50] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *IEEE Trans. Info. Theory*, vol. 52, pp. 255–261, Dec. 2006.

[51] J. Ranieri, A. Vincenzi, A. Chebira, D. Atienza, and M. Vetterli, "EigenMaps: Algorithms for optimal thermal maps extraction and sensor placement on multicore processors," in *Proc. ACM/EDAC/IEEE Des. Auto. Conf. (DAC'12)*, pp. 636–641, Jun. 2012.

[52] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proc. Ann. ACM Theory Comput. (STOC'08)*, pp. 45–54, May 2008.

[53] D. Golovin, M. Faulkner, and A. Krause, "Online distributed sensor selection," in *Proc. ACM Intl. Conf. Inf. Process. Sens. Net.*, pp. 220–231, Apr. 2010.

[54] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," *arXiv preprint arXiv:1102.3975*, 2011.

[55] D. M. Steinberg and W. G. Hunter, "Experimental design: Review and comment," *Technometrics*, vol. 26, pp. 71–97, May 1984.

[56] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, Feb. 2008.

[57] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proc. Intl. ACM Conf. Inf. Process. Sens. Net.*, pp. 36–45, Mar. 2004.

[58] I. Ford, D. M. Titterington, and C. P. Kitsos, "Recent advances in nonlinear experimental design," *Technometrics*, vol. 31, no. 1, pp. 49–60x, 1989.

[59] S. P. Chepuri and G. Leus, "Sparsity-promoting sensor selection for non-linear measurement models," *IEEE Trans. Signal Process.*, vol. 63, pp. 684–698, Feb. 2015.

[60] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Found. Comput. Math.*, vol. 9, pp. 317–334, Jun. 2009.

[61] D. M. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE J. Sel. Top. Signal Process.*, vol. 4, pp. 310–316, Apr. 2010.

[62] A. Amini and F. Marvasti, "Deterministic construction of binary, bipolar, and ternary compressed sensing matrices," *IEEE Trans. Inf. Theory*, vol. 57, pp. 2360–2370, Apr. 2011.

[63] T. Strohmer and R. W. Heath, "Grassmannian frames with applications to coding and communication," *Appl. Comput. Harmon. Anal.*, vol. 14, pp. 257–275, May 2003.

[64] J. Tropp, I. S. Dhillon, R. W. Heath Jr, and T. Strohmer, "Designing structured tight frames via an alternating projection method," *IEEE Trans. Inf. Theory*, vol. 51, pp. 188–209, Jan. 2005.

[65] M. Elad, "Optimized projections for compressed sensing," *IEEE Trans. Signal Process.*, vol. 55, no. 12, pp. 5695–5702, 2007.

[66] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Process.*, vol. 18, pp. 1395–1408, Jun. 2009.

[67] D. Balageas, C. P. Fritzen, and A. Güemes, *Structural Health Monitoring*, vol. 493. Wiley Online Library, 2006.

[68] X. Q. Zhou, Y. Xia, and S. Weng, "L_1 regularization approach to structural damage detection using frequency data," *Struct. Health Monit.*, pp. 14759217–15604386, 2015.

[69] D. Sen, A. Aghazadeh, A. Mousavi, S. Nagarajaiah, and R. G. Baraniuk, "Sparsity-based data-driven approaches for damage detection in plates," *Mech. Syst. Signal Process.*, Under Review.

[70] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," tech. rep., Technical University of Denmark, 2008.

[71] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Math. Program.*, vol. 137, pp. 91–129, Aug. 2013.

[72] Y. Nesterov, "Gradient methods for minimizing composite objective function," 2007.

[73] Y. Chen and X. Ye, "Projection onto a simplex," *arXiv preprint arXiv:1101.6081*, 2011.

[74] W. Wang and M. A. Carreira-Perpinán, "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," *arXiv preprint arXiv:1309.1541*, 2013.

[75] L. Condat, "Fast projection onto the simplex and the ℓ_1-ball," 2014.

[76] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, Aug. 1998.

[77] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. Annual Symposium on Computational Geometry*, pp. 253–262, June 2004.

[78] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe lsh: efficient indexing for high-dimensional similarity search," in *Proc. Int. Conf. on Very large data bases*, pp. 950–961, Sep. 2007.

[79] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," *arXiv preprint arXiv:1408.2927*, 2014.

[80] Y. Weiss, R. Fergus, and A. Torralba, "Multidimensional spectral hashing," in *European Conf. on Computer Vision*, pp. 340–353, Oct. 2012.

[81] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Adv. in Neural Info. Proces. Systems*, pp. 1753–1760, Dec. 2009.

[82] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Adv. in Neural Info. Proces. Systems*, pp. 1042–1050, Dec. 2009.

[83] Q. Y. Jiang and W. J. Li, "Scalable graph hashing with feature transformation," in *Proc Int. Joint Conf. on Artificial Intelligence*, Sep. 2015.

[84] I. Jolliffe, *Principal Component Analysis*. Wiley Online Library, 2002.

[85] W. Kong and W. J. Li, "Isotropic hashing," in *Adv. in Neural Info. Proces. Systems*, pp. 1646–1654, Dec. 2012.

[86] W. Liu, J. Wang, S. Kumar, and S. Chang, "Hashing with graphs," in *Proc. Intl. Conf. on Machine Learning*, pp. 1–8, July 2011.

[87] S. Chatterjee and A. S. Hadi, *Sensitivity Analysis in Linear Regression*, vol. 327. John Wiley & Sons, 2009.

[88] D. Du and P. M. Pardalos, *Minimax and Applications*, vol. 4. Springer, 2013.

[89] R. Hartley and F. Schaffalitzky, "L_∞ minimization in geometric reconstruction problems," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 504–509, June 2004.

[90] H. Xu, C. Caramanis, and S. Mannor, "Robustness and regularization of support vector machines," *The Journal of Machine Learning Research*, vol. 10, pp. 1485–1510, 2009.

[91] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," 1998.

[92] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, pp. 1–122, Jan. 2011.

[93] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, pp. 101–111, Feb. 1960.

[94] J. Heo, Y. Lee, J. He, S. Chang, and S. Yoon, "Spherical hashing," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2957–2964, June 2012.

[95] F. X. Yu, S. Kumar, Y. Gong, and S. Chang, "Circulant binary embedding," in *Proc. Int. Conf. on Machine Learning*, pp. 946–954, June 2014.

[96] Y. Plan and R. Vershynin, "Dimension reduction by random hyperplane tessellations," *Discrete & Computational Geometry*, vol. 51, pp. 438–461, Mar. 2014.

[97] Y. Nesterov, "Gradient methods for minimizing composite objective function," tech. rep., Université Catholique de Louvain, Sep. 2007.

[98] C. Studer, T. Goldstein, W. Yin, and R. G. Baraniuk, "Democratic representations," *arXiv preprint arXiv:1401.3420*, 2014.

[99] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, pp. 1495–1502, Apr. 2007.

[100] C. Hegde, A. C. Sankaranarayanan, W. Yin, and R. G. Baraniuk, "Numax: A convex approach for learning near-isometric linear embeddings," *IEEE Trans. Signal Processing*, vol. 63, pp. 6109–6121, Nov. 2015.

[101] J. He, S. Kumar, and S. Chang, "On the difficulty of nearest neighbor search," in *Proc. Int. Conf. on Machine Learning*, pp. 1127–1134, June 2012.

[102] M. Ledoux and M. Talagrand, *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag, 1991.

[103] M. J. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (Lasso)," *IEEE Trans. Information Theory*, vol. 55, pp. 2183–2202, May 2009.

[104] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[105] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[106] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. Annual ACM Symposium on Theory of Computing*, pp. 604–613, May 1998.

[107] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, pp. 81–93, June 1938.

[108] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, pp. 91–110, Nov. 2004.

[109] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graphics*, vol. 25, pp. 835–846, July 2006.

[110] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.

[111] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *Proc. Int. Conf. on Machine Learning*, pp. 353–360, June 2011.

[112] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, "A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation," *SIAM Journal on Scientific Computing*, vol. 32, pp. 1832–1857, June 2010.

[113] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PloS one*, vol. 10, no. 11, p. e0141287, 2015.

[114] D. Kimothi, A. Soni, P. Biyani, and J. M. Hogan, "Distributed representations for biological sequence analysis," *arXiv preprint arXiv:1608.05949*, 2016.

[115] P. Ng, "dna2vec: Consistent vector representations of variable-length k-mers," *arXiv preprint arXiv:1701.06279*, 2017.

[116] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[117] C. A. Arias, D. Panesso, D. M. McGrath, X. Qin, M. F. Mojica, C. Miller, L. Diaz, T. Tran, S. Rincon, and E. M. Barbu, "Genetic basis for in vivo Daptomycin resistance in Enterococci," *New England Journal of Medicine*, vol. 365, no. 10, pp. 892–900, 2011.

[118] D. Wei, S. Mona A, M. Olgica, B. Richard G, *et al.*, "Compressive sensing dna microarrays," *EURASIP journal on bioinformatics and systems biology*, vol. 2009, 2009.