Honesty Is Not Always the Best Policy: Defending Against Membership Inference Attacks on Genomic Data

Rajagopal Venkatesaramani

Follow this and additional works at: https://openscholarship.wustl.edu/eng_etds

Recommended Citation
Venkatesaramani, Rajagopal, "Honesty Is Not Always the Best Policy: Defending Against Membership Inference Attacks on Genomic Data" (2023). McKelvey School of Engineering Theses & Dissertations. 917.
https://openscholarship.wustl.edu/eng_etds/917

This Dissertation is brought to you for free and open access by the McKelvey School of Engineering at Washington University Open Scholarship. It has been accepted for inclusion in McKelvey School of Engineering Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.
Honesty Is Not Always the Best Policy:
Defending Against Membership Inference Attacks on Genomic Data
by
Rajagopal Venkatesaramani

A dissertation presented to
the McKelvey School of Engineering
of Washington University
in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

August 2023
St. Louis, Missouri
Table of Contents

List of Figures ......................................................................................................................... vi
List of Tables ......................................................................................................................... xii
Acknowledgements ................................................................................................................. xiii
Abstract ................................................................................................................................. xvi
Chapter 1: Introduction ............................................................................................................ 1
   1.1 Motivation ................................................................................................................... 1
       1.1.1 Re-identification in Anonymized Genomic Datasets using Phenotypes ....... 2
       1.1.2 Membership Inference Attacks on Genomic Summary Statistics .............. 3
   1.2 Overview of the Thesis ................................................................................................. 4
Chapter 2: Background and Related Work ................................................................................ 6
   2.1 Privacy Preserving Data Sharing ................................................................................. 12

I  Re-identification of Individuals in Genomic Datasets Using Public Face Images 19

Chapter 3: Linking Images to DNA Using Genotype-Phenotype Associations...................... 20
    3.1 SNP-Phenotype Associations .................................................................................... 22
    3.2 Dataset ...................................................................................................................... 22
    3.3 Predicting Phenotypes from Face Images .................................................................. 24
    3.4 Results ....................................................................................................................... 26
       3.4.1 Average Re-identification Risk is Low in Practice ........................................... 27
    3.5 Matching Accuracy on Synthetic Datasets ............................................................... 30
    3.6 Conclusion ................................................................................................................ 30

Chapter 4: Defending Against Image-Phenotype Linkage Using Adversarial Examples ........ 33
    4.1 Solution Approach ..................................................................................................... 34
       4.1.1 Protecting Privacy by Adding Noise to Face Images ........................................ 34
       4.1.2 Training Robust Phenotype Classification Models ......................................... 36
    4.2 Results ....................................................................................................................... 37
II Defending Against Membership-Inference Attacks on Genomic Beacons and Summary Statistics

Chapter 5: Membership-Inference Attacks on Genomic Beacons and Summary Statistics

5.1 Data Representation
5.2 Likelihood Ratio Test (LRT) Statistics
5.3 Threat Models
5.3.1 Choosing the Inference Threshold
5.3.2 Query Process - Beacons
5.4 The Data Custodian’s Options
5.4.1 Injecting Noise
5.4.2 Suppressing SNVs
5.5 Conclusion

Chapter 6: Protecting Privacy in Genomic Data Beacons

6.1 Privacy and Utility Goals
6.2 The Batch Setting
6.2.1 Fixed-Threshold Attacks
6.2.2 Adaptive Attacks
6.3 The Online Setting
6.3.1 A Model of the Online Beacon
6.3.2 Authenticated Access
6.3.3 Unauthenticated Access
6.4 Experiments
6.4.1 Experiment Setup
6.4.2 The Batch Setting
6.4.3 The Online Setting
6.4.4 Allele-Inference Attack
6.4.5 Mimicry Attack
6.5 Conclusion
Appendix B: Supplement for Chapter 7

B.1 SPG Runtime Analysis

B.2 Improving Runtimes for SPG-R

B.3 Additional Results - Fixed Threshold

B.4 Additional Results - Adaptive Threshold

B.5 Additional Results - Bounded Risk

B.6 Additional Results - LD

B.7 Impact of Increasing Population Sizes

B.8 Empirical Runtime - ILP vs SPG-B
List of Figures

Figure 3.1: The VGGFace Architecture................................................................. 26

Figure 3.2: Fig. 1. Effectiveness of matching individuals’ photos to their DNA sequences in OpenSNP. (A) Success rate for top 1 matching for the Real dataset. (B) Success rate for top 5 matching for the Real dataset. (C) Success rate for top 1 matching in the Synthetic-Ideal dataset. (D) ROC curve for 126 individuals. (A) to (C) present matching success results as a function of the population size (the number of individual genomes to match a face image to) for a fixed $k$. ........ 28

Figure 3.3: Matching accuracy with synthetic datasets. (A)-(F): Matching Accuracy with the Ideal (top row) and Realistic (bottom row) synthetic datasets for top-1, top-3 and top-5. The ground truth accuracy is much higher in the ideal scenario - this is to be expected as genomes are explicitly picked to be the most representative of a corresponding image. The accuracy of predicted matches is much lower than the results on the OpenSNP individuals would suggest. We narrow the cause of this down to the challenges in predicting eye color from images. .......... 31

Figure 4.1: Impact of increasing maximum permitted perturbation, $\varepsilon$. As the value of $\varepsilon$ increases above 0.01, the noise added to the image becomes more visible. ....... 35

Figure 4.2: Evaluating small image perturbations as a defense. (A) Effectiveness of perturbations as a defense against re-identification for $k = 1$ (i.e., the attacker considers only the top match). Pixel values are normalized to a $[0, 1]$ interval, and perturbation strengths $\varepsilon$ are with respect to these normalized pixel values. It can be seen that prediction accuracy is near zero at a perturbation strength $\varepsilon \geq 0.01$. Moreover, even for very small amounts of adversarial noise, such as $\varepsilon = 0.001$, matching success is nearly indistinguishable from random matching if we have at least 20 individuals in a consideration pool. (B) Effectiveness of perturbations that only target sex prediction from a face image. The effect of larger perturbations ($\varepsilon \geq 0.01$) is similar to (A). However, smaller perturbations are considerably less effective. (C) Example images on the right (Photo Credits: The CelebA Face Dataset [72]) illustrate the visible effect of introducing small perturbations to images. The perturbations are essentially imperceptible to a human until $\varepsilon > 0.01$, when the effect becomes clearly visible......................... 38

Figure 4.3: Attacking phenotypes other than sex. Top-1 matching performance when attacking phenotypes other than sex independently. While effective relative to clean images, attacking hair color, skin color or eye color alone does not reduce accuracy to below random except for fairly large populations, where matching accuracy is low to begin with................................................................. 39
Figure 4.4: **Top-3 matching accuracy with perturbed images.** Accuracy of Top-3 matching with perturbed facial images, at different strengths of attack, i.e., values of $\epsilon$, for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color and (E) Universal Noise. Similar to the top-1 case, attacking sex is much more effective compared to attacking other phenotypes independently, and directly minimizing overall matching log-likelihood remains highly effective.

Figure 4.5: **Top-5 matching accuracy with perturbed images.** Accuracy of Top-5 matching with perturbed facial images, at different strengths of attack, i.e., values of $\epsilon$, for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color and (E) Universal Noise. Once again, while attacking sex manages to lower accuracy to below random for even small populations, attacking other phenotypes proves relatively ineffective. Yet again, direct minimization of the matching log-likelihood proves highly effective in preserving privacy.

Figure 4.6: **Evaluation of models that are trained to increase robustness to small perturbations through adversarial training when only the top match is considered in re-identification.** (A) Matching accuracy of “robust” models trained by adding perturbations with varying levels of $\epsilon$ when unperturbed face images are used as inputs. Using $\epsilon > 0.01$ causes matching accuracy to be effectively equivalent to random. (B) Matching accuracy of “robust” models trained by adding perturbations with $\epsilon = 0.01$ when input face images are perturbed with varying levels of adversarial noise. Using $\epsilon > 0.01$ is sufficient to cause sub-random matching accuracy. For noise with $\epsilon = 0.01$, matching accuracy degrades from the original, but remains higher than random.

Figure 4.7: **Baseline Performance.** Top-1 Baseline accuracy of matching with adversarially trained classifiers, but clean images for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color. Legends in each figure show the values of $\epsilon$ used for training the network.

Figure 4.8: **Impact of adversarial training.** (A)-(D): Top-1 Matching accuracy with robust classifiers, with images perturbed at $\epsilon$ between 0 and 0.05 for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color. Each classifier was trained with adversarial examples at $\epsilon = 0.01$.

Figure 4.9: **Baseline top-3 accuracy.** Top-3 Baseline accuracy of matching with adversarially trained classifiers, but clean images for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color and (E) Universal Noise. The legend shows the values of $\epsilon$ used for training the network.

Figure 4.10: **Impact of adversarial training on top-3 matching.** (A)-(E): Top-3 Matching accuracy with robust classifiers, with images perturbed at $\epsilon$ between 0 and 0.05 for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color and (E) Universal Noise. Each classifier was trained with adversarial examples at $\epsilon = 0.01$. .........
Figure 4.11: **Impact of attacking the sex-classifier.** Matching Accuracy with PGD-perturbed images targeting the sex-classifier, for the synthetic datasets (Ideal - top row, Realistic - bottom row) for top-1, top-3 and top-5 matching (left to right). Much like the OpenSNP data, fooling the prediction of sex from images proves to be a highly effective defense for the synthetic datasets, although at a higher value of $\epsilon = 0.025$.

Figure 4.12: Protecting privacy by attacking all classifiers in parallel. Matching Accuracy with images perturbed with Universal Noise, for the Ideal (top row) and Realistic (bottom row) synthetic datasets for top-1, top-3 and top-5. Here, the effectiveness of this method relative to simply attacking sex is much more pronounced. For a very reasonable attack $\epsilon$ of 0.01, the accuracy is reduced to near-random, and for attacks that are stronger, accuracy is reduced to zero, even for very small populations.

Figure 4.13: **Baseline performance of adversarially trained sex-classifier.** (A)-(F): Matching accuracy with sex-classifiers adversarially trained at various values of $\epsilon$ but using clean images for the ideal (top row) and realistic (bottom row) synthetic datasets, for top-1, top-3 and top-5. Yet again, we observe that training against a strong enough adversarial noise attack incurs a significant performance penalty on clean images, making retraining detrimental to the malicious actor attempting re-identification.

Figure 4.14: **Impact of adversarially training the sex-classifier.** (A)-(F) Matching accuracy with a sex-classifier adversarially trained at $\epsilon = 0.01$, attacked with adversarial images for various values of $\epsilon$, for the ideal (top row) and realistic (bottom row) synthetic datasets, for top-1, top-3 and top-5. Similar to our results on the OpenSNP data, retraining boosts robustness to attacks of equal or lower strength as the retraining $\epsilon$, but fail to be robust to stronger attacks. However, in contrast to the OpenSNP data, to reduce accuracy below random in this scenario requires an attack with $\epsilon = 0.05$, at which point adversarial noise starts to become visually evident.

Figure 5.1: An overview of the various threat models for Beacon services.

Figure 6.1: Number of SNVs flipped to guarantee privacy in a Beacon compared to baselines ($n = 400$ individuals).

Figure 6.2: Comparing utility and LRT scores in the fixed threshold batch setting with general AAFs.

Figure 6.3: Comparing utility, privacy and LRT scores in the adaptive attack batch setting.

Figure 6.4: Performance in the adaptive attack batch setting.

Figure 6.5: Comparing utility and privacy in the authenticated online setting with fixed-threshold attacks; $\theta = 0$.

Figure 6.6: Comparing utility and privacy in the authenticated online setting with adaptive attacks.
Figure 6.7: ROC curves, authenticated online setting. ................................................... 95
Figure 6.8: ROC curves, authenticated online setting. All methods tuned for similar utility… 96
Figure 6.9: Comparing utility and privacy in the online unauthenticated fixed-threshold attack setting; $\theta = -1000$. ........................................................................................................ 97
Figure 6.10: Comparing utility and privacy in the online unauthenticated adaptive attack setting; $K = 10$.............................................................................................................. 97
Figure 6.11: ROC curves, unauthenticated online setting. .............................................. 98
Figure 6.12: ROC curves, unauthenticated online setting. All methods tuned for similar utility. 98
Figure 6.13: Comparing privacy in the batch setting against the allele-inference attack. ....... 99
Figure 6.14: Comparing privacy in the batch setting against the mimicry attack................ 99
Figure 6.15: Comparing privacy in the authenticated online setting against the mimicry attack.100
Figure 7.1: Utility-privacy plots for the fixed threshold attack model for Beacons, compared to baselines. A) $\theta = -250$, baselines only flip SNVs. Zoomed-in portions shown in the top two subplots. B) $\theta = -250$, baselines only mask SNVs. ........118
Figure 7.2: Utility-privacy plots for the fixed threshold attack model for AAF releases, compared to baselines. A) $\theta = 0, \alpha = 0.5$. B) $\theta = 0, \alpha = 0.9$............................119
Figure 7.3: Utility-privacy plots for the adaptive threshold attack model for Beacons, compared to baselines. A) $K = 10$, baselines only flip SNVs. B) $K = 10$, baselines only mask SNVs..............................................................120
Figure 7.4: Utility-privacy plots for the adaptive threshold attack model for AAF releases, compared to baselines. A) $K = 10, \alpha = 0.75$. B) $K = 10, \alpha = 0.9$.....................121
Figure 7.5: Fixed threshold attack model, where the attacker leverages correlation data, and baselines only flip SNVs. A) $\theta = 1000$, baselines only flip SNVs. B) $\theta = -250$, baselines only mask SNVs. .................................................................122
Figure 7.6: Adaptive threshold attack model, where the attacker leverages correlation data, baselines flip SNVs. $K = 5$, baselines only flip SNVs..............................................123
Figure 8.1: Model architecture for the deep generative neural network trained to infer membership in genomic data releases, with 10000 SNVs and a 50-dimensional auxiliary noise input. ..............................................................................127
Figure 8.2: Comparison of the performance of the neural-network-based attacker (NNA) to the likelihood-ratio-based attacker (LRT) for Beacon services. ......................129
Figure 8.3: Comparison of the performance of the neural-network-based attacker (NNA) to the likelihood-ratio-based attacker (LRT) for AAF releases. .........................129
Figure 8.4: Impact of MIG on NNA and LRT attack models........................................130
Figure 8.5: Impact of randomly flipping and masking SNVs on NNA and LRT attack models.131
Figure 8.6: Impact of adding Laplacian noise to AAFs on NNA and LRT attack models......132
Figure 8.7: Relative performance of NNA and LRT attack models against differentially-private Laplace mechanism. .................................................................132

Figure A.1: **Receiver Operating Characteristic curves for various population sizes.** (A)-(L): Top $k$ entries in the sorted list of DNA sequences per image are predicted to be matches. (M)-(X): Predictions are made independently for each image-DNA pair. .....................................................................................................[150]

Figure A.2: **Using SVMs to fine-tune matching.** (A)-(C): Matching accuracy when SVMs are used on the 126 individuals to fine-tune. From left to right, (A) Top 1, (B) Top 3, (C) Top 5 matching accuracy with SVMs. The input to the SVM for an image-DNA pair is a vector of probabilities of phenotypes, where phenotypes are extracted from the image and conditional probabilities are calculated from the DNA sequence, and the labels are binary indicating a match or otherwise. For each image in the test set, we rank all DNA sequences in order of their distance from the separating hyperplane, from most likely match to least likely match, and consider the top $k$ as true matches. We note that this approach does not significantly improve matching accuracy, which we believe to be the result of low signal-noise ratio. .....................................................................................................[151]

Figure A.3: **Effect of eye-color prediction accuracy.** (A)-(B): Matching accuracies with and without considering the eye-color phenotype, both predicted and ground-truth for (A) Ideal and (B) Realistic synthetic datasets. Notice the significant drop in ground-truth accuracy when eye-color is disregarded entirely. This points to the high importance of eye-color in matching. At the same time, notice that in the ideal dataset, disregarding eye-color entirely produces better matching accuracy than when including it, although it does not make a significant difference to the realistic dataset. This points to the presence of significant noise in our eye-color predictions. (C)-(D): Matching accuracies when the eye-color predictions from images are replaced by ground-truth values, for (C) Ideal and (D) Realistic synthetic datasets. The significant increase in matching accuracy (nearly the upper bound) in both synthetic datasets strongly suggests that we are limited in our matching ability by the poor performance of eye-color prediction. (E)-(G): Top-1, Top-3 and Top-5 matching accuracies with various eye-color prediction techniques. None of them are particularly effective. ............[152]

Figure B.1: Relative performance of SPG-R (binary) and SPG-R (parallel) under the adaptive threshold model............................................................................156

Figure B.2: Runtime comparison between SPG-R variants and baselines. ......................157

Figure B.3: Utility-privacy plots for the fixed threshold attack model for Beacons, compared to baselines...............................................................158

Figure B.4: Fixed threshold attack model when all approaches only flip SNVs...............159

Figure B.5: Fixed threshold attack model when all approaches only mask SNVs. ..........159
Figure B.6: Utility-privacy plots for the adaptive threshold attack model for AAF releases, compared to baselines .................................................................[160]

Figure B.7: Adaptive threshold attack model, where all approaches only flip SNVs ........[161]

Figure B.8: Performance of SPG-R compared to baselines in the fixed threshold setting (bounded risk) \( \theta = 0 \) ..............................................................................[162]

Figure B.9: Performance of SPG-R compared to baselines in the adaptive threshold setting (bounded risk) for \( K = 5 \) and \( K = 10 \) .................................................................[162]

Figure B.10: Adaptive threshold attack model, where the attacker leverages correlation data, and baselines only flip SNVs. The Greedy approach is permitted to flip or mask all SNVs. \( K = 5 \) ....................................................................................[162]

Figure B.11: Relative performance of DP and SPG-B with increasing dataset size (n). Zoomed-in portions are shown on top.................................................................[164]

Figure B.12: Relative performance of DP and SPG-R with increasing dataset size (n) ..........[164]

Figure B.13: Empirical Comparison of running times for ILP and SPG-B (using only SNVs identified by MIG) .................................................................................[165]
### List of Tables

Table 2.1:  An example of a $k$-anonymized table .......................................................... 12
Table 2.2:  An example of a 3-diverse table. ................................................................. 13
Table 3.1:  Phenotypes considered for matching images to DNA, and corresponding SNPs ... 23
Table 5.1:  General Notation ......................................................................................... 55
Acknowledgments

First and foremost, I would like to express my gratitude to my advisor, Dr. Yevgeniy Vorobeychik. The depth of his thoughts and his passion for excellence have been a constant source of inspiration, and I am incredibly lucky to have been on the receiving end for the past six years. I am also immensely grateful for his patience and understanding during the worst of it all. Without his constant guidance and support, none of this would have been possible.

I am deeply grateful to Dr. Brad Malin who practically became my second advisor. It is not often that one gets to work closely with a researcher of his stature, and I am extremely fortunate to have had the opportunity to do so. His knowledge, encouragement, constant feedback and his diligence regarding our writing have gone a very long way in making me the researcher that I am today.

I would like to thank the members of my committee, Dr. Jeremy Buhler, Dr. Netanel Raviv, Dr. Brendan Juba and Dr. Brad Malin for our discussions and their feedback, without which the ideas in this thesis would not have come to fruition.

I am grateful to members of The Center for Genetic Privacy and Identity in Community Settings at Vanderbilt University for their guidance and support. Being part of this group has been a great privilege. This work would also not have been possible without the support of the National Human Genome Research Institute, National Institutes of Health (grant number RM1HG009034) and the National Science Foundation (grant number IIS-1905558).
I would like to thank my colleagues, Zhiyu Wan, Ayan Mukhopadhyay, Jian Lou, Chen Hajaj, Sweta Panda, Liang Tong, Sixie Yu, Kai Zhou, Jinghan Yang, Andrew Estornell, Junlin Wu, Anindya Sarkar, Stylianos Vasileiou, Shayan Monadjemi and Ashwin Kumar for their generous support, and several insightful discussions throughout my Ph.D. journey.

I am also extremely grateful to all my friends whose shoulders I’ve leaned on countless times. Shounak, Chitra, Praneet, Sayali, Saki, Pooja, Shivangi, Ananya, Lucas, Shayan, Ashwin and Rashmi deserve a special shout-out for their never-ending support. While there are far too many more to name, they know full well who they are. I am also immensely grateful to my guru, Dr. Sanjay Subramanian - your music has kept me sane, and I am forever indebted to you.

I am very grateful to my family whose many sacrifices brought me to these shores. I have had the privilege of being inspired by the people in my own home growing up, and that is a fine thing indeed. I would like to especially thank my parents and my sister for their unconditional love and support. Without their efforts, none of this would have ever happened.

Finally, I am grateful to the British Broadcasting Corporation for their fine productions. They brought me much joy in trying times.

Rajagopal Venkatesaramani

Washington University in St. Louis
August 2023
In loving memory of my grandfather.

This thesis is dedicated to all whom I call my family,
and my mentors whose shoulders I stand on.
Over the last decade, there has been a sharp surge in the collection, sharing and use of genomic data in a broad array of settings - including but not limited to direct-to-consumer (D2C) products such as ancestry and genetic testing kits sold by 23andMe and AncestryDNA, clinical applications including personalized medicine, as well as for broader research purposes in repositories like the 1000 Genomes Project. Further, several open-access repositories where consumers may upload their sequenced genomes, such as the Personal Genome Project and OpenSNP, are becoming increasingly popular. Users of said repositories are motivated either by altruism or by a sense of community built around ancestry or rare genetic traits that users may have in common. As a natural consequence, the sharing of such data has been accompanied by acute privacy concerns, both among researchers as well as the general public. Several of these concerns revolve around membership-inference attacks, whereby an adversarial actor attempts to infer a target individual’s presence in a given dataset. Membership-inference attacks pose a significant privacy risk, as genomic datasets often contain potentially sensitive metadata, such as underlying medical conditions which the attacker may now infer about individuals predicted to be present in the dataset.

In this thesis, I formalize membership-inference attacks on genomic data, and present optimization-based techniques to defend against them, with the aim of minimizing the impact on the utility of the data while preserving individual privacy. I present two such membership-inference attack
paradigms. In the first model, an attacker attempts to establish an individual’s presence in an open-access but anonymized dataset such as OpenSNP. In this setting, I assume that the attacker uses information about phenotypes (specifically, facial features) of known target individuals (such as relatives) to predict if a genome in the dataset is a potential match. In particular, I consider the case where the attacker has access to a face photograph for each target individual, and attempts to create a probabilistic matching between each photograph and genomes in the dataset. I make two main contributions: a) I evaluate the risk of such an attack on photographs obtained in the wild, i.e. user-uploaded images, in contrast to photographs produced in controlled lab settings or 3-D face scans used in prior literature, and b) I propose and evaluate the use of gradient-based techniques to add adversarial noise to face images as an effective defense against such attacks.

In the second model, I consider membership-inference attacks on genomic summary statistics, which may take the form of binary allele presence/absence query responses for each position, or their respective alternate allele frequencies. In this model, the attacker has access to a set of target genomes and attempts to infer whether each target genome was likely part of the dataset over which the summary was computed. In this setting, I make four contributions: a) I propose a novel attack model which leverages the distributional separation between likelihood ratio test scores of individuals in a given dataset and a reference population of individuals not in the dataset, b) I present the first formalization of membership-inference attacks in an online query setting, and authenticated and unauthenticated forms of access-control, c) I present highly-scalable greedy approaches to optimizing the privacy-utility tradeoff for genomic summary statistics by inducing binary or real-valued noise to the data release and selectively suppressing a subset of the release, and d) I use generative deep neural networks to construct a more powerful membership-inference model that is fairly robust to data perturbation.
Chapter 1: Introduction

1.1 Motivation

Human genomic sequencing has become increasingly popular, finding use in a wide range of applications over the past several years in clinical, research and commercial settings. Common use-cases include diagnosis of genetic disorders and personalized medicine [1–3], cancer treatment [4, 5], pharmacogenomics [6–8], forensic analysis [9–11], anthropology and population genetics [12, 13]. In addition to medical and academic settings, direct-to-consumer (D2C) genomic testing companies such as 23andMe, AncestryDNA and FamilyTreeDNA attract a combined customer base of tens of millions of individuals. Using user-collected saliva samples obtained through kits mailed directly to consumers, these companies identify specific variations in an individual’s DNA which may be indicative of ethnic and geographic ancestry, predisposition to certain medical conditions, genetic traits such as the ability to taste certain flavors, and carrier status, i.e. whether the potentially asymptomatic individual has genetic markers which could be passed on to their children. In both settings, the widespread collection of genomic data has sparked acute privacy concerns - for instance, information about the increased risk of contracting certain diseases may be misused by insurers or employers, disproportionately affecting certain segments of the population. Privacy concerns surrounding the violation of user consent provided at the time of data collection are also prevalent, such as the case where anonymized genomic data are later de-anonymized by adversarial actors, using externally available information. An adversary who infers an individual’s presence in a genomic data release is subsequently able to infer potentially sensitive information about the individual such as underlying medical conditions, either inferred directly from the (now de-anonymized) genome or the metadata associated with the data release. The work presented in
this thesis is chiefly motivated by such privacy concerns, broadly divided into two categories based on the threat model.

### 1.1.1 Re-Identification in Anonymized Genomic Datasets using Phenotypes

A common way to collect and share genomic data is through *anonymized* repositories, with varying degrees of access control. While the sharing of such data in academic and medical settings is often highly regulated (for example, through the use of data sharing agreements [14]), cloud-based public-domain repositories also exist where individuals may choose to share their sequenced genomic data (as obtained from a D2C company or a medical provider), under the guise of anonymity through the use of pseudonymous usernames. Examples of such repositories include OpenSNP [15] and the Personal Genome Project [16]. Both these repositories crowdsource genomic sequencing data from thousands of individuals, motivated by potential advancements in scientific research as a result of sharing one’s data. Both platforms also promote community-building centered around interacting with other individuals with similar characteristics or conditions and also act as curators of information about variations in the human genome that are linked to specific characteristics.

A natural privacy concern is whether one can, in combination with externally obtained information, discern the identities of individuals who contributed to the dataset. Thus, in the first category of attacks on genomic data, I consider the threat model where an attacker uses information about an individual’s phenotypes (physical features) to make probabilistic claims about whether a set of an individual’s features correspond to a genome in an anonymized dataset. Specifically, I consider facial phenotypes in my work, and attempt to answer the following questions: given a set of anonymized genomes and a set of face photographs of the individuals obtained *in the wild*, is it possible to successfully construct a matching between the two? If so, how can we prevent such a re-identification attack?

Prior work in the field [17–21] demonstrates a reasonable degree of success in doing so, albeit with some caveats - such as the use of high-quality 3-dimensional face maps, which are difficult
to produce outside lab settings - that I address in my work by instead using face photographs of
the kind typically uploaded by users to various social media platforms. Such images vary vastly
in background settings, the alignment of the face with respect to the camera, and the accuracy of
one’s true facial features which may be obscured, be it through the use of technological filters,
artificial hair color, or as a result of poor image or lighting quality for example. To this end,
this thesis evaluates the risk of re-identification in anonymized genomic datasets using publicly
posted, real-world face photographs and proposes defenses using gradient-based techniques to add
adversarial noise to the image that is imperceptible to humans, but sufficient to fool a neural network
into misclassifying phenotypes from face images.

1.1.2 Membership Inference Attacks on Genomic Summary Statistics

In contrast to the previous setting where entire genomic sequences are shared, several data custodians
choose to instead only share summary statistics for a set of individuals. A notable example is the
Beacon service [22], introduced by the Global Alliance for Genomic Health (GA4GH) in 2015. The
Beacon service (or simply, Beacon) responds with yes or no to questions of the form: “Does any
individual in this dataset have allele A at position 1,234,567?” Other methods include releasing a
set of minor allele frequencies in the dataset for each sequenced position on the genome [23].

Such approaches were thought to sufficiently protect privacy, given that no individual-level
information is released. However, despite being restricted to binary presence/absence queries,
both Beacon services and allele frequencies were shown to be susceptible to membership-inference
attacks that leverage likelihood ratio test (LRT) scores [24, 25]. Given a target individual, an attacker
computes the relative log-likelihood that the individual’s genome was in the dataset, compared
to the null hypothesis that it was not. Privacy concerns in this setting are of a similar nature - by
inferring an individual’s participation in say, a research study, the attacker may learn potentially
sensitive information about them.

Common approaches to mitigate membership-inference attacks of this nature typically involve
the addition of noise to the summary release, or selective suppression of a subset thereof. This poses
a unique challenge for data custodians in the form of a privacy-utility tradeoff, where the publication
of true Beacon responses or allele frequencies and the risk of re-identification to individuals in the
data are at odds. Prior approaches are limited in the following ways: either they use the addition
of noise or data suppression to counter membership-inference attacks, but not both; and they do
not provide a way for custodians to explicitly tune for a desired privacy-utility balance. In this
thesis, I contextualize notions of privacy in the context of LRT-based membership-inference attacks,
present novel attack models that are even more powerful than those proposed in prior literature,
formalize the case where queries to the Beacon are made in an online fashion, and finally, present
highly-scalable methods that combine the addition of noise and data suppression, which allow data
custodians to finely tune the privacy-utility balance of a summary data release.

1.2 Overview of the Thesis

With these motivating factors in mind, I make several contributions to defending against membership-
inference attacks on genomic data, organized into the following directions: a) evaluating the
empirical risk of re-identification in genomic datasets using face photos in the wild, b) the use and
evaluation of machine learning techniques to defend against genome-face re-identification attacks,
c) a novel, stronger attack model for genomic summary statistics, d) formalization of query process
and access control in genomic Beacons and e) approaches to balance privacy and utility in genomic
summary data releases.

The remainder of this thesis is organized as follows. Chapter 2 provides a comprehensive
overview of prior literature related to my work. Chapter 3 presents the problem of matching
genomic sequencing data to face photographs in the wild, including preliminaries, implementational
considerations and challenges. Chapter 4 presents and evaluates various defenses against the attack
proposed in Chapter 3 and a discussion of the robustness of deep neural networks to adversarial
examples and the effects of adversarial training. Chapter 5 presents the preliminaries for the second
attack model involving genomic summary statistics, including an overview of existing attack models from prior literature, a stronger novel attack model proposed as part of this thesis and a formalization of the query process (batch queries versus online sequential queries) and access control (authorized access using user accounts versus unauthorized access) in genomic Beacons. Chapter 6 presents greedy solutions to preserving privacy for all individuals in Beacon services, in various settings categorized by query process and access control. Chapter 7 presents highly scalable heuristic approaches to tune the privacy-utility tradeoff for a genomic summary release. Chapter 8 explores the use of generative deep neural networks to construct a membership-inference attack that is more powerful than the state of the art and evaluates them against various defensive mechanisms. Finally, Chapter 9 presents future directions and concludes the thesis.
Chapter 2: Background and Related Work

The notion of privacy in genomic databases is a complex one and has been the subject of constant debate and discussion in social, legal and academic circles. With the rising popularity of genomic testing in both clinical and direct-to-consumer settings, privacy protections are offered through a variety of safeguards. In the United States, the first layer of protection is offered through the Health Insurance Portability and Accountability Act (HIPAA) [26], which lays down laws regarding the sharing of data that involves sensitive patient health information. Introduced in 1996, HIPAA’s protections in the context of genomic data sharing are implemented through two sets of laws, referred to as the Privacy Rule and the Security Rule. The Privacy Rule defines covered entities under HIPAA which include healthcare providers, health insurance companies and clearinghouses that receive and format health information for say, electronic medical records, and sets rules about the use and disclosure of electronic protected health information (PHI). The definition of PHI covers any health information about an individual such as diagnoses, healthcare history and payment records maintained or transmitted by a covered entity. The Security Rule complements the Privacy Rule by requiring covered entities to implement physical, administrative and technical safeguards against unauthorized access of PHI. Similar protections exist in the European Union in the form of the General Data Protection Regulation (GDPR) [27], introduced in 2018. GDPR is a more general personal-data protection law that codifies the consent process, enforces that only necessary information be collected, states the rights individuals have to access and erase their data as well as receive their data in a structured format, and notification policies and penalties in case of a data breach.

In the case of HIPAA, 18 attributes including individuals’ names, addresses (at a finer resolution than the state of residence), dates and months of birthdays and healthcare interactions, social
security numbers, telephone numbers, photographic images, voice recordings etc. are listed as \textit{personally identifying}, and must be completely removed from any data which are to be considered de-identified for sharing purposes. Yet, researchers have shown such protections to be insufficient in several regards. In addition to not providing formal privacy guarantees, a key concern with such redaction methods is that they fail to account for externally available information, which, combined with even a limited data release, might be sufficient to identify an individual in an otherwise de-identified database. Such attributes that are themselves insufficient to identify someone in a database, but combined with external information may do so are referred to as quasi-identifiers. Several successful instances of reidentifying individuals - both in scientific literature and in the wild - call into question the protections thus offered by selective redaction. In 1997, a year after HIPAA was signed into law, Sweeney \cite{28} demonstrated the shortcomings of removing identifying attributes for anonymization - including the successful re-identification of 97\% of the 1997 voter list for Cambridge, MA, using only a combination of full postal codes and birthdates. In 2006, AOL released de-identified information about over 650,000 users’ search queries, each assigned a unique ID. Within a month, a New York Times article \cite{29} was published, revealing that a reporter was successfully able to track down user number 4417749, a certain Mrs. Thelma Arnold, based on her search queries alone, leading to AOL promptly removing the dataset from circulation, the CTO resigning from the company, and a class-action lawsuit being filed against AOL.

About two years later, Netflix released movie selections of 450,000 users under unique pseudonyms, as part of a challenge to improve their recommender system, along with a $1 million prize. Shortly thereafter, Narayanan and Shmatikov \cite{30} successfully de-identified users by combining movie selections with user ratings on IMDb, which were often posted under real names. Much like the AOL case, a class-action lawsuit was filed against Netflix in December 2009. Similarly, in 2014, New York City released logs of over 173 million taxi trips in response to an information request. In order to anonymize the data, city officials used the md5 \cite{31} hashing algorithm on drivers’ hack license numbers and medallion numbers. Vijay Pandurangan, a software engineer, realized that
while md5 was mathematically irreversible in practice, drivers’ hack license numbers and medallion numbers were finite in number and followed a certain pattern in their structure, and was able to compute the corresponding hashes for all possible combinations in under 2 hours, de-anonymizing the entire dataset [32]. In 2016, the Australian government published de-identified medical billing records of approximately 2.9 million individuals. Subsequently, Culnane, Rubinstein and Teague [33] decrypted IDs of healthcare providers by reverse engineering the pseudo-randomization of de-identified, but unique provider IDs, as well as patients by linking parts of each record to other externally known information about the individual. In 2013, Sweeney [34] showed that it was possible to identify 35 out of 81 patients from de-identified health data sold by Washington State using newspaper reports about why they were hospitalized, such as accidents, along with their residential information in the reports. Sweeney, Abu and Winn [35] also identified upto 97% of participants in the Personal Genome Project [16] by combining demographic information from profiles on the platform with public records such as voter lists.

Similar re-identification attacks extend to genomic data as well. In repositories such as OpenSNP, 1000 Genomes and the Personal Genome Project [15, 16, 36], a participant’s genomic sequence is often accessible in the form of single nucleotide polymorphisms (SNPs) or single nucleotide variants (SNVs), which are genetic variations in an individual’s genome at specific positions. A SNP (pronounced “snip”) typically occurs in a small fraction of the population (thereby, we refer to the allele occurring in the majority of individuals as the major allele, and the variant as the minor allele), whereas SNV refers to any single-nucleotide changes, including SNPs (where we pick any allele as the reference allele, and alleles varying from the reference are called alternate alleles). Genotypes are a set of alleles in an individual’s DNA that are responsible for their physical characteristics (called phenotypes).

In 2008, Homer et al. [25] demonstrated that it was possible to predict individual membership in a complex genotype mixture, even when an individual’s DNA accounted for less than 0.1% of the mixture. In 2009, Sankararaman et al. [37] studied the statistical power of membership-
inference attacks on published allele frequencies, using likelihood-ratio test (LRT) scores. These studies sparked several follow-ups in the scientific community, exploring the vulnerabilities of various genomic data-sharing mechanisms. Wang et al. [38] extended the attack of Homer et al. by predicting membership in genome-wide association studies (GWAS) using correlations between a few hundred SNP allele frequencies. Gymrek et al. [39] showed in 2013 that it was possible to identify individuals in genomic databases by inferring their most likely surnames from public genealogy websites by searching for relatives by comparing Y-chromosome sequences and combining them with age and state of residency.

In response, data custodians switched to limiting the information shared through, for example, the use of Beacon services [22] that only offer allele presence or absence responses to user queries, or by only sharing alternate or minor allele frequencies [40, 41]. However, in 2015, Shringarpure and Bustamante [24] demonstrated that Beacon services too were vulnerable to membership-inference attacks that compute a likelihood ratio test (LRT) score as follows: given a Beacon service and a target genome, the attacker computes the test statistic as the ratio of the likelihood that the target individual is in the Beacon to the null hypothesis that the target is not in the Beacon. Further, the attacker has access to the allele frequencies, assumed to be drawn from a Beta distribution and the number of individuals in the Beacon. The proposed attack also accounts for genomic sequencing errors as part of the LRT formulation. The LRT attack was later extended using real allele frequencies by Raisaro et al. [42]. Liu et al. [43] demonstrated the risk of re-identification using phenotypic summary data at scale with a population of 2 million individuals with over 8000 targets in 2018.

These attacks assume that SNVs are independent, however, this is not necessarily the case in practice, and correlations between SNVs may reveal more information than initially apparent. Von Thenen et al. [44] proposed an allele-inference technique that leverages such correlations and allows the adversary to make fewer queries to the model and rely on reconstructed alleles to perform membership inference. This may be a factor when a Beacon or similar service imposes
a budget constraint for users, limiting the maximum number of queries as a privacy-protecting measure. Ayoz et al. [45] proposed a genome-reconstruction attack on dynamic Beacons that evolve with new members being added, where the membership of an individual in a Beacon is easier to infer, but the Beacon carries no sensitive metadata. By reconstructing an individual’s genome from the Beacon’s responses, an attacker may be able to infer potentially sensitive information such as medical conditions. Bu et al. [46] proposed a two-step membership inference pipeline, by first predicting haplotypes from genomic summary data, and then using the predicted haplotypes to perform membership inference.

In 2022, a new specification for genomic Beacons, named Beacon v2, was proposed [47] with a three-tiered user authentication model as a security measure. A user’s access to the Beacon’s data is limited by the tier at which it is accessed - anonymous, registered, or controlled. Data at the anonymous tier of classification may be accessed by anyone, as a guest user. Accessing data at the registered tier requires identification by means of a sign-up process using personal credentials. Accessing data at the controlled tier requires the user to specifically apply, and be granted access to do so. The specification, however, makes no note of the aforementioned membership inference attacks proposed over the years against version 1 of the Beacon. While this class of membership-inference attacks on genomic data primarily concerns data custodians and identifies the shortcomings of legally mandated privacy-protection methods, there exists another class of attacks that has far-reaching implications for social media users who choose to share genomic data. In the set of attacks that I next describe, the risk to individual privacy arises from genomic data being linked to individuals’ physical features - information that is much easier to obtain.

Genome-Wide Association Studies (GWAS) are employed to identify SNPs/SNVs which are highly correlated with phenotypes such as eye color, hair color, etc., medical conditions such as cancer and diabetes, or other biological traits. Once variants correlated with desired traits are identified, further association studies on specific populations are often conducted by empirically measuring correlations. Several studies in recent literature identify genetic factors for various
physical attributes. In 2016, pigmentation phenotypes in German populations were found to be determined by a small number of SNPs by Caliebe et al. [21], along with identifying a SNP highly correlated with blue eye color. Genes defining the face shape of Han Chinese individuals were identified by Qiao et al. [20], using 3D face scans of Uyghurs, an admixed population. The study also proposed a model to predict face shape from genomic data, with applications in forensic scenarios. Similarly, Liu et al. [48] identified loci that influence European face shapes. Genomic samples of twins were used to identify strong hereditary indicators of face shape and eyes, combined with a high-precision 3D camera system for phenotyping.

The existence of associations between genotypes and phenotypes raises the potential for an added risk to privacy, with easily obtainable phenotype information being used to re-identify an individual in a genomic database. Several studies published over the last decade illustrate the risk of re-identifying an individual in an anonymized genomic dataset by constructing a probabilistic matching between a set of physical features and a genomic sequence. Humbert et al. [17] first showed that such vulnerabilities exist in OpenSNP data using user-reported phenotypic information, computing the likelihood of a set of known phenotypes given each genome in the dataset and constructing a matching. The authors showed that it was possible to identify around half of the participants with a dataset size of 10, and around 13% with a dataset size of 80, both of which were well above a random baseline. The study by Lippert et al. [18] sparked widespread concerns [49] after claiming to be able to identify greater than 8 out of 10 held-out individuals, combining whole genome sequencing and high-quality phenotyping using statistical models. The study drew criticism that it overstates the ability to match faces to DNA [50] - simulations were used to show that only 3 traits were sufficient to uniquely identify an individual in a cohort of 10, prompting the authors to issue a response [51] - arguing that the simulations assume knowledge of phenotypes (age, ethnicity and gender), whereas the original study predicts these from genomic samples.
2.1 Privacy Preserving Data Sharing

As various data-sharing mechanisms were found to be vulnerable to inference attacks such as the ones discussed above, the scientific community also proposed several notions of privacy in order to provide formal guarantees against particular classes of attacks. One of the earliest notions of privacy introduced shortly after the signing of HIPAA was that of $k$-anonymity. Proposed in 1998, the key idea behind $k$-anonymity is that a table protected under this guarantee cannot be resolved to fewer than $k$ individual entries by combining any subset of quasi-identifying columns in the table. This is often achieved by decreasing the granularity of various attributes - such as restricting temporal information to the year, obfuscating zipcodes to the first two digits, or removing zipcodes entirely in favor of city or state as an attribute. However, $k$-anonymity falls prey to the same shortcomings as de-identified information by removing named identifiers, where externally available information may be combined with entries in the dataset to correctly identify an individual.

Consider the following example.

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Zipcode</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ava</td>
<td>37</td>
<td>63108</td>
<td>Hypertension</td>
</tr>
<tr>
<td>Brandon</td>
<td>33</td>
<td>63130</td>
<td>Hypertension</td>
</tr>
<tr>
<td>Cameron</td>
<td>32</td>
<td>63130</td>
<td>Hypertension</td>
</tr>
<tr>
<td>David</td>
<td>20</td>
<td>63006</td>
<td>Cancer</td>
</tr>
<tr>
<td>Emily</td>
<td>19</td>
<td>63004</td>
<td>Asthma</td>
</tr>
<tr>
<td>Fiona</td>
<td>21</td>
<td>63008</td>
<td>Diabetes</td>
</tr>
<tr>
<td>Gabriel</td>
<td>58</td>
<td>63130</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>Hannah</td>
<td>52</td>
<td>63108</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>Isaac</td>
<td>48</td>
<td>63108</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

(A) Original table.

<table>
<thead>
<tr>
<th>ID</th>
<th>Age</th>
<th>Zipcode</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3*</td>
<td>631**</td>
<td>Hypertension</td>
</tr>
<tr>
<td>2</td>
<td>3*</td>
<td>631**</td>
<td>Hypertension</td>
</tr>
<tr>
<td>3</td>
<td>3*</td>
<td>631**</td>
<td>Hypertension</td>
</tr>
<tr>
<td>4</td>
<td>≤25</td>
<td>6300*</td>
<td>Cancer</td>
</tr>
<tr>
<td>5</td>
<td>≤25</td>
<td>6300*</td>
<td>Asthma</td>
</tr>
<tr>
<td>6</td>
<td>≤25</td>
<td>6300*</td>
<td>Diabetes</td>
</tr>
<tr>
<td>7</td>
<td>≥45</td>
<td>631**</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>8</td>
<td>≥45</td>
<td>631**</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>9</td>
<td>≥45</td>
<td>631**</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

(B) Anonymized table for $k = 3$.

Table 2.1: An example of a $k$-anonymized table.

While names are entirely suppressed, age and zipcodes are resolved to groups, such that each individual’s quasi-identifiers match at least two others in the table, satisfying 3-anonymity. However, this method is still vulnerable to attacks that leverage external information, as illustrated
by Machanavajjhala et al [52] in 2017. Consider Ava, a 37-year-old living in the Central West End (zipcode 63108). If someone - say, an acquaintance - who knows Ava also discovers that their record is in this dataset, then they can deduce that Ava suffers from hypertension, due to the lack of diversity in the sensitive attribute for IDs 1-3 - in this case, disease. This is an example of a homogeneity attack. For an example of the second type of attack, which the authors call the background-knowledge attack, consider Gabriel, a 58-year-old living in University City (zipcode 63130). If someone knows that Gabriel’s record is in this data release, they can deduce that their record corresponds to ID 7, 8 or 9. If they have further background knowledge about Gabriel - say, that Gabriel has a very low risk for heart disease, given their lifestyle and diet - they can posit that Gabriel has cancer with reasonable confidence.

<table>
<thead>
<tr>
<th>ID</th>
<th>Age</th>
<th>Zipcode</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3*</td>
<td>631**</td>
<td>Hypertension</td>
</tr>
<tr>
<td>2</td>
<td>3*</td>
<td>631**</td>
<td>Asthma</td>
</tr>
<tr>
<td>3</td>
<td>3*</td>
<td>631**</td>
<td>Diabetes</td>
</tr>
<tr>
<td>4</td>
<td>≤25</td>
<td>6300*</td>
<td>Cancer</td>
</tr>
<tr>
<td>5</td>
<td>≤25</td>
<td>6300*</td>
<td>Asthma</td>
</tr>
<tr>
<td>6</td>
<td>≤25</td>
<td>6300*</td>
<td>Diabetes</td>
</tr>
<tr>
<td>7</td>
<td>≥45</td>
<td>631**</td>
<td>Asthma</td>
</tr>
<tr>
<td>8</td>
<td>≥45</td>
<td>631**</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>9</td>
<td>≥45</td>
<td>631**</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

Table 2.2: An example of a 3-diverse table.

As a response to these kinds of attacks, Machanavajjhala and colleagues [52] also presented a new model for privacy called $\ell$-diversity. In a table, an equivalence class is a set of records with identical quasi-identifiers - essentially, these records are indistinguishable from each other. The $\ell$-diversity principle requires that an equivalence class has at least $\ell$ well-represented values in the sensitive attribute. A table is said to be $\ell$-diverse if each equivalence class in the table satisfies $\ell$-diversity. In the above example, the first equivalence class consisting of IDs 1-3 does not satisfy $\ell$-diversity for $\ell > 1$. However, if the disease column had different values for the three individuals, then a homogeneity attack would fail. For a background-knowledge attack to fail, the parameter
\( \ell \) would need to be set to a larger value, such that a process of elimination does not lead to a unique value in the sensitive attribute. Table 2.2, for instance, satisfies 3-diversity. Note that this is not the same table as in the previous examples (values of sensitive attributes changed to illustrate 3-diversity).

Such anonymization is still not infallible to probabilistic inference attacks, the distribution over entries within an equivalence class may still reveal enough information for an adversary to infer a record’s sensitive attribute with high confidence. For example, if the disease column for an equivalence class contains HIV for 80% of the records, then probabilistically, an attacker may conclude that an individual whose record is in that class has HIV with high confidence. Addressing such concerns, the original authors also proposed two other notions of \( \ell \)-diversity. The first of these notions is Entropy \( \ell \)-diversity. Let \( S \) be the domain of a sensitive attribute, let \( E \) be the equivalence class under consideration and \( p(E, s) \) be the fraction of records in \( E \) that have a sensitive value \( s \). The entropy of the equivalence class is defined as:

\[
\text{Entropy}(E) = - \sum_{s \in S} p(E, s) \log p(E, s)
\]

A table has entropy \( \ell \)-diversity if \( \text{Entropy}(E) \geq \log \ell \) \( \forall E \). This, however, may be too restrictive in that entropy \( \ell \)-diversity may be impossible to achieve with overrepresented sensitive attributes. Additionally, some positive disclosure may be tolerable - following the example by the original authors [52], records from a clinic specializing in heart problems would naturally disclose that their patients suffer from that condition, and accounting for such allowable tolerances allows for a less conservative notion of \( \ell \)-diversity, known as Recursive \( \ell \)-diversity. In this model, in each equivalence class \( E \), let \( r_i \) be the number of occurrences of the \( i^{th} \) most frequent sensitive attribute in \( E \). Then given a constant \( c \), \( E \) is recursive \((c, \ell)\)-diverse if \( r_\ell < c(r_\ell + r_{\ell + 1} + \cdots + r_m) \). This version has the advantage of being able to exclude a subset of sensitive attributes for which positive disclosure is acceptable.

However, \( \ell \)-diversity also has certain vulnerabilities. Li et al. [53] present two attacks on \( \ell \)-diversity namely the skewness attack and the similarity attack. In the skewness attack, an attacker
leverages distributional information, such as a table where a large majority of records (say, 99%) have a negative disclosure (no disease), with the remaining minority falling entirely in an equivalence class where the sensitive attribute has an equal number of positive and negative entries. While the equivalence class satisfies both definitions of $\ell$-diversity for $\ell = 2$, the members of the class are inferred to have a disease with a 50% probability. In the similarity attack, an adversary learns information from the semantic similarity between entries in the equivalence class, such as, for example, sensitive attributes in a certain equivalence class are all different heart conditions. While the exact disease may not be able to be inferred, the adversary still knows that a record in this class has a heart-related disease.

As a result, the notion of $t$-closeness was proposed by Li et al. [53], which defines an equivalence class to satisfy $t$-closeness if the distance between the distribution of a sensitive attribute within the class $E$ and in the entire table is no more than a threshold $t$. In this case, the authors recommended the use of the Earth Mover’s Distance (EMD) to measure the distance between the two distributions. For categorical variables, EMD is calculated in terms of the variational distance, which is half the sum of differences between each entry’s frequency in the equivalence class and its frequency in the table [54]. As an example, consider Table 2.2 again, where the first equivalence class (IDs 1-3) contains entries from the set \{Hypertension, Asthma, Diabetes\} in the sensitive attribute, Disease. Hypertension, Asthma and Diabetes each occur with a frequency of 1/3 in the equivalence class. Hypertension occurs with a frequency of 1/9 in the table. Similarly, we can compute occurrence frequencies for each possible entry for the sensitive attribute. Let the (ordered) set of all entries be \{Hypertension, Asthma, Diabetes, Cancer, Heart Disease\}. The EMD for this particular equivalence class, $E_1$ is thus computed as:

$$EMD(E_1) = \frac{1}{2} \left[ \left| \frac{1}{3} - \frac{1}{9} \right| + \left| \frac{1}{3} - \frac{1}{3} \right| + \left| \frac{3}{9} - \frac{2}{9} \right| + \left| \frac{0}{3} - \frac{2}{9} \right| + \left| \frac{0}{3} - \frac{1}{9} \right| \right] \approx 0.333$$

Similarly, we can calculate the EMD for the other two equivalence classes (IDs 4-6, and 7-9 respectively), and we find that $EMD(E_2) \approx 0.222$ and $EMD(E_3) \approx 0.2777$, and thus we say that the table satisfies 0.333-closeness. To define a measure of distance between categorical entries of
an attribute such as disease, one may also rely on taxonomical hierarchies instead, vis a vis their corresponding distance in a knowledge graph, such as the one proposed by Fu et al. [55] in 2023. Despite addressing the shortcomings of $\ell$-diversity and its predecessors, $t$-closeness is also limited to privacy in the context of attribute disclosure and is effective with range or threshold queries. To allow for a more flexible notion of privacy that accounts for arbitrary inference, as well as supports queries such as counts, the idea of differential privacy was introduced in 2006 by Dwork [56, 57]. Under differential privacy, the probability of obtaining an output $s$ from a randomized mechanism $M$ on a dataset $D$ is almost the same as the probability of obtaining the same output on a dataset $D'$ which differs in one record. Formally, this is quantified as follows. Let $M$ be a randomized mechanism, and let $S$ be any subset of the range of $M$. Then, for an exogenously specified privacy parameter $\epsilon$, the mechanism $M$ is said to be $\epsilon$-differentially-private if for two datasets $D$ and $D'$ differing in one record,

$$\frac{\Pr(M(D) \in S)}{\Pr(M(D') \in S)} \leq e^\epsilon$$

In other words, differential privacy aims to guarantee that no individual’s record has a significant impact on the output by itself, which in turn, ensures that the mechanism does not reveal individual-level data. In practice, several randomized noise-addition techniques are used to satisfy differential privacy, such as the perturbation of records using noise sampled from Laplacian or exponential distributions [58] or the so-called staircase mechanism which utilizes a geometric mixture of uniform distributions to produce a symmetric staircase-like distribution [59]. Less conservative variants of differential privacy also exist, such as the notion of $(\epsilon, \delta)$-differential privacy [58], where given two datasets $D$ and $D'$ differing in one record, a mechanism $M$ is said to be $(\epsilon, \delta)$-differentially private if

$$\frac{\Pr(M(D) \in S)}{\Pr(M(D') \in S) + \delta} \leq e^\epsilon$$

The addition of the $\delta$ term allows for a user-specified relaxation of the privacy budget, and setting it to 0 is equivalent to satisfying $\epsilon$-differential privacy. Local differential privacy (LDP) [60] is another variant that has recently gained popularity, applied to settings where data are collected from several individuals, or distributed data-collection centers (these could be various medical institutions, for
instance), and then aggregated by a central entity. Under this framework, differentially-private mechanisms are used to perturb the data at each local site after collection, before being sent to the aggregator to process. Several algorithmic innovations have also been proposed to facilitate LDP at scale. Wang et al. [61] proposed novel LDP mechanisms and extended them to aggregate computations on locally perturbed multidimensional data. Gu et al. [62] addressed the potential differences in privacy requirements at each data collection source for LDP with an input-discriminative extension to LDP.

Despite their popularity, DP-based approaches often require the addition of a prohibitive amount of noise into the data to guarantee privacy, especially in the context of attacks on genomic databases where the number of SNPs/SNVs tends to be in the millions. For instance, a differential-privacy-based approach that leverages a geometric mechanism was adopted by Cho et al. [63] in the context of Beacon services. However, this approach incurs a very large loss in utility as we increase the number of SNPs/SNVs. Takagi et al. [64] proposed P3GM, a differentially-private generative model based on variational autoencoders to overcome the issue of a large amount of noise injected into high-dimensional data by traditional DP techniques. However, such an approach typically requires a large amount of training data which is often not readily available in the case of genomic databases and also does not scale to the order of millions of genomic positions.

Various other defenses have been proposed against attacks on genomic data. These approaches typically rely on some degree of obfuscation through noise injection or suppression of a subset of the data. The efficacy of masking a subset of shared genomic data was studied by Sankararaman et al. [37], relying on an upper bound on the power of a likelihood-ratio test for allele frequency releases. More recently, preserving privacy was translated into a game-theoretic perspective by Wan et al. [65, 66], who proposed a mathematical model that considered how to account for the capabilities of an attacker, and the costs related to re-identification - both in terms of access costs to the attacker, and costs to the defender such as payouts in case of a breach of contract in terms of preserving privacy. Various genomic privacy-preserving methods have been summarized by
Bonomi and colleagues [67]. Specific to Beacon services, Raisaro et al. [42] present a random-flipping heuristic that perturbs unique alleles in the database using a binomial distribution, as well as a query-budget approach for authenticated users. A similar approach that amends rare alleles in online Beacon services was presented by Bu et al. [68], where queries to the Beacon are assumed to be made sequentially. The winning entry by Wan et al. [69] to the 2016 iDash Privacy and Security Workshop [70] challenge defines a differential discriminative power that captures an SNV’s marginal contribution to the LRT score. The approach selected SNVs to flip in decreasing order of the proposed metric, followed by a greedy local search to improve utility.
Part I

Re-identification of Individuals in Genomic Datasets Using Public Face Images
Chapter 3: Linking Images to DNA Using Genotype-Phenotype Associations

As discussed in the last two chapters, individual privacy may be put at risk by the sharing of genomic data in several settings. The focus of this chapter is on open genomic data sharing platforms such as OpenSNP, where individuals upload their genomic information as sequenced by direct-to-consumer firms such as 23andMe. These sequenced genomes are of great use to the medical research community, providing more data for genome-phenome association studies, aiding in early disease diagnoses and personalized medicine. These platforms also have a social aspect to them, with communities built around shared, potentially rare or unique phenotypes, and the curation of information about genetic markers for various conditions. One particularly acute privacy concern raised in recent literature is in the ability to link a genome to the photograph of an individual’s face [18–21]. Specifically, these studies have shown that one can effectively match high-quality three-dimensional face maps of individuals with their associated low-noise sequencing data, leveraging known associations between phenotypes, such as eye color, and genotypes, which, for the purposes of this study, refer to the variations in our genes that impact physical traits. However, for a number of reasons, it is unclear whether these demonstrations translate into practical privacy concerns.

First, the studies to date have relied on high-quality, often proprietary, data that is not publicly available. This is a concern because such high-quality data is in fact, quite difficult to obtain in practice. While many people post images of their faces in public, these are generally two-dimensional, with quality that varies considerably depending on a variety of factors, such as resolution, lighting conditions, camera angle, and background objects. Phenotype-association studies, in contrast, typically use high-resolution 3-dimensional face maps captured by dedicated hardware [18–20]
or photographs captured in laboratory-controlled lighting conditions to ensure minimal impact on visible features such as eye color [21]. From a computer vision perspective, ideal datasets would have subjects directly facing the camera, with a plain background [71]; however, this is rarely the case with images in the wild. In addition, observed phenotypes in real photographs need not match actual phenotypes, thereby making it challenging to correctly infer one’s genotype and vice versa. For example, people may color their hair, or eyes (through contact lenses). Finally, increasing population size poses a considerable challenge to the performance of genome-photograph linkage: given a target individual and a fixed collection of features (the predicted phenotypes in our case), the chances of encountering others who are similar to the target individual in this feature space increase with population size. Another related study by Humbert et al. [17] investigates the re-identification risk of OpenSNP data, but assumes complete knowledge of a collection of phenotypes, including many that are not observable from photographs, such as asthma and lactose intolerance. We consider this approach to be a theoretical upper bound in our study, that is, matching performance when ground-truth phenotypes are known \textit{a priori}, as opposed to when predicted from face images. Given these potential confounders in the real world, we study the risk of re-identification of shared genomic data when it can potentially be linked to publicly posted face images. To this end, we use the OpenSNP [15] database, along with a new dataset of face images collected from an online setting and paired with a select subset of 126 genomes. We develop a re-identification method that integrates deep neural networks for face-to-phenotype prediction (e.g., eye color) with probabilistic information about the relationship between these phenotypes and single nucleotide polymorphisms (SNPs), which are nucleotide variations distributed across the genome, to score potential image-genome matches. The purpose of this part of our study is to assess how significant the average risk is, as a function of population size, given the nature of available data as well as current technology.
3.1 SNP-Phenotype Associations

Recall that a single nucleotide polymorphism (SNP) is a variation on an individual’s genome, where a different allele is present compared to the majority of the population. As discussed in the previous chapter, a number of such SNPs have been found to be correlated to an individual’s physical traits, such as eye color, hair color, etc., - also known as phenotypes. For purposes of our study, we consider four phenotypes: eye color (either blue, brown which includes black as its darkest variant, or an intermediate color such as green), skin color (on a 3-point scale - pale, intermediate or dark), hair color (blonde, brown or black) and the individual’s sex (male or female, not considering individuals with XYY chromosomes). To associate phenotypes predicted from images to genomes, we consider the same set of SNPs as used in the study by Humbert et al. [17]. The SNPs considered in the study are presented in Table 3.1.

We refer to the possible values of each phenotype in Table 3.1 (for example, blonde for hair color) as a variant. Given a genotype (the set of relevant SNPs for the purposes of our discussion), we can compute the corresponding log-likelihood of a specified set of phenotypic variants, which, in turn, can be predicted from face images as we show next.

3.2 Dataset

We investigate the risk of re-identification in genomic datasets “in the wild” based on linkage with publicly posted photos. To produce the dataset used in this study, we first collected genomes uploaded to OpenSNP that were sequenced by 23andMe. We then filtered the users who have self-reported all four phenotypes we are interested in. We used a holistic approach to associate genomes to images as follows. If a user’s picture was posted on OpenSNP, higher-quality pictures could often be found under the same username on a different website. When no picture was posted for a certain user on OpenSNP, we found pictures posted on different websites under the same username, and used self-reported phenotypes on OpenSNP to ensure with a reasonable degree of
certainty that the image corresponds to the genome. This process yielded 126 pairs of face images and DNA profiles of individuals that were carefully curated. This resulted in a dataset of SNPs with the corresponding photos of individuals, which we refer to as the Real dataset.

The full study was approved by the Washington University in St. Louis Institutional Review Board, with the condition that we will not publicly release this curated dataset in order to mitigate any possible harm to these individuals. However, we are able to share it privately with other researchers for validation purposes, subject to an agreement preventing public release.

To characterize the error rate in phenotype prediction from images, we constructed two synthetic datasets, leveraging a subset of the CelebA face image dataset [72], and OpenSNP. In this study, synthetic data refers to image-genome pairs that are generated by combining these two unrelated

<table>
<thead>
<tr>
<th>Phenotype</th>
<th>Traits</th>
<th>SNPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>M F</td>
<td>Sex Chromosome</td>
</tr>
<tr>
<td>Skin Color</td>
<td>Pale, Int, Dark</td>
<td>rs26722, rs1667394, rs16891982</td>
</tr>
<tr>
<td>Hair Color</td>
<td>Blonde, Brown, Black</td>
<td>rs12821256, rs35264875</td>
</tr>
<tr>
<td>Eye Color</td>
<td>Blue, Brown, Int</td>
<td>rs916977, rs1129038, rs1800401, rs2238289, rs2240203, rs3935591, rs4778241, rs7183877, rs8028689, rs12593929, rs1800407, rs7495174</td>
</tr>
</tbody>
</table>

Table 3.1: Phenotypes considered for matching images to DNA, and corresponding SNPs.
datasets, where the genome in a given pair does not correspond to the individual in the image (taken from CelebA), but comes instead from an individual with the same set of phenotypes (taken from OpenSNP). We created artificial genotypes for each image (here, genotype refers only to the small subset of SNPs we are interested in; we refer the reader to Table S1 for the full list) using all available data from OpenSNP where self-reported phenotypes are present. First, we consider an idealized setting where for each individual, we select a genotype from the OpenSNP dataset that corresponds to an individual with the same phenotypes, such that the probability of the selected phenotypes is maximized, given the genotype. In other words, we pick the genotype from the OpenSNP data that is most representative of an individual with a given set of phenotypes. We refer to this dataset as Synthetic-Ideal.

Second, we consider a more realistic scenario where for each individual we select a genotype from the OpenSNP dataset that also corresponds to an individual with the same phenotypes, but this time at random according to the empirical distribution of phenotypes for particular SNPs in our data. Since CelebA does not have labels for all considered phenotypes, 1000 images from this dataset were manually labeled by one of the authors. After cleaning and removing ambiguous cases, the resulting dataset consisted of 456 records. We refer to this dataset as Synthetic-Realistic.

### 3.3 Predicting Phenotypes from Face Images

In computer vision, the use of deep neural networks is common for classification tasks where face images are used as inputs. Examples include face recognition [73] for authentication and identification [74, 75] or for automated photo-labeling such as the model used in Google Photos [76], emotion detection [77] to aid in cognitive science studies [78], eye-tracking [79] and motion tracking [80, 81] for games, virtual reality headsets and other interactive experiences, facial landmark detection [82, 83] for augmented-reality applications such as the virtual trying on of sunglasses or spectacles, and age and gender estimation [84] for targeted advertisement and marketing.

For our study, we use a model trained for face recognition as a starting point and build phenotype
classifiers that learn to extract the four aforementioned features of interest from face images. To accomplish this, we employ transfer learning techniques [85], where a deep neural network model trained for a certain task is used for a different, but related task by modifying the final output layers and re-training on a dataset suited to the new task. Doing so is useful because training data in certain domains may be scarce, but data in a related domain may be plentiful and easy to obtain, and training a model from scratch on a small set of data may lead to significant overfitting or outright poor performance. Transfer learning has been extensively and successfully used for classification tasks where labeled data are scarce [86–93].

Specifically, we started with the VGGFace model by Parkhi et al. [73] previously trained on the CelebA dataset [72] which contains over 200,000 celebrity images and 40 attributes for a face recognition task. The model architecture is presented in Fig. 3.1. We then fine-tuned separate copies of the model on subsets of the CelebA dataset, one for each phenotype of interest. For sex and hair color, the CelebA dataset already contains labels for all ~ 200,000 images, and we thus used the entire dataset to fine-tune the sex prediction classifier. For hair color, we found that fine-tuning on a subset of 10000 images with an equal number of blonde, brown, and black hair color images outperforms a model trained on the entire dataset; we thus used the latter. For skin color, 1000 images were labeled on a 5-point scale by Amazon Mechanical Turk (AMT) workers, and then manually sorted into 3 classes. For eye color prediction, 1000 images were labeled by AMT workers; however, after manual verification of these labels, we retained ~ 850 images, dropping the rest because the eye color was indeterminate.

To match faces to genomic records, we use the following procedure. After learning phenotype classifiers for each visible phenotype, we predict the most likely variant (i.e., the one with the largest predicted probability) for each face image in test data. We then use this prediction for matching the face image to a DNA record as follows. Let an image be denoted by $x_i$ and a genome by $y_j$. We use the following matching score, where phenotype variant $z_{i,p}$ is the most likely predicted variant:

$$ R_{ij} = \prod_{p \in \{\text{sex, hair, skin, eye}\}} P(z_{i,p} | y_j). $$

(3.1)
To ensure numerical stability, we transform it into log space, resulting in

$$p_{ij} = \log p_{ij} = \sum_{p} \log P(z_{i,p}|y_{j}) \quad (3.2)$$

The probability of each phenotype variant given a genome $P(z_{i,p}|y_{j})$ is in turn expressed as a product of probabilities over relevant SNPs as listed in Table 3.1. The probability of a specific phenotypic variant given a certain SNP is calculated empirically from all OpenSNP data for which the corresponding individuals self-reported all four phenotypes considered in the study. There were 367 such individuals, including the 126 individuals from the Real dataset to ensure a sufficient amount of data.

Having calculated the likelihood of a match between image $X_i$ and DNA sequence $y_j$ for all images, for all DNA sequences, we rank the DNA sequences in decreasing order of matching likelihood for each image. The results presented in this chapter correspond to when the correct match resides in the top-scored 1, 3 or 5 entries in this sorted list.

### 3.4 Results

Our approach described above is similar to the one introduced by Humbert et al. [17], but differs in that we predict phenotypes from face images as opposed to assuming complete knowledge of them. Armed with the predicted log-likelihood scores $p_{ij}$ for genotype-image pairs, we select the top-$k$-
scored genotypes for each face image, where $k$ is a tunable parameter that allows a precision-recall trade-off in the matching predictions.

The effectiveness of re-identification is strongly related to both the choice of $k$ above, as well as the size of the population that one is trying to match against. More specifically, as we increase $k$, one would naturally expect recall (and, thus, the number of successful re-identifications) to increase. On the other hand, a larger population raises the difficulty of the task by increasing the likelihood of spurious matches. We therefore evaluate the impact of both of these factors empirically.

### 3.4.1 Average Re-identification Risk is Low in Practice

We evaluate the effectiveness of re-identification attacks using two complementary measures: 1) the fraction of successful matches and 2) the area under the receiver operating characteristic (ROC) curve (AUC). The former enables us to study re-identification success (while focusing on recall) as a function of population size, while the latter paints a more complete picture of the trade-off between true positive and false positive rates. First, we consider the proportion of successful matches as a function of population size, i.e., the number of individuals in the genomic database. To do this, we consider several fixed values of $k$, where a match from a face image $x_i$ to a genome $y_j$ is considered successful if the associated log-likelihood score $p_{ij}$ is among the top $k$ for the image $x_i$.

The results for the *Real* dataset for $k = 1$ and $k = 5$ are shown in Figs. 3.2A and 3.2B respectively. We compare the success of our re-identification approach to two baselines: 1) when matches are made randomly (a lower bound) and 2) when matches use *actual*, rather than predicted, phenotypes (an upper bound). We can see from Fig. 3.2A, that matching success (where we solely take the top-scoring match) is relatively low even for the upper bound, where we actually know the phenotypes (and, consequently, do not need the images). Nevertheless, the top-1 matching success rate is close to the upper bound (which assumes perfect knowledge of phenotypes) and is considerably better than random. As expected [18], prediction efficacy declines as population size grows.
Figure 3.2: Effectiveness of matching individuals’ photos to their DNA sequences in OpenSNP. (A) Success rate for top 1 matching for the Real dataset. (B) Success rate for top 5 matching for the Real dataset. (C) Success rate for top 1 matching in the Synthetic-Ideal dataset. (D) ROC curve for 126 individuals. (A) to (C) present matching success results as a function of the population size (the number of individual genomes to match a face image to) for a fixed $k$. 

AUC : 0.7265
Fig. 3.2C shows that in an idealized setting, re-identification accuracy can be considerably higher; however, effectively predicting eye color is crucial, and this appears to be a major limitation of existing techniques. Fig. 3.2D shows that when we treat matching as a binary prediction problem, the effectiveness is well above that achieved by randomly guessing. Nevertheless, re-identification risk “in the wild” does not appear to be especially high. While we observe success rates as high as 25%, this is only achieved when the genomic dataset is extremely small—on the order of 10 individuals. In contrast, the success rate for top-1 matching drops quickly and is negligible for populations of over 100 individuals. Moreover, it should be kept in mind that this result assumes that we can predict the phenotypes perfectly.

The overall pattern does not substantially change when $k = 5$. However, in this case, the matching success rates naturally increase, approaching 80% for small populations, and slightly below 20% for populations above 100 individuals. In this case, we do observe that our re-identification approach, while significantly better than random, is also considerably below the theoretical upper bound. This suggests that, when more than a single re-identification claim is permitted for each image, the error in phenotype prediction from face images has a greater influence.

Next, we turn our attention to a different manner of evaluating prediction efficacy: the tradeoff between false positive and false negative rates that obtains as we vary $k$. The results, shown in Fig. 3.2D for a population size of 126 individuals, suggest that the overall re-identification method is relatively effective (AUC > 70%) when viewed as a binary predictor (match vs. non-match) for given genome-image pairs, particularly when compared to random matching. ROC curves when thresholding on $k$ for various population sizes are presented in Figs. A.1A-A.1L in Appendix A, while in Figs. A.1M-A.1X in Appendix A we also consider a common alternative where we use a tunable threshold $\theta$ on the predicted log-likelihood to claim when a match occurs.
3.5 Matching Accuracy on Synthetic Datasets

Next, we delve more deeply into the nature of re-identification risk using the larger synthetic datasets. We present the results for Synthetic-Ideal in Fig. 3.2C. Additional results for both the Synthetic-Ideal and Synthetic-Real datasets, when the top 1, 3, and 5 matches are predicted to be true matches, are provided in Figs. 3.3A-3.3F. These results offer two insights. First, if an attacker has access to particularly high-quality data, re-identification risk can be relatively high for small populations. For example, the upper bound is now over 60% in some cases. However, it can also be seen that of the phenotypes we aim to predict, eye color is both the most difficult and highly influential in matching. If we assume that we know this phenotype, and only have to predict the others, re-identification risk is near its upper bound (which assumes that we know the true phenotypes). This is even more striking in the case of the Synthetic-Real data, as shown in Figs. A.3A-A.3D in Appendix A. To determine if this result was an artifact of the specific method we selected for eye color prediction, we considered several alternative methods for eye color prediction [94], ranging from traditional computer vision techniques to deep learning (see Figs. A.3E-A.3G in Appendix A). None of these methods were particularly efficacious.

3.6 Conclusion

Overall, our results suggest that it is indeed sometimes possible to link public face images and public genomic data, but the success rates are well below what prior literature appears to suggest, even in idealized settings. We believe there are several contributing factors behind this observation. First, the quality of face images in the wild is much lower than the high-definition 3D images obtained in highly controlled settings in prior studies [18–21]. Second, there is a relative scarcity of high-quality training and validation data for this particular task. While there are large, well-labeled datasets for face classification [73, 95–99], the data required for re-identification requires paired instances of genomes and images, which is far more challenging to obtain at scale. Third, visible phenotypes
Figure 3.3: **Matching accuracy with synthetic datasets.** (A)-(F): Matching Accuracy with the Ideal (top row) and Realistic (bottom row) synthetic datasets for top-1, top-3 and top-5. The ground truth accuracy is much higher in the ideal scenario - this is to be expected as genomes are explicitly picked to be the most representative of a corresponding image. The accuracy of predicted matches is much lower than the results on the OpenSNP individuals would suggest. We narrow the cause of this down to the challenges in predicting eye color from images.

are influenced by factors other than just the SNPs that are known to have a relationship with them, particularly when you add artificial factors, such as dying one’s hair or wearing tinted contact lenses, which introduce considerable noise in the matching. Finally, our analysis assumed (as did all prior studies) that we already know that there is, in fact, a match in the genomic dataset corresponding to each face. In reality, success rates would be even lower, since a malicious actor is unlikely to be certain about this [100].

Our findings suggest that the concerns about privacy risks to shared genomic data stemming from the attacks matching genomes to publicly published face photographs are low. Of course,
our results do not imply that shared genomic data is free of concern. There are certainly other potential privacy risks, such as membership attacks on genomic summary statistics [24, 42, 43, 65, 69, 101–104], which would allow the recipient of the data to determine the presence of an individual in the underlying dataset. This type of attack is of concern because it would allow an attacker to associate the targeted individual with the semantics inherent in the dataset. For instance, if the dataset was solely composed of individuals diagnosed with a sexually transmitted disease, then membership detection would permit the attacker to learn that the target had the disease in question. Moreover, we emphasize that our results are based on current technology; it is possible that improvements in either the quality of data, such as broad availability of high-definition 3D photography, or the quality of AI, such as highly effective approaches for inferring eye color from images, will indeed significantly elevate risks of re-identification.
Chapter 4: Defending Against Image-Phenotype Linkage Using Adversarial Examples

In the previous chapter, we have seen how publicly shared genotype data may be probabilistically linked to face images obtained in the wild. Despite the several challenges posed by the quality of such images, we observed that a certain degree of success in linkage can be achieved. While our previous assessment suggests that the re-identification risk to an average individual is likely lower than what has been suggested in prior literature, it is nevertheless evident that some individuals are at risk. Moreover, if the attacker has sufficient prior knowledge to narrow down the size of the population to which a face can be matched, our results do show that even on average the re-identification risk becomes non-negligible. This led us to investigate the natural question: how can we most effectively mitigate re-identification risks associated with the joint public release of both genomic data and face images? Our specific goal is to provide tools that can reduce re-identification risks to individuals who publicly post their photos. Such tools can be used either directly by individuals to manage their own risks or by platforms where photos are posted to manage risks to their subscribers. In particular, we show that this can be accomplished by adding small image perturbations to reduce the effectiveness of genomic linkage attacks.

Our purpose in this chapter is to introduce a practical tool to manage individual risk that enables either those who post face images online or social media platforms that manage this data, to trade off risk and utility from posted images according to their preferences. We emphasize that in the threat model invoked in this research, we consider only face images posted by individual users on social media. In this respect, the utility is implicit, as it is natural to assume that when a user uploads a photograph to a website, they would prefer to retain as much of the original detail as possible, and visible distortions would be unwelcome in such a setting.
Our primary contribution is a method for adding small perturbations to face images prior to posting them that aims to minimize the likelihood of the correct match (that is, to minimize individual risk). This framework is tunable in the sense that the user can specify the amount of noise they can tolerate, with greater noise added having a greater deleterious effect on re-identification success. We show that even using imperceptible noise we can often successfully reduce privacy risk, even if we specifically train deep neural networks to be robust to such noise. Furthermore, adding noise that is mildly perceptible further reduces the success rate of re-identification to no better than random guessing. We note that our privacy model here differs from common conventional models, such as $k$-anonymity [28, 105–108] and $l$-diversity [52] discussed in Chapter 2. Rather, our privacy assessment is closely tied to our risk analysis framework that combines phenotype inference from face images using machine learning with the particular approach to quantifying re-identification risk that we describe.

### 4.1 Solution Approach

#### 4.1.1 Protecting Privacy by Adding Noise to Face Images

Our approach is closely related to adversarial perturbations in computer vision [109, 110]. The idea behind adversarial perturbations is to inject a small amount of noise into an image in order to cause a misprediction, where “small” is quantified using an $l_p$ norm with the maximum allowable perturbation controlled by a parameter $\varepsilon$ ranging from 0 to 1. Examples of the visual impact of increasing $\varepsilon$ are shown in Fig. 4.1.

In our case, however, we do not have a single deep neural network making predictions, but rather a collection of independent phenotype predictors *using the same image as input*. One direct application of adversarial perturbations in our setting would be to arbitrarily choose a phenotype (say, sex, which is the most informative) and target this phenotype for misprediction, with the anticipation that this would cause re-identification to fail. However, we can actually do much better
Figure 4.1: **Impact of increasing maximum permitted perturbation, \( \epsilon \).** As the value of \( \epsilon \) increases above 0.01, the noise added to the image becomes more visible.

at protecting privacy by tailoring perturbations to our specific task. This is because our ultimate goal is not to cause mispredictions of phenotypes per se, but rather to cause an attacker to fail to link the image with the correct genome. Throughout the chapter, we refer to adversarial perturbations thus tailored to our task by attacking all phenotype-classifiers at once as *Universal Noise*, as opposed to attacking any one phenotype classifier in isolation. Recall that the matching score is given by:

\[
p_{ij} = \log R_{ij} = \sum_{p} \log P(z_i, p | y_j)
\]  

Specifically, we leverage the scoring function in Eq. 4.1 to minimize the score \( p_{ij} \) for image \( x_i \) and the correct corresponding genome \( y_j \). However, this is a non-trivial task, since the score function has a discontinuous dependence on phenotype predictions. Thus, we augment the score with the log of the predicted phenotype probabilities. Specifically, let \( g_p(v_p, x_i) \) denote the probability that the neural network predicts a variant \( v_p \) for a phenotype \( p \) (e.g., eye color) given the input face image \( x_i \); these are just the outputs of the corresponding softmax layers of the neural networks. The problem we aim to solve is to find a small (at most \( \epsilon \) in the \( l_\infty \) norm) perturbation to the input image
\[\delta^* \text{ that solves the following problem:}\]
\[
\min_{\frac{-\epsilon}{\epsilon}} \sum_{p \in \{\text{sex, hair, skin, eye}\}} \sum_{v_p} \log g_p(v_p, x_i + \delta) \log P(v_p|y_j) 
\]

(4.2)

We use projected gradient descent (PGD) to solve this problem, invoking a combination of automated differentiation in pytorch [111], and the Adam optimizer [112]. After each gradient descent step, we simply clip the noise to be in the \([-\epsilon, \epsilon]\) range, and also clip pixel values to ensure that these remain within the valid range. We use the original image as the starting point of this procedure (i.e., initializing \(\delta = 0\)).

Recall that \(g_p(v_p, x_i)\) denotes the probability that a phenotype variant \(v_p\) is predicted from image \(x_i\). Slightly abusing notation, let \(g_p(x_i)\) be the probability distribution over variants given an input \(x_i\). Let \(L(g_p(x_i), y_p)\) be the loss associated with the true variant \(y_p\) and predicted variant distribution \(g_p(x_i)\). The goal of the PGD approach for generating adversarial noise is to maximize loss:

\[
\max_{-\epsilon \leq \delta \leq \epsilon} L(g_p(x_i + \delta), y_p). 
\]

We can do this by using a form of gradient descent. Specifically, let \(\delta_k\) be the value of noise in iteration \(k\) (starting with \(\delta_0 = 0\), or a small random noise). Then

\[
\delta_{k+1} = \delta_k + \alpha \text{sgn} \nabla L(g_p(x_i + \delta_k), y_p),
\]

where \(\alpha\) is the learning rate. This process is run for a fixed number of iterations, or until convergence.

4.1.2 Training Robust Phenotype Classification Models

While the idea of adding small noise as a privacy-protection mechanism works well when we use regularly trained phenotype prediction models, one can make such models more robust, albeit at a cost to accuracy on noise-free data. As such, we investigate how effective adversarial training [110, 113], a state-of-the-art approach for making predictions robust to adversarial noise, is at overcoming our noise injection approach.
The main premise of adversarial training is as follows. The broad goal is to solve the following optimization problem:

$$\min_{\theta} \sum_{x, y \in D} L(\theta, x + \delta^*(\theta), y),$$

where $\delta^*(\theta)$ is adversarially induced noise aimed at the model with parameters $\theta$, and $L(\cdot)$ is the loss function. In practice, computing an optimal noise to add is difficult, and we instead use the approaches such as projected gradient descent described above. However, adversarial training proceeds like regular training (using gradient descent), except that each training input is $x + \delta^*(\theta)$ rather than $x$, that is, we use perturbed inputs in place of regular inputs. In generating these instances, we use random starting points for generating such perturbations.

The adversarial training approach improves robustness by essentially augmenting each iteration with training images that actually embed the small perturbations we have designed, using the model from the previous iteration as a reference for devising the perturbations. The downside of adversarial training, however, is that robustness to adversarial inputs often comes at the cost of accuracy on unaltered inputs, and a careful balance must be achieved between adversarial robustness and baseline accuracy.

### 4.2 Results

#### 4.2.1 Achieving Privacy through Small Image Perturbations

Our first evaluation, shown in Fig. 4.2, presents the effectiveness of our method for preserving privacy in public face images. Fig. 4.2A, in particular, demonstrates that when we take deep neural networks for phenotype prediction as given, the effectiveness of the re-identification attack described above declines significantly even for very small levels of noise added to images. For sufficiently large noise (e.g., $\epsilon = 0.01$), the success rate is close to zero, which is considerably lower than random matching. Moreover, by comparing Fig. 4.2A to Fig. 4.2B, it can be seen that our approach is also more effective than designing small perturbations that target a single sex phenotype.
Figure 4.2: Evaluating small image perturbations as a defense. (A) Effectiveness of perturbations as a defense against re-identification for $k = 1$ (i.e., the attacker considers only the top match). Pixel values are normalized to a $[0, 1]$ interval, and perturbation strengths $\epsilon$ are with respect to these normalized pixel values. It can be seen that prediction accuracy is near zero at a perturbation strength $\epsilon \geq 0.01$. Moreover, even for very small amounts of adversarial noise, such as $\epsilon = 0.001$, matching success is nearly indistinguishable from random matching if we have at least 20 individuals in a consideration pool. (B) Effectiveness of perturbations that only target sex prediction from a face image. The effect of larger perturbations ($\epsilon \geq 0.01$) is similar to (A). However, smaller perturbations are considerably less effective. (C) Example images on the right (Photo Credits: The CelebA Face Dataset [72]) illustrate the visible effect of introducing small perturbations to images. The perturbations are essentially imperceptible to a human until $\epsilon > 0.01$, when the effect becomes clearly visible.
Figure 4.3: **Attacking phenotypes other than sex.** Top-1 matching performance when attacking phenotypes other than sex independently. While effective relative to clean images, attacking hair color, skin color or eye color alone does not reduce accuracy to below random except for fairly large populations, where matching accuracy is low to begin with.

The effectiveness of targeting other phenotypes is provided in Fig. 4.3, where it can be seen that perturbations that target only hair color, eye color or skin color predictions are insufficient to induce a significant level of re-identification risk reduction. While the presented results in Fig. 4.3 are only for $k = 1$ (i.e., the attacker only considers the top-scoring match), results for $k = 3$ and $k = 5$ offer similar qualitative insights (as shown in Fig. 4.4 and Fig. 4.5).

The visual effect of the designed image perturbations is illustrated on the right in Fig. 4.2 using images drawn from the public celebrity face image dataset. As can be seen, most of the levels of added noise have negligible visual impact. It is only when we add noise at $\varepsilon = 0.025$ that we begin to clearly discern the perturbations. However, it appears that perturbations of magnitude no greater than $\varepsilon = 0.01$ are sufficient to achieve a high degree of privacy, with success rates of re-identification attacks nearing zero.

### 4.2.2 Evaluating Adversarial Training

Given the relatively limited effectiveness of the re-identification approach in the previous chapter and all of the practical limitations of that exercise, we investigate whether, in practice, adversarial training helps the attacker deflect the small perturbations we introduce to protect privacy. To
Figure 4.4: **Top-3 matching accuracy with perturbed images.** Accuracy of Top-3 matching with perturbed facial images, at different strengths of attack, i.e., values of \( \varepsilon \), for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color and (E) *Universal Noise*. Similar to the top-1 case, attacking sex is much more effective compared to attacking other phenotypes independently, and directly minimizing overall matching log-likelihood remains highly effective.

evaluate the effect of adversarial training on re-identification, we run further training iterations on the phenotype-prediction models with small perturbations generated over subsets of the original training sets. We make five passes over the perturbed data, each time using the model from the previous iteration to generate these small perturbations to the images. Since the matching score depends on images as well as corresponding genomes, we use paired genome-image datasets for adversarial training (the most optimistic setting from the re-identification perspective). Specifically, we use a random subset of 77 image-DNA pairs (approximately 60%) from our OpenSNP dataset for training, and the remaining 49 for testing the matching accuracy. We construct five sets of adversarially robust phenotype prediction models using this procedure, each set adversarially trained using a different amount of added adversarial noise, from \( \varepsilon = 0.001 \) to \( \varepsilon = 0.05 \).
Figure 4.5: **Top-5 matching accuracy with perturbed images.** Accuracy of Top-5 matching with perturbed facial images, at different strengths of attack, i.e., values of \( \varepsilon \), for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color and (E) Universal Noise. Once again, while attacking sex manages to lower accuracy to below random for even small populations, attacking other phenotypes proves relatively ineffective. Yet again, direct minimization of the matching log-likelihood proves highly effective in preserving privacy.

Fig. 4.6A illustrates that baseline prediction accuracy (i.e., using original face images without perturbations) declines as the strength of the perturbation used for adversarial training increases. Indeed, once \( \varepsilon > 0.01 \), the effectiveness of matching is essentially equivalent to random, suggesting that the most robust model that holds any utility is the one with \( \varepsilon = 0.01 \).

Next, we consider how robust this model is to images that now include small perturbations of varying magnitudes generated using the procedure we describe above. The results are shown in Fig. 4.6B. Notably, noise with \( \varepsilon = 0.025 \) again yields near-zero matching success, *even for robust models*. A smaller amount of noise that preserves imperceptibility, such as \( \varepsilon = 0.01 \), is still effective at reducing the accuracy of the robust model, although the resulting re-identification success rate is
now above random matching. Nevertheless, re-identification success in this case is ~ 20% even in the most optimistic case. Moreover, our framework is sufficiently flexible that a particularly risk-averse individual may simply raise the noise level to 0.025, accepting some visible corruption to the image, but effectively eliminating privacy risk.

Fig. 4.7 and Fig. 4.9 present top-1 and top-3 matching results respectively for various values of $\epsilon$ at which the model was adversarially trained, but using clean images. In all cases, we observe a performance decrease, compared to the original models’ performance on the same clean images. We observe that performance degrades to the point where adversarial training may be more detrimental than beneficial to a malicious actor attempting re-identification on a genomic dataset, especially in the case of sex classification, where training against sufficiently strong adversarial noise ($\epsilon = 0.025$) reduces accuracy to below random guessing. In the case of the Universal Noise approach, while accuracy remains above random, note that retraining was only run for a very small number of data points, due to the lack of paired image-DNA data, which is not required when individually retraining phenotype classifiers against PGD.

Fig. 4.8 shows top-1 matching results for phenotype classifiers adversarially trained at $\epsilon = 0.01$ and attacked at various values of $\epsilon$ ranging from 0.001 to 0.05. Similarly, Fig. 4.10 show top-3 matching results. In the case of sex and skin color, adversarial training boosts robustness to perturbed images as expected, it makes no difference or even degrades performance slightly for eye color and hair color prediction. The slight loss in performance on adversarial examples is somewhat unusual, and most likely due to limited training data. Adversarial training against the Universal Noise approach boosts accuracy for small populations but quickly falls to zero as the population size approaches 50.

4.3 **Evaluations on the synthetic datasets**

Finally, we run similar evaluations on the two synthetic datasets introduced in Chapter 3, beginning with the impact of adversarial examples on matching accuracy. Fig. 4.11 presents results when
Figure 4.6: Evaluation of models that are trained to increase robustness to small perturbations through adversarial training when only the top match is considered in re-identification. (A) Matching accuracy of “robust” models trained by adding perturbations with varying levels of $\epsilon$ when unperturbed face images are used as inputs. Using $\epsilon > 0.01$ causes matching accuracy to be effectively equivalent to random. (B) Matching accuracy of “robust” models trained by adding perturbations with $\epsilon = 0.01$ when input face images are perturbed with varying levels of adversarial noise. Using $\epsilon > 0.01$ is sufficient to cause sub-random matching accuracy. For noise with $\epsilon = 0.01$, matching accuracy degrades from the original, but remains higher than random.

attacking only the sex-classifier, and Fig. 4.12 presents results when attacking all classifiers at once in the Universal Noise setting. Fig. 4.13 shows the baseline performance of an adversarially trained sex-classifier when using clean images, and Fig. 4.14 shows the performance of an adversarially trained sex-classifier, but using perturbed images. Note that we refrain from evaluating robustness for other phenotypes as they proved to be largely ineffective on the OpenSNP data.

### 4.4 Conclusion

Our findings in this chapter suggest that it is relatively easy to defend against re-identification attacks that attempt to link face photographs in the wild to genomic sequences publicly shared on platforms such as OpenSNP. However, we reiterate the scope of our results that we discussed in
Figure 4.7: **Baseline Performance.** Top-1 Baseline accuracy of matching with adversarially trained classifiers, but clean images for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color. Legends in each figure show the values of $\epsilon$ used for training the network.

Chapter 3, which is to say that our findings reflect the limitations of the data at hand. More advanced attacks that leverage other externally available information (say, a dataset of government ID photos, which are far more standardized in their composition) may be designed that our method does not account for. However, through several studies which include synthetic variants controlling for the quality of data, as well as evaluations that assume we can infer observable face phenotypes with perfect accuracy, we show that even with advances in technology the risk is likely to remain limited.
Figure 4.8: **Impact of adversarial training.** (A)-(D): Top-1 Matching accuracy with robust classifiers, with images perturbed at $\epsilon$ between 0 and 0.05 for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color. Each classifier was trained with adversarial examples at $\epsilon = 0.01$. 
Figure 4.9: **Baseline top-3 accuracy.** Top-3 Baseline accuracy of matching with adversarially trained classifiers, but clean images for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color and (E) Universal Noise. The legend shows the values of $\varepsilon$ used for training the network.
Figure 4.10: **Impact of adversarial training on top-3 matching.** (A)-(E): Top-3 Matching accuracy with robust classifiers, with images perturbed at $\varepsilon$ between 0 and 0.05 for (A) Sex, (B) Skin Color, (C) Eye Color, (D) Hair Color and (E) Universal Noise. Each classifier was trained with adversarial examples at $\varepsilon = 0.01$. 
Figure 4.11: **Impact of attacking the sex-classifier.** Matching Accuracy with PGD-perturbed images targeting the sex-classifier, for the synthetic datasets (Ideal - top row, Realistic - bottom row) for top-1, top-3 and top-5 matching (left to right). Much like the OpenSNP data, fooling the prediction of sex from images proves to be a highly effective defense for the synthetic datasets, although at a higher value of $\varepsilon = 0.025$. 
Figure 4.12: Protecting privacy by attacking all classifiers in parallel. Matching Accuracy with images perturbed with Universal Noise, for the Ideal (top row) and Realistic (bottom row) synthetic datasets for top-1, top-3 and top-5. Here, the effectiveness of this method relative to simply attacking sex is much more pronounced. For a very reasonable attack $\epsilon$ of 0.01, the accuracy is reduced to near-random, and for attacks that are stronger, accuracy is reduced to zero, even for very small populations.
Figure 4.13: Baseline performance of adversarially trained sex-classifier. (A)-(F): Matching accuracy with sex-classifiers adversarially trained at various values of $\varepsilon$ but using clean images for the ideal (top row) and realistic (bottom row) synthetic datasets, for top-1, top-3 and top-5. Yet again, we observe that training against a strong enough adversarial noise attack incurs a significant performance penalty on clean images, making retraining detrimental to the malicious actor attempting re-identification.
Figure 4.14: **Impact of adversarially training the sex-classifier.** (A)-(F) Matching accuracy with a sex-classifier adversarially trained at $\varepsilon = 0.01$, attacked with adversarial images for various values of $\varepsilon$, for the ideal (top row) and realistic (bottom row) synthetic datasets, for top-1, top-3 and top-5. Similar to our results on the OpenSNP data, retraining boosts robustness to attacks of equal or lower strength as the retraining $\varepsilon$, but fail to be robust to stronger attacks. However, in contrast to the OpenSNP data, to reduce accuracy below random in this scenario requires an attack with $\varepsilon = 0.05$, at which point adversarial noise starts to become visually evident.
Part II

Defending Against Membership-Inference Attacks on Genomic Beacons and Summary Statistics
Chapter 5: Membership-Inference Attacks on Genomic Beacons and Summary Statistics

In the last chapter, we considered the setting where genomic data that is stripped of personally identifying attributes such as name, etc. are published in their entirety, and an attacker attempts to link publicly posted face images to these genomic sequences by leveraging externally available information about genotype-phenotype associations. In this part of the thesis, we consider a different scenario: one where the genomic data itself is considered too sensitive to publish as is, and instead a data custodian resorts to publishing summary information alone. Depending on the institution or use case, this may take many forms, but in this thesis, we deal with two particular instantiations: a) Beacon services, and b) publishing alternate allele frequencies (AAF).

As introduced in Chapter 1, Beacon services (or simply, Beacons) are web services that expose the presence or absence of certain alleles in a genomic dataset. For instance, given a dataset of some $n$ individuals, a Beacon may respond yes or no to whether any individual in this dataset has a particular allele (say G) at a specified position (say position 1,234,567 on chromosome 10). In this chapter, we also categorize Beacons by query process and access control and study the salient differences in how privacy is at risk in each case. In the second setting considered in this thesis, the data custodian chooses to release alternate allele frequencies for each desired position on the genome, which we take to mean the fraction of individuals in the dataset who possess the alternate allele for each position. Recall that for bi-allelic SNVs, any one of the two possible alleles may be defined as the reference allele and the other allele is called the alternate allele.

While exposing such limited information may appear safe, it has been shown to be vulnerable to membership-inference (MI) attacks because it allows users to issue queries for every region of the genome [24, 69]. These attacks assume that the attacker knows the genome of the target and
leverage a statistical test, often in the form of a likelihood ratio, that couples this information with the Beacon response to a collection of queries or AAFs to determine whether the target is a member of the original dataset. The resulting membership inference can, in turn, reveal sensitive information about the individual, such as their health status, that membership in this dataset entails, or simply be in violation of the privacy promises made to the dataset constituents when the data was collected.

In this chapter, we present attacks from prior literature that leverage likelihood ratio tests (LRT) and then propose a novel, more powerful attacker model which leverages an external dataset of individuals not part of the original data release to try and account for changes made to the data release using defensive measures the data custodian may employ. In the case of Beacon services, we also formalize such attacks for different modes of querying the beacon - namely a batch query process where all genomic positions are queried at once and all the information returned is assumed to be used in the attack process, and an online setting where a user queries the Beacon one SNV at a time. We also make a distinction between authenticated and unauthenticated access for online Beacons, such that in the authenticated model, the defender has access to each user’s query history whereas, in the unauthenticated model, users may query the Beacon as a guest.

5.1 Data Representation

A single nucleotide variant (SNV) is a position on the genome where the allele present differs across the population. The fraction of individuals with the alternate allele in a dataset $D$ of $n$ individuals is referred to as the alternate allele frequency (AAF), which we denote by $p_j$ for the $j^{th}$ SNV. The frequency of the alternate allele in a reference population of individuals not in the dataset (we call this $\bar{D}$) is denoted $\bar{p}_j$. An SNV for each individual $i$ actually contains two alleles, one from each parent but here, we are only concerned with whether or not an individual has an alternate allele, and not whether both alleles at the chosen position are the alternate allele. Therefore, the binary variable $d_{ij}$ denotes whether individual $i$ has at least one alternate allele ($d_{ij} = 1$) at position $j$, and $d_{ij} = 0$ otherwise. The total number of SNVs in the dataset is denoted $m$. Let $Q$ refer to a set of $m$ SNV
positions that can be queried. Let $\gamma$ be the genomic sequencing error rate (usually on the order of $10^{-6}$). The probability that no individual in $D$ has an alternate allele (equivalently, all individuals have two reference alleles) at position $j$ is given by $R_n^j = (1 - \bar{p}_j)^{2n}$. Let the summary release be represented by the vector $x$. In the case of Beacons, $x$ is binary, with $x_j = 1$ if $\exists i \in D, d_{ij} = 1$ and $x_j = 0$ otherwise - indicating the presence or absence of an alternate allele in the dataset. In the case of summary statistics, the release is a vector of AAFs, therefore, $x_j \in [0, 1]$.

Table 5.1: General Notation

| $m$ | The number of SNVs. |
| $n$ | The number of individuals. |
| $\gamma$ | The genomic sequencing error. |
| $D$ | The dataset of individuals for whom summary information is released. |
| $\bar{D}$ | A set of individuals not in the dataset $D$. |
| $\bar{D}^{(K)}$ | The set of individuals in $\bar{D}$ with LRT scores in the lowest $K$ percentile. |
| $p_j$ | Alternate allele frequency (AAF) for SNV $j$ for individuals in dataset $D$. |
| $\bar{p}_j$ | Alternate allele frequency (AAF) for SNV $j$ for individuals in reference set $\bar{D}$. |
| $d_{ij}$ | Binary indicator for whether individual $i$ has an alternate allele for SNV $j$. |
| $R_n^j$ | Probability that no individual in $D$ has alternate allele for SNV $j$. |
| $x_j$ | Summary release for SNV $j$ - binary for Beacons, real-valued for AAF. |
| $\delta$ | Noise added to summary release - binary for Beacons, real-valued for AAF. |
| $Q$ | Set of SNVs queried. |
| $M$ | Set of SNVs masked. |
| $L$ | Likelihood Ratio Test (LRT) score. |
| $\theta$ | LRT threshold used by the attacker. |
| $K$ | User-specified percentile of individuals in $\bar{D}$ with the lowest LRT scores. |
| $Z$ | Set of individuals in $D$ for whom privacy is preserved. |
| $\alpha$ | User specified relative cost of flipping to masking. |
| $w$ | User specified relative weight of privacy versus utility. |
5.2 Likelihood Ratio Test (LRT) Statistics

The two membership inference attacks considered in this work are based on likelihood-ratio test (LRT) statistics. These statistics represent the relative likelihood of an individual $i$ being in the dataset $D$ upon which the summary release was computed, to the likelihood that $i$ is in a reference population. An attacker is assumed to have a set of target genomes, for which membership inference is carried out using released summary information, namely Beacon responses or alternate allele frequencies. In the case of Beacon responses, we use the LRT statistic proposed by Raisaro et al. [42], which in turn extends the attack originally proposed by Shringarpure and Bustamante [24]. The original attack assumed AAFs to be drawn from the Beta distribution, whereas the extended version uses real AAFs instead. The statistic is computed as follows: let $T$ be the set of target individuals. Then, given the vector $x$ of Beacon responses to queries $Q$, the likelihood-ratio test (LRT) score for individual $i \in T$ is:

$$L(Q, d_i, x) = \sum_{j \in Q} d_{ij} \left( x_j \log \frac{1 - R^j_n}{1 - \gamma R^j_{n-1}} + (1 - x_j) \log \frac{R^j_n}{\gamma R^j_{n-1}} \right)$$  \hspace{1cm} (5.1)

The LRT statistic for alternate allele frequencies is calculated in a similar fashion and was proposed by [37]. Suppose that we have chosen to release AAFs (real-valued vector $x$) for set $Q$ of SNVs. An attacker who is in possession of the genome of a particular individual $i$ can calculate the log-likelihood ratio statistic for $i$ as follows:

$$L(Q, d_i, x) = \sum_{j \in Q} d_{ij} \log \frac{\tilde{p}_j}{x_j} + (1 - d_{ij}) \log \frac{1 - \tilde{p}_j}{1 - x_j}.$$  \hspace{1cm} (5.2)

Finally, the attacker claims that an individual $i \in T$ is in the dataset $D$ when $L(Q, d_i, x) \leq \theta$. The choice of $\theta$ reflects the adversary’s preferred balance between sensitivity and specificity of the membership inference attack.
5.3 Threat Models

Our threat models for membership inference attacks on the Beacon service and summary statistics are anchored in the likelihood ratio test (LRT) attack described above. However, the LRT attack leaves open three questions. First, what is the attacker’s target set $T$? Second, how does the attacker arrive at the choice of a threshold $\theta$ to determine which membership claims are made? Third, what is the set of queries used in the attack? Since we do not know a priori which individuals will be targeted, we make the worst-case assumption that $D \subseteq T$; in other words, the attacker targets everyone in the dataset, possibly along with others. Our threat modeling leads to several variants of the LRT attack along the remaining two dimensions.

5.3.1 Choosing the Inference Threshold

We investigate two approaches that an adversary may use to determine when to make a membership inference claim about an individual: fixed-threshold and adaptive attacks. In fixed-threshold attacks, an adversary uses a predefined threshold $\theta$, which is fixed for the inference attack. This is a common threat model in the literature [69, 114] and reflects an opportunistic attacker who initially uses a private dataset to simulate LRT attacks by splitting individuals into those in a simulated data release and those who are not. These offline simulations are then used to identify the $\theta$ that best
balances precision and recall with respect to the attacker’s preferences about these. The practical consequence of assuming a fixed $\theta$ at the time of the attack is that $\theta$ is not adjusted based on query responses.

Not considering queries in determining the threshold $\theta$ is consequential once we consider defensive measures that modify query responses. The fixed-threshold model has some limitations - the attacker could, for instance, choose a higher value of $\theta$ than what was used by any implemented defensive measures, which would subsequently lead to a violation of privacy. Further, the attacker might look to the distribution of LRT scores in order to separate the two populations. For example, if modified query responses preserve a clear separation in LRT statistics between individuals who are in and not in the data release, a simple clustering of the statistics would enable the attacker to effectively identify those in the release. Consequently, we additionally consider a stronger adaptive threat model that sets $\theta(x_Q)$ as a function of the responses $x_Q$ to queries $Q$, but with the aim of limiting the false positive rate for any membership claims to be at most $\alpha$. This in turn forces the defensive measures to ensure that the LRT scores for individuals in the release and those not in the release be sufficiently well-mixed. Further, limiting the false positive rate avoids having to consider unreasonable attacks such as claiming that all individuals are in the release. This adaptive threat model requires that the attacker can set the threshold in precisely the right place based on actual queries $Q$. However, this can effectively be accomplished with the aid of simulation experiments using a private dataset (now, simulating data releases that implement the defensive measures). This threat model captures highly informed attackers, for example, those who do systematic harvesting of sensitive data for profit.

In this thesis, we consider the following instantiation of an adaptive attacker. Given a dataset $D$ for which summary information $x$ is released either in the form of a Beacon or AAFs, and a dataset $\bar{D}$ of individuals not in $D$, an adaptive attacker sets the prediction threshold $\theta$ to be the average of the $K$ lowest LRT scores of individuals in $\bar{D}$. Let $\bar{D}^{(K)} \subseteq \bar{D}$ be the set of individuals with the $K$
lowest LRT scores in $\tilde{D}$.

$$\theta(\tilde{D}, x) = \frac{1}{K} \sum_{i \in \tilde{D}^{(K)}} L(Q, d_i, x)$$  \hspace{1cm} (5.3)

Since the attacker knows that none of the individuals in $\tilde{D}$ are present in the dataset $D$, setting $K$ allows the attacker to tune the maximum allowable false-positive rate for predictions using the data release $x$.

### 5.3.2 Query Process - Beacons

In this chapter, we distinguish between three mechanisms of query access that can be provisioned to a Beacon: 1) **batch** query, 2) **unauthenticated online** query access, and 3) **authenticated online** query access. In the **batch** setting, it is assumed that the attacker queries all $m$ SNP positions effectively simultaneously. In a sense, this is the most favorable setting for the adversary, as it provides maximum information for making membership claim decisions. It is also the setting that has received much of the attention in prior literature [69, 114]. However, typical Beacons in practice (such as the service provided by the GA4GH Beacon Network) can be queried **sequentially**, and we thus need to ensure that privacy is guaranteed even in such settings. Consequently, we also consider two online settings. The **unauthenticated online** setting assumes that the attacker can submit an arbitrary subset of queries. This is because if queries are not connected to a particular identity, there is no way for the Beacon to know which queries have been made by the same individual in the past. The public Beacon Network is an example of this situation. The **authenticated online** setting, by contrast, assumes that we can keep track of all the past queries by each individual, including the potential adversary. This entails only allowing registered and verified users access (we assume no collusion), and allows (as we show in the next chapter) privacy guarantees with significantly higher utility for such users. Fig. 5.1 provides an overview of the threat models, categorized by inference threshold and query process.

We now make an observation that enables us to uniformly talk about the three variants of the query process above; note that the sole mathematical distinction between them is the set of
queries $Q$ that we are concerned about. In the batch setting, $Q = S$, the set of all queries: that is, the LRT statistics of relevance for the purposes of membership inference attack is computed with respect to the set of all possible queries. In the online authenticated online setting, $Q$ is the set of all past queries, together with the current query $j$, since we are concerned that an individual $i$ may be identifiable as soon as $L(Q, d_i, x)$ drops below $\theta$ (or $\hat{\theta}(Q)$, in the adaptive model). Finally, in the unauthenticated online setting, since we do not know which set of queries have been asked by the adversary, or what the target set is, we make the worst-case assumption that the adversary makes most identifying queries for each individual $i$, that is, the set of queries $Q_i$ is specific to each individual in the Beacon and minimizes $L(Q_i, d_i, x) - \hat{\theta}(Q_i)$ for each individual. Since the choices for adversarial queries $Q$ (or $Q_i$ in the unauthenticated online case) are thus isomorphic with the particular query process in the threat model, we henceforth simply focus on two aspects of threat models: 1) the choice of the query set $Q$ and 2) whether $\hat{\theta}(Q)$ depends on the responses to $Q$.

5.4 The Data Custodian’s Options

Armed with the above attack models, we now briefly discuss potential defensive measures against such attacks. While we leave context-specific details to the upcoming two chapters, we now describe some techniques the data custodian may employ to try and defend against MI attacks. Broadly, these techniques involve changing the data release in some form, typically by injecting noise or by suppressing a subset of the data. A summary of prior work in this direction is included in Chapter 2. In both cases, this presents an inherent tradeoff between the privacy of individuals in the dataset, and the utility of the dataset, which naturally decreases as the data release is amended prior to its release.

5.4.1 Injecting Noise

In the first technique considered, the data release is modified by the injection of noise designed to thwart MI attacks. In the case of Beacons, this takes the form of falsifying Beacon responses,
which is to answer a query about an allele’s presence in the dataset untruthfully. We refer to this as *flipping* SNVs and treat the noise injected as a binary vector of decisions, i.e. whether or not the response for each SNV is falsified. The next chapter discusses factors that affect whether or not the output for a given SNV may be flipped prior to release, leveraging bounds on the genomic sequencing error rate.

In the case of AAFs, the noise injected into the data takes the form of a real-valued vector $\delta$, such that the AAF $x_j$ for each SNV $j$ becomes $x_j + \delta_j$. In other words, this is a randomization-based defense where, by effectively falsifying the frequency of each alternate allele, the defender reduces the attacker’s ability to accurately calculate LRT scores that in turn rely on accurately reported AAFs.

### 5.4.2 Suppressing SNVs

The second method the defender may employ is to selectively remove all information about certain SNVs from the data release. Qualitatively, this is different from adding high-magnitude noise to an SNV because a noisy output provides some information (even though it may be false information), but suppressing the SNV offers no information at all. We refer to the act of suppressing SNVs as *masking* in the remainder of this thesis.

### 5.5 Conclusion

In this chapter, we presented two models of genomic data sharing - namely Beacon services and the sharing of alternate allele frequencies. We classified Beacons based on whether SNVs are queried all at once or one at a time, and by whether user authentication is required to use the beacon. We further presented an overview of LRT-based MI attack models along with a novel adaptive attacker model that seeks to separate the LRT scores of individuals in a target dataset from the scores of a separate set of individuals not in the dataset. In the upcoming chapters, we formally present the problem
of protecting privacy in such data releases as optimization problems and propose highly-scalable greedy algorithms to approximately solve them.
Chapter 6: Protecting Privacy in Genomic Data Beacons

In the previous chapter, we laid the foundations of genomic data-sharing mechanisms, including the use of Beacon services that expose the presence or absence of alternate alleles in a dataset. In this chapter, we set out to formalize the problem of defending against LRT-based membership inference attacks as discussed in Chapter 5 in the context of Beacon services and develop highly scalable algorithms to do so. To this end, we leverage problem structure to first identify certain special cases that allow us to simplify the setting for the sake of theoretical understanding, which we then extend to settings with more realistic assumptions, as well as the novel adaptive attack model presented in Chapter 5. In particular, in this chapter, we assume that the defender aims to preserve privacy for all individuals in the dataset. Using the notation from the last chapter, we say that an individual who is part of the dataset $D$ is in the Beacon, and individuals who are not in the dataset (the set $\bar{D}$) are considered not in the Beacon.

6.1 Privacy and Utility Goals

We now formalize the goals for protecting Beacon service privacy. Following the framework of the 2016 iDash Practical Protection of Genomic Data Sharing through Beacon Services challenge [69], where the Beacon privacy problem was standardized, the primary means we consider for protecting the privacy of individuals in the Beacon is by flipping the responses to a subset of possible SNV queries. We encode the choice of which responses to flip as a binary vector $\delta = \{\delta_1, \ldots, \delta_m\}$, where $\delta_j = 1$ implies that the response to query $j \in S$ is flipped, and $\delta_j = 0$ means that the query answer is unchanged. We denote the subset of flipped queries by $F \subseteq S$, where $F = \{j \in S : \delta_j = 1\}$. We
define $x_Q(F)$ as the vector of Beacon responses to queries $Q$ when the set $F$ of responses is flipped.

Our privacy goal is to ensure that privacy is preserved for all individuals in the Beacon, where privacy is measured with respect to threat models discussed in Section 5.3, each of which ultimately leverages a form of the LRT attack for membership inference. Let $L(Q_i, d_i, x_{Q_i}(F))$ be the LRT statistic for individual $i$ and $\theta(x_{Q_i}(F))$ the threshold after we flip the set $F$ of query responses. Formally, we wish to guarantee that

$$\forall i \in D, \quad L(Q_i, d_i, x_{Q_i}(F)) - \theta(x_{Q_i}(F)) \geq 0. \quad (6.1)$$

In the case of a fixed-threshold attack, $\theta(x_{Q_i}(F))$ is a constant independent of $x$; in the batch setting, $Q_i = S$ for all $i$; and in the authenticated online setting, $Q_i = Q$ for all $i$, where $Q$ is the set of queries made thus far by the authenticated user.

Clearly, we can preserve privacy by simply shutting down the Beacon service. However, there is value in genomic data sharing, and it is this value that has motivated creative ideas for sharing it in a privacy-respectful manner. Our broader goal, therefore, is thus to achieve privacy, as defined by Eq. (6.1), with a minimal impact on utility, which in this context means minimizing the number of query responses that are flipped. We formalize the resulting optimization problem, which we refer to as the Beacon-Privacy-Problem as follows:

$$\min_{F \subseteq S} |F| \text{ subject to:}$$

$$L(Q_i, d_i, x_{Q_i}(F)) - \theta(x_{Q_i}(F)) \geq 0 \forall i \in D. \quad (6.2)$$

We aim to solve Problem 6.2 effectively and efficiently for each of the threat model settings described in Section 5.3.

To begin, we make a useful structural observation that significantly limits the set of query responses to be considered for flips: we would never want to flip responses from 0 to 1. We formalize this in the following proposition.

**Proposition 6.1.1.** Suppose that $x_j = 0$ given the Beacon dataset. Then $\delta_j = 1$ can never increase the LRT statistic for any individual $i \in D$, provided sampling error $\gamma < \frac{R_j}{R_{n-1}}$ for all $j$.  

64
Proof. Consider the $j^{th}$ query. If individual $i$ does not have an alternate allele at position $j$, flipping the beacon response makes no difference (refer Eq. 5.1; $d_{ij} = 0$ when an individual does not have the alternate allele at position $j$). When the individual does have an alternate allele at position $j$ (i.e. $d_{ij} = 1$), changing the beacon response $x_j$ from 0 to 1 changes the contribution of query $j$ to the LRT score from $\log \frac{R^j_n}{\gamma R^{j-1}_n}$ to $\log \frac{1-R^j_n}{1-\gamma R^{j-1}_n}$. Given sampling error $\gamma < \frac{R^j_n}{R^{j-1}_n}$, dividing on both sides by $\gamma$, we have $\gamma R^{j-1}_n > 1$ (as $\gamma \geq 0$), and, consequently, $\log \frac{R^j_n}{\gamma R^{j-1}_n} > 0$. Since $\gamma < \frac{R^j_n}{R^{j-1}_n}$, multiplying both sides by $R^{j-1}_n$ yields $R^j_n > \gamma R^{j-1}_n$ (since $R^{j-1}_n \in [0, 1]$), which implies that $1 - R^j_n < 1 - \gamma R^{j-1}_n$. Dividing both sides by $1 - \gamma R^{j-1}_n$, we have $\frac{1-R^j_n}{1-\gamma R^{j-1}_n} < 1$, since $\gamma \in [0, 1]$ and $R^{j-1}_n \in [0, 1]$, which in turn implies that $\log \frac{1-R^j_n}{1-\gamma R^{j-1}_n} < 0$. □

Since a privacy violation means that the LRT statistic for at least one individual in the Beacon is too small, our goal is necessarily to increase these scores until privacy is guaranteed for all individuals in the Beacon. Henceforth, we assume that $\gamma < 0.25 \leq \frac{R^j_n}{R^{j-1}_n} = (1 - p_j)^2$ (since $p_j < 0.5$ for all $j$). Consequently, Proposition 6.1.1 implies that flipping a 0 response to a 1 is counterproductive, and we need not consider it as a possible solution to Problem 6.2. While flipping a 0 response to a 1 may in principle help lower the scores for individuals not in the beacon and therefore result in better-mixed LRT scores, we consider the issue of mixing more systematically as a part of the adaptive attack model below. Further, a majority (over 96%) of beacon responses in our data are initially 1.

Then, without loss of generality, we can assume that our consideration set $S$ includes only the query responses which are initially 1. Next, we show that even in a very restricted special case, the problem of minimizing the number of flips in order to guarantee the privacy of all individuals in the Beacon is $\mathbb{NP}$-Hard. First, we define a decision version of Problem 6.2, which we refer to as Beacon-Privacy-D.

**Definition 6.1.2 (Beacon-Privacy-D). Input:** A collection of individuals $i \in D$ with genomic information, and induced Beacon query responses $x$; a constant $k$. **Question:** Can we flip a subset $F$ of query responses where $|F| \leq k$ and $L(Q_i, d_i, x_{Q_i}(F)) - \theta(x_{Q_i}(F)) \geq 0$?
Theorem 6.1.3. **Beacon-Privacy-D** is \(\mathbb{NP}\)-Complete even if \(\gamma = 0\) and \(\Theta(x_Q(F))\) is a constant.

Proof. We reduce from the Set Cover problem, which we now formally define.

**Definition 6.1.4** (Set Cover). **Input:** A universe \(U\) of elements, and a collection of sets \(H = \{H_1, \ldots, H_n\}\) with \(H_j \subseteq U\) and \(\cup_j H_j = U\); a constant \(k\). **Question:** Is there a subset \(T \subseteq H\) of sets such that \(U = \cup_{t \in T} t\) and \(|T| \leq k\).

First, note that Beacon-Privacy-D is in \(\mathbb{NP}\), since given a set \(F\) of flips, it is straightforward to verify that the privacy constraint holds for each individual \(i\). To prove that the problem is \(\mathbb{NP}\)-hard, we reduce from the Set Cover problem. First, observe that in the case where \(\gamma = 0\), and by Proposition 6.1.1, to guarantee privacy for any individual \(i \in D\), it suffices to flip a single response \(x_j\) from 1 to 0 from all with \(d_{ij} = 1\) (if we use the convention that division by 0 results in \(\infty\), any such flip causes \(L(Q_i, d_i, x) = \infty\)).

Now, let elements of \(U\) correspond to individuals in the Beacon, i.e., \(D = U\). Let subsets \(H_j\) correspond to queries \(j\), where each element represents an individual \(i\) with \(d_{ij} = 1\). Since without loss of generality, we can assume that each \(H_j\) is non-empty (since we can ignore any empty subsets in both Set Cover, and in the construction of Beacon-Privacy-D instance by Proposition 6.1.1), this also implies that the corresponding query response is \(x_j = 1\), as at least one individual has \(d_{ij} = 1\). For any individual (element of \(U\)) \(i \notin H_j\), we set \(d_{ij} = 0\). Furthermore, since \(\cup_j H_j = U\), each individual has at least one \(j\) with \(d_{ij} = 1\). Finally, the constant \(k\) is now the constraint on the size of \(F\), the subsets of queries to flip.

Suppose that we find the set \(F\) of queries to flip that guarantees privacy. Let \(T = F\), that is, indices of subsets \(H_j\) in Set Cover. Since \(|F| \leq k\), \(|T| \leq k\), so it suffices to show that \(U = \cup_{t \in T} t\). Solving Beacon-Privacy-D means that the set of flips \(F\) guarantees privacy for each \(i \in D = U\). By the observation above that it suffices to flip any query \(j\) with \(d_{ij} = 1\) to guarantee the privacy of \(i\), \(H_j\) is the subset of individuals for whom privacy is guaranteed, and, thus, \(\cup_{j \in F} H_j = U\), since we must guarantee privacy for all individuals. Since \(T = F\), we have covered the universe \(U\).
For the other direction, suppose that there exists a solution to Set Cover, $T$ with $|T| \leq k$ and $U = \bigcup_{t \in T} t$. Set $F = T$ and flip all queries with $j \in F$. Since it suffices to guarantee privacy for any $i \in D$ by flipping any query $j$ with $d_{ij} = 1$, and since $H_j$ is the collection of all individuals for whom we can guarantee privacy by flipping $j$, and since $\bigcup_{t \in T} t = U$, by our construction this implies that privacy is guaranteed for all $i \in D$. \qed

### 6.2 The Batch Setting

We begin by considering the batch setting in which the adversary submits a set of queries $Q$ all at once, where $Q \neq \emptyset$. This provides the building blocks for all the query settings in our threat model, both what we called batch query setting with $Q = S$ above, and for the online settings.

Recall that a binary vector $x$ corresponds to the true query responses, while $\delta$ represents whether responses have been flipped. Let $Q_1$ be the subset of queries $Q$ with $x_j = 1$ and $Q_0$ be the subset with $x_j = 0$. Let $A_j = \log \frac{1 - R^n_j}{1 - \gamma R^n_{j-1}}$ and $B_j = \log \frac{R^n_j}{\gamma R^n_{j-1}}$. We can then rewrite $L_i(Q, d, x)$ as follows:

$$L_i(Q, d, x) = \sum_{j \in Q_1} d_{ij} A_j + \sum_{j \in Q_0} d_{ij} B_j.$$  

Moreover, by Proposition 6.1.1, we never flip queries in $Q_0$, which means that for our purposes the second term above is a constant. Now, if we apply the query flip strategy $\delta$, the resulting LRT statistic, which we denote by $L_i(Q, x, \delta)$, becomes

$$L_i(Q, x, \delta) = \sum_{j \in Q_1} d_{ij}((1 - \delta_j)A_j + \delta_j B_j) + \sum_{j \in Q_0} d_{ij} B_j$$

$$= \sum_{j \in Q_1} \delta_j d_{ij} (B_j - A_j) + \sum_{j \in Q_1} d_{ij} A_j + \sum_{j \in Q_0} d_{ij} B_j.$$  

Define $\Delta_{ij} = d_{ij}(B_j - A_j)$ and $\eta_i = \sum_{j \in Q_1} d_{ij} A_j + \sum_{j \in Q_0} d_{ij} B_j$. Then

$$L_i(Q, x, \delta) = \sum_{j \in Q_1} \Delta_{ij} \delta_j + \eta_i.$$  

(6.3)

Note that $\eta_i$ is actually also a function of the set of queries $Q$. For the remainder of this section, this will not be important and so we omit this dependence. However, this becomes important in the
online setting below. Next, we consider approaches for solving the Beacon-Privacy-Problem first in the fixed-threshold and subsequently in the adaptive attacks.

### 6.2.1 Fixed-Threshold Attacks

We begin by presenting an integer linear programming (ILP) approach for optimally solving the general Beacon-Privacy-Problem in the batch setting with fixed-threshold attacks. This is a straightforward consequence of the problem structure that we had already derived above. First, note that we wish to minimize the number of flips, which is equivalent to minimizing the number of 1s in $y$. Second, note that the privacy constraint is $L_i(Q, x, \delta) \geq \theta$ which is linear in $y$. Consequently, the following ILP solves the Beacon-Privacy-Problem:

$$\min_{\delta \in \{0,1\}^m} \sum_j \delta_j \ \text{subject to:}$$

$$\sum_{j \in Q_i} \Delta_{ij} \delta_j + \eta_i \geq \theta \ \forall \ i \in D.$$  \hspace{1cm} (6.4)

This ILP has worst-case exponential running time and, as a result, will have trouble scaling to large problems that include thousands of individuals and millions of SNPs. We address the scalability in two ways: first, we identify important special cases which either enable the ILP solvers to leverage problem structure, or admit an approximation algorithm with worst-case guarantees; second, we present two greedy algorithms for solving the general variant of the Beacon-Privacy-Problem. The key property of all the solutions we propose is that they satisfy the privacy constraints by construction.

### Small Sequencing Error Rates

Genomic sequencing error rates $\gamma$ are often quite small, on the order of $10^{-6}$. We now show that for sufficiently small sequencing error rates (with $\gamma = 0$ a special case), we can represent the Beacon-Privacy-Problem as a Minimum Set Cover instance. This, in turn, implies that we can solve our problem using a greedy algorithm with a logarithmic worst-case approximation guarantee.
Generally speaking, for a sufficiently small $\gamma$, the terms $B_j$ will be extremely large for any query $j$ that we may choose to flip from a 1 to a 0 and, in particular, will be much larger than $A_j$. Thus, for every individual $i$ in the Beacon, flipping any $j \in Q_1$ will result in a very large increase in $\Delta_{ij} = d_{ij}(B_j - A_j)$. This increase will, indeed, be so large as to guarantee that $L_i(Q, x, \delta) \geq \theta$. As a consequence, it will suffice to flip any query with $d_{ij} = 1$ for $i$ to no longer be categorized as in the Beacon by the attack. Of course, just how small $\gamma$ needs to be for this to work will be a function of both the problem parameters $R^l_j$ and $R^l_{n-1}$, as well as $\theta$. We emphasize that this line of reasoning is specific to the fixed-threshold attack model; the issue is far more subtle in adaptive attacks (Section 6.2.2). Next, we make this premise precise.

For each individual $i \in D$, define $P_i = \{j \in Q_1|d_{ij} = 1\}$; in other words, $P_i$ is the set of all queries $j$ for which a) $x_j = 1$ and b) the individual $i$ actually has the associated alternate allele for query $j$, i.e., $d_{ij} = 1$. We now provide a sufficient condition on $\gamma$ such that we can flip any query in $P_i$ for each $i \in D$ to guarantee privacy under fixed-threshold attacks.

**Definition 6.2.1.** A set of queries $F$ that have been chosen to flip is a Beacon-Cover if $\forall i \in D, F \cap P_i \neq \emptyset$.

In words, $F$ is a Beacon-Cover if each individual in the Beacon is covered by some query flipped in $F$ which is also in $P_i$. We now define two additional components to the notation that will be useful throughout our analysis. First, define

$$R_n = \min_{j \in Q_1} \log\left(\frac{R^l_j}{1 - R^l_{n-1}}\right).$$

Second, define

$$\eta = \min_i \left( \sum_{j \in Q_1} d_{ij} \log(1 - R^l_n) + \sum_{j \in Q_0} d_{ij} \log \frac{R^l_j}{0.25 R^l_{n-1}} \right)$$

The following proposition presents a bound on $\gamma$ that ensures that a Beacon-Cover guarantees privacy against fixed-threshold attacks.

**Theorem 6.2.2.** Suppose that $\gamma \leq \frac{1}{1 + e^{\theta - \eta - R_n}}$. Then, if $F$ is a Beacon-Cover, it guarantees privacy for all $i \in D$ against fixed-threshold attacks with threshold $\theta$. 69
Proof. Recall that for a fixed-threshold attack, Beacon privacy guarantee for a given \( F \) and associated indicator vector \( y \), formalized in Equation (6.1), is that

\[
\forall i \in D, \quad L_i(Q, x, \delta) = \sum_{j \in Q_1} \Delta_i \delta_j + \eta_i \geq \theta.
\]

Let \( \Delta_i = \min_{j \in P_i} \Delta_{ij} \). Then if for each \( i, \Delta_i + \eta \geq \theta \), the condition above certainly follows as well, since \( \sum_{j \in Q_1} \Delta_i \delta_j \geq \Delta_i \) by definition of a Beacon-Cover, and

\[
\eta = \min_i \left( \sum_{j \in Q_1} d_{ij} \log(1 - R_{nj}^j) + \sum_{j \in Q_0} d_{ij} \log \frac{R_{nj}^j}{0.25R_{nj-1}} \right)
\]

\[
\leq \min_i \left( \sum_{j \in Q_1} d_{ij} \log \frac{1 - R_{nj}^j}{1 - \gamma R_{nj-1}} + \sum_{j \in Q_0} d_{ij} \log \frac{R_{nj}^j}{\gamma R_{nj-1}} \right)
\]

\[
\leq \eta_i.
\]

Now, \( \Delta_i = \min_{j \in P_i} d_{ij}(B_j - A_j) \), and since \( \Delta_{ij} > 0 \) for any \( j \in P_i \), \( d_{ij} = 1 \) for any \( j \in P_i \). Consequently,

\[
\Delta_i = \min_{j \in P_i} (B_j - A_j) = \min_{j \in P_i} \left( \log \frac{R_{nj}^j}{\gamma R_{nj-1}^j} - \log \frac{1 - R_{nj}^j}{1 - \gamma R_{nj-1}^j} \right)
\]

\[
= \min_{j \in P_i} \left( \log \frac{R_{nj}^j}{1 - R_{nj}} + \log \frac{1 - \gamma R_{nj-1}^j}{\gamma R_{nj-1}} \right)
\]

\[
\geq \min_{j \in P_i} \left( \log \frac{R_{nj}^j}{1 - R_{nj}} \right) + \min_{j \in P_i} \left( \log \frac{1 - \gamma R_{nj-1}^j}{\gamma R_{nj-1}} \right)
\]

\[
\geq R_n + \min_{j \in P_i} \left( \log \left( \frac{1}{\gamma R_{nj-1}} - 1 \right) \right) \geq R_n + \log \left( \frac{1}{\gamma} - 1 \right),
\]

where the last inequality follows since \( R_{nj-1} \leq 1 \). Now, if \( \gamma \leq \frac{1}{1 + e^{\theta - \eta - R_n}} \), then

\[
\log \left( \frac{1}{\gamma} - 1 \right) \geq \log (e^{\theta - \eta - R_n}) = \theta - \eta - R_n.
\]

Consequently, \( \Delta_i \geq \theta - \eta \) for each \( i \in D \), which is just a rearranging of the desired condition above. \( \square \)
The benefit of Theorem 6.2.2 is that it suffices for $F$ to “cover” each individual in the sense that for every individual $i$ in the Beacon, there is at least one flipped query in $F$ that suffices to ensure that the score $L_i(Q, x, \delta) \geq \vartheta$, that is, to ensure that $i$’s privacy is preserved under the fixed-threshold threat model in the batch setting. This, in turn, allows us to represent the Beacon-Privacy-Problem as a Min-Set-Cover instance. The Min-Set-Cover problem is the optimization variant of Set Cover, which we now define formally.

**Definition 6.2.3 (Min-Set-Cover).** **Input:** A universe $U$ of elements, and a collection of sets $H = \{H_1, \ldots, H_n\}$ with $H_j \subseteq U$ and $\bigcup_{j} H_j = U$; a constant $k$. **Goal:** Minimize $|T|$ over $T \subseteq H$ such that $U = \bigcup_{t \in T} t$.

We now show how to represent our problem as an instance of the Min-Set-Cover problem. The key advantage of this representation will be a greedy algorithm for solving the Beacon-Privacy-Problem in this setting that yields a logarithmic worst-case approximation guarantee [115]. The representation is similar to the one used in the proof of Theorem 6.1.3 but, of course, is in the opposite direction. Specifically, we are given a Beacon-Privacy-Problem instance, which we now use to construct a Min-Set-Cover instance. Let $U = D$, the set of the individuals in the Beacon, while each $H_j$ corresponds to query $j$, and is comprised of the individuals $i \in D$ whose privacy will be protected if we flip $j$. Formally, $H_j = \{i \in D | j \in R\}$.

Now, we can leverage the greedy algorithm for Min-Set-Cover to solve our problem. The greedy algorithm works as follows. The collection of subsets $T$ is initialized to be empty. Then, in each iteration, we add a subset $S_j$ to $T$ which maximizes the number of elements in $U$ it covers that are not already covered by $T$. We stop when the entire universe $U$ is covered. Algorithm 1, which we refer to as Greedy Min Beacon Cover presents a direct adaptation of this to our problem. The following is, thus, a direct corollary of Theorem 6.2.2.

**Corollary 6.2.4.** Suppose that $\gamma \leq \frac{1}{1+e^{\delta - \eta - R_n}}$. Then Algorithm 1 gives an $O(\log(n))$-approximation to the Beacon-Privacy-Problem.
Algorithm 1: Greedy Min Beacon Cover Algorithm.

**Input:** A set \( D \) of individuals in the Beacon, a subset \( P_i \) for each individual, and a collection of queries \( Q \).

**Output:** Subsets of queries \( F \subseteq S \) to flip.

**Initialization:** \( F = \emptyset, C = \emptyset \).

**while** \( (B \setminus C) \neq \emptyset \) **do**

1. Set \( l = 1, T = \emptyset, N = -1 \).
2. **for** \( j \in (Q \setminus F) \) **do**
   1. Set \( T_j = \{i \in (D \setminus C)|j \in P_i\} \).
   2. **if** \( |T_j| > N \) **then**
      1. Set \( T = T_j \).
      2. Set \( N = |T| \).
      3. Set \( l = j \).
   **end**
3. **end**
4. Set \( F = F \cup l \).
5. Set \( C = C \cup T \).

**end**

Alternate Allele Frequencies Drawn from the Beta Distribution

A common assumption in prior literature is that the alternate allele frequencies (AAF) are drawn from a Beta distribution [69], replacing the \( R^j_n \) and \( R^j_{n-1} \) terms in the LRT score calculation with the expectation over the distribution, which we denote by \( \bar{R}_n \) and \( \bar{R}_{n-1} \), respectively. This, in turn, means that \( A_j \) and \( B_j \) are independent of \( j \), and we now denote them by constants \( A \) and \( B \), respectively. As a result, \( \Delta_{ij} = d_{ij}(B - A) \) and \( \eta_i = A \sum_{j \in Q_1} d_{ij} + B \sum_{j \in Q_0} d_{ij} \), and we obtain a simpler expression for the LRT statistic induced by query flips \( y \):

\[
L_l(Q, x, \delta) = (B - A) \sum_{j \in Q_1} d_{ij} \delta_j + \eta_i.
\]

Consequently, the privacy condition for each \( i \in D \) is equivalent to

\[
\sum_{j \in P_i} \delta_j \geq k_i, \quad \text{where} \quad k_i = \frac{\theta - \eta_i}{B - A}.
\]

Note that, under our assumption, \( \gamma < 1/4, \quad B - A > 0 \). This has two algorithmic implications. First, it yields a significantly simpler set of privacy constraints in the integer linear program (6.4)
to obtain the optimal solution to the Beacon-Privacy-Problem. Second, we can derive a natural greedy heuristic for this case which generalizes the Greedy Min Beacon Cover algorithm above. The high-level idea of the greedy heuristic is to iteratively choose a query result to flip that affects the largest number of individuals. This idea is formalized in Algorithm 2, which we term Greedy $k$-Cover.

Algorithm 2: Greedy $k$-Cover

**Input:** A set $D$ of individuals in the Beacon, a subset $P_i$ and a constant $k_i$ for each individual, and a collection of queries $Q$.

**Output:** Subsets of queries $F \subseteq S$ to flip.

**Initialization:** $F = \emptyset$, $C = \emptyset$.

while $(D \setminus C) \neq \emptyset$ do
  Set $l = 1, N = -1$.
  for $j \in (Q \setminus F)$ do
    Set $T_j = \{i \in (D \setminus C) | j \in P_i\}$.
    if $|T_j| > N$ then
      Set $N = |T_j|$.
      Set $l = j$.
    end
  end
  Set $F = F \cup l$.
  for $i \in (D \setminus C)$ do
    if $F \cap P_i \geq k_i$ then
      Set $C = C \cup i$.
    end
  end
end

Heuristic Approach for the General Case

While the two special cases considered above are instructive, the assumptions in these do not always hold. On the other hand, the integer programming approach (6.4) is unlikely to scale to large problems, especially when we have millions of queries to consider. We now present a general-purpose greedy heuristic that builds on the Greedy $k$-Cover algorithm. First, observe that in the
general setting, there is no longer a meaningful notion of "cover", since each query and individual have an associated specific contribution $\Delta_{ij}$. On the other hand, recall that $\Delta_{ij} = d_{ij}(B_j - A_j)$ and, consequently, if $j \in P_i$, then the marginal impact of flipping query $j$ on the LRT statistic of $i$ only depends on query $j$. Define $\Delta_j = (B_j - A_j)$, so that $\Delta_{ij} = d_{ij}\Delta_j$. For any subset of individuals $P \subseteq D$ and query $j$, let $T_j = \{i \in P | j \in P_i\}$ be the set of individuals for whom $j \in P_i$. We can then define the average marginal contribution of each query $j$ and population $P$ as

$$\bar{\Delta}_j(P) = \frac{|T_j|\Delta_j}{|P|}.$$ 

**Algorithm 3: MI Greedy**

**Input:** A set $D$ of individuals in the Beacon, a subset $P_i$ and $\eta_i$ for each individual, marginal contributions $\Delta_j$ for each query, a collection of queries $Q$, and a threshold $\theta$.

**Output:** Subsets of queries $F \subseteq S$ to flip.

**Initialization:** $F = \emptyset$, $C = \emptyset$.

while $(D \setminus C) \neq \emptyset$, $C = \emptyset$.

for $j \in (Q \setminus F)$ do

    Set $T_j = \{i \in (D \setminus C) | j \in P_i\}$.

    Set $\bar{\Delta}_j = \Delta_j \cdot \frac{|T_j|}{|D \setminus C|}$.

    if $\bar{\Delta}_j > N$ then

        Set $N = \bar{\Delta}_j$.

        Set $l = j$.

    end

end

Set $F = F \cup l$.

for $i \in (D \setminus C)$ do

    if $\sum_{j \in F} \Delta_{ij} + \eta_i \geq \theta$ then

        Set $C = C \cup i$.

    end

end

The greedy heuristic we propose iteratively chooses a query $j$ to flip with the largest marginal contribution $\bar{\Delta}_j(P)$, where the population $P$ consists of the individuals whose privacy has yet to
be guaranteed. Note that for this heuristic to work reasonably well, it is crucial that $\Delta_j > 0$. This is indeed the case as shown in the proof of Proposition 6.1.1 (which implies that $B_j - A_j > 0$) if $\gamma < 1/4$. This means that as we flip query responses, we cannot decrease the LRT score for any individual, and, consequently, any individual $i$ whose privacy is already protected remains protected. This heuristic, which we call MI-Greedy (where MI stands for Marginal Impact), is formalized as Algorithm 3.

### 6.2.2 Adaptive Attacks

The threat model articulated so far has assumed the attacker fixes the decision threshold $\theta$ prior to executing any queries, and $\theta$ is independent of queries. However, since our defense involves the alteration of query responses, an adaptive attacker should make use of query responses in identifying an appropriate threshold. In other words, the adaptive attacker chooses an inference threshold based on a maximum allowable false positive rate, *given the defense*. Limiting the false positive rate ensures that the attack is reasonable, i.e., avoids cases such as claiming that all individuals or no individuals are in the beacon. From the perspective of privacy protection, this means that it is not sufficient to ensure that LRT statistics for all individuals exceed some predefined threshold, but we must actually ensure that the Beacon and non-Beacon populations are *well-mixed* in terms of the respective LRT statistics as calculated *based on the modified query responses*. Analogously, this can be interpreted as minimizing the area under the ROC curve for attacker performance. We now formalize this idea.

Consider our encoding $\delta_y$ of which query responses to flip and let $\bar{D}$ denote a set of individual genomes not in the Beacon (e.g., a data sample of these from the general population). The LRT statistic for each individual $i \in \bar{D}$ can be computed just as for any $i \in D$. Let us take $K$ individuals from $\bar{D}$ with the lowest LRT statistics, denoting the set of these individuals by $\bar{D}^{(K)}$. The concrete instantiation of the adaptive threat model then uses the following threshold:

$$\theta(Q) = \frac{1}{K} \sum_{k \in \bar{D}^{(K)}} \left( \sum_{j \in Q_1} \Delta_{kj} \delta_j + \eta_k \right)$$
We can interpret this as representing an attacker’s tolerance for false positives. For example, if the distribution of LRT scores is approximately symmetric around the mean, then $K/(2|\bar{D}|)$ is approximately the false positive rate. As a result, the privacy constraint for each $i \in D$ in the adaptive attack setting becomes

\[
\sum_{j \in Q_1} \Delta_{ij} \delta_j + \eta_i \geq \frac{1}{K} \sum_{j \in Q_1} \left( \sum_{k \in D^{(K)}} \Delta_{kj} \delta_j + \eta_k \right)
= \sum_{j \in Q_1} \sum_{k \in D^{(K)}} \left( \frac{\Delta_{kj}}{K} \right) \delta_j + \sum_{k \in D^{(K)}} \left( \frac{\eta_k}{K} \right).
\]

Define
\[
\Delta_j^{(K)} = \sum_{k \in D^{(K)}} \frac{\Delta_{kj}}{K} \quad \text{and} \quad \eta^{(K)} = \sum_{k \in D^{(K)}} \frac{\eta_k}{K}.
\]

Rewriting the expression above, we then obtain the following privacy condition for $i \in B$:

\[
\sum_{j \in Q_1} (\Delta_{ij} - \Delta_j^{(K)}) \delta_j + \eta_i \geq \eta^{(K)}.
\] (6.5)

Finally, by defining $\Delta_{ij}^{(K)} = \Delta_{ij} - \Delta_j^{(K)}$, we can rewrite this in the form identical to Equation (6.3) for the fixed threshold attacks:

\[
\sum_{j \in Q_1} \Delta_{ij}^{(K)} \delta_j + \eta_i \geq \eta^{(K)}.
\] (6.6)

Superficially, this suggests that we can directly apply the methods developed in Section 6.2.1 for privacy protection against fixed-threshold attacks. And, indeed, we can directly incorporate the linear privacy constraint (6.6) into the linear integer program (6.4). However, this threat model now breaks the greedy algorithms we previously proposed. The first reason is that $\delta$ now figures as a part of the threshold $\hat{\theta}(Q)$ and, consequently, is embedded in $\Delta_{ij}^{(K)}$ in two potentially conflicting ways. The second (and related) issue is that while a fixed-threshold threat model implied, for $\gamma < 0.25$, that $\Delta_{ij} > 0$, this is clearly no longer necessarily the case for $\Delta_{ij} - \Delta_j^{(K)}$. This has two consequences: 1) greedily adding one query $j$ to the flip set $F$ may actually cause privacy violation of an individual whose privacy constraint was previously satisfied, and 2) the integer linear program (6.4) may no longer have a feasible solution even though it is feasible for a fixed $\hat{\theta}$. In practice, this means
that the choice of $K$ cannot be overly conservative. Moreover, to enable us to directly reuse the general-purpose greedy algorithm from Section 6.2.1 for privacy protection against the adaptive threat model, we only consider flipping queries $j$ for which $\Delta_{ij}^{(K)} \geq 0$ for all $i \in D$.

### 6.3 The Online Setting

Thus far, we assumed that the attacker submits all queries $Q$ all at once, computes LRT statistics, and decides which individuals to make membership claims about. In practice, however, queries to the Beacon arrive over time, and privacy violations may arise even inadvertently if the attacker is, say, a relative of an individual in the Beacon who happens to observe that a rare collection of minor alleles that their family member possesses is also in the Beacon. Since individual queries may increase as well as decrease LRT statistics, it may well be the case that queries flipped in anticipation of a batch attack—even with $Q = S$—nevertheless violate privacy for some query sequences. Consequently, in the online setting, we need to assure Beacon service privacy for subsets of queries.

However, note that in practice we may not need to be concerned about arbitrary subsets of queries: since it is optimal from an attacker’s perspective to make use of all query responses they have observed, we need only guarantee privacy for the subset of queries submitted by any user thus far. That is, of course, if we know which queries the user submitted. This issue of whether or not the Beacon service knows which queries have previously been submitted by a user motivates a natural distinction between two classes of online use settings that we discussed in Section 5.3: authenticated access, where the Beacon knows all prior queries (i.e., access requires authentication and identity is carefully verified), and unauthenticated access, where the Beacon does not have this information. Next, we formalize the online query setting, and subsequently consider in turn authenticated and unauthenticated access.
6.3.1 A Model of the Online Beacon

The online query setting is characterized by a sequence of \( T \) queries \( \{q_1, \ldots, q_T\} \), with \( q_t \in S \) denoting a \( t \)th query about a particular SNV (we alternatively refer to this as a query at time \( t \), with time here being equivalent to the order in the query sequence). Similarly, at each point in time, including \( t = 0 \) (i.e., before any queries), the Beacon can decide to flip a subset of query responses \( F_t \). In this setting, the set of all queries flipped is \( F = \bigcup_t F_t \); however, in the online setting, we need not flip them all at once. The reason we may choose to defer flipping a particular query response is that observed queries are informative, and a particular observed query sequence may warrant flipping many fewer responses than, say, a worst-case sequence or the batch of all queries \( S \). There is an additional constraint that we must impose on \( F_t \): query responses are commitments, in the sense that if at any point \( t \) we choose to honestly respond to a query \( j \), we must do so in the future; similarly, if we chose to flip the query response, we must do so in the future as well. Given this constraint, we assume without loss of generality that the query sequence is non-repeating, that is, for all \( 1 \leq t, t' \leq T, q_t \neq q_{t'} \) (since future identical queries are responded to exactly as the first time they are encountered).

At time \( 1 \leq t \leq T \), we have a collection \( Q_{t-1} \) of past queries, along with the query \( q_t \) that just arrived, resulting in the query set \( Q_t \) observed thus far. A privacy guarantee now entails that privacy of no individual \( i \in B \) is violated at any time \( t \). For a fixed-threshold threat model, this translates into the following privacy condition;

\[
\forall i, t, \quad L_i(Q_t, x) \geq \vartheta(Q_t).
\]

As we observed in Section 6.2.2, the condition has an analogous form for adaptive attacks. Since we are in the online setting, we can now choose subsets of queries to flip over time, rather than all at once. We can encode the associated decisions \( F_t \) as binary vectors \( y_t \), resulting in the following privacy condition:

\[
\forall i, t, \quad L_i(Q_t, x, \delta_t) \equiv \sum_{j \in Q_{1,t}} A_{ij} \delta_{j,t} + \eta_i(Q_t) \geq \vartheta(Q_t),
\]
where we now make it explicit that $\eta_i$ in the modified LRT statistics depends on the query set $Q_t$.

### 6.3.2 Authenticated Access

The key feature of authenticated access settings that we can leverage is the knowledge at any time $t$ of the prior queries $Q_{t-1}$ as well as the current query $q_t$ to which the Beacon is about to respond (effectively, assuming that there is no collusion among Beacon clients, unlike in the unauthenticated setting below). The following proposition makes the intuitive observation that in the authenticated access setting, one never needs to make a decision whether to flip a query or not at time $t$ for any $j \neq q_t$.

**Proposition 6.3.1.** For any $1 \leq t \leq T$, there is an optimal query flip policy with $F_t \subseteq \{q_t\}$.

This follows because if you wish to flip a particular query $j$, you need not implement this decision until you actually observe the query.

### Fixed-Threshold Attacks

We begin in the authenticated setting by again considering the fixed-threshold attacks. For this setting, our assumptions imply that $\Delta_j = B_j - A_j > 0$ for all $j$. Consequently, if responding honestly to the query $q_t$ would violate privacy, we would always wish to flip the response. This is captured in the following proposition.

**Proposition 6.3.2.** In the fixed-threshold threat model and authenticated access setting, if $\exists i \in D$ such that $L_i(Q_t, x, \delta) < \theta$, then $F_t = \{q_t\}$.

Proposition 6.3.2 suggests a simple online heuristic for ensuring privacy while minimizing the number of query flips: flip $j$ if and only if $q_t = j$ and adding $q_t$ violates privacy. This is formalized in Algorithm 4. Our experiments below demonstrate that this simple heuristic is remarkably effective in practice. We observe that, though this heuristic may not be optimal, it does guarantee privacy in
this setting under the reasonable assumption that $\theta \leq 0$ (otherwise, privacy is impossible, due to the fact that the constraint is violated even before any queries are made).

---

**Algorithm 4: Online Greedy Algorithm**

**Input:** A set $D$ of individuals in the Beacon, previous queries $Q$, history of which queries have been flipped previously $\delta$, the incoming query $q$, and threshold $\theta$.

**Output:** Decision whether or not to flip query $q$.

```python
if $\exists i \in D : L_i(Q \cup q, x, \delta) < \theta$ then
    return Yes
end
return No
```

---

**Proposition 6.3.3.** Suppose that $\theta \leq 0$. Then Online Greedy Algorithm guarantees privacy against fixed-threshold attacks in the online authenticated access setting.

**Proof.** We prove this by induction. For base case, note that privacy is guaranteed at $t = 0$ since $L_i(\emptyset, x, \delta) = 0 \geq \theta$ for all $i$ and $\theta \leq 0$. Next, suppose that $L_i(Q_{t-1}, x, \delta_{t-1}) \geq \theta$. If $L_i(Q_t, x, \delta_{t-1}) \geq \theta$, that is, we need not flip the response to the current query $q_t$, privacy is not violated at time $t$.

Suppose that $L_i(Q_t, x, \delta_{t-1}) < \theta$, which means that we flip the response to query $q_t$ in the Online Greedy Algorithm. Let $\delta_{j,t} = \delta_{j,t-1}$ for all $j \neq q_t$ and $\delta_{j,t} = 1$ for $j = q_t$. Then

$$L_i(Q_t, x, \delta_t) = \sum_{j \in Q_{t-1}} \Delta_{ij}\delta_{j,t} + \eta_i(Q_t)$$

$$= \sum_{j \in Q_{t-1}} \Delta_{ij}\delta_{j,t-1} + \eta_i(Q_{t-1}) + \eta_i(q_t) + \Delta_{i,q_t}$$

$$= L_i(Q_{t-1}, x, \delta_{t-1}) + \eta_i(q_t) + \Delta_{i,q_t}.$$ 

Now, $\eta_i(q_t) = d_{i,q_t}(x_{q_t}A_{q_t} + (1 - x_{q_t})B_{q_t})$, while $\Delta_{i,q_t} = d_{i,q_t}(B_{q_t} - A_{q_t})$. Moreover, recall that if $\gamma < 0.25$, $B_j > 0$ and $A_j < 0$ for all queries $j$. Since $L_i(Q_t, x, \delta_{t-1}) < \theta$, it must be that $d_{i,q_t} = 1$, since otherwise $\eta_i(q_t) = 0$, and $x_{q_t} = 1$, since otherwise $\eta_i(q_t) > 0$. Thus, $\eta_i(q_t) = A_{q_t}$ and $\Delta_{i,q_t} = B_{q_t} - A_{q_t}$. Consequently, $\eta_i(q_t) + \Delta_{i,q_t} = A_{q_t} + B_{q_t} - A_{q_t} = B_{q_t} > 0 \geq \theta$. Since this holds for every individual $i$ and time $t$, privacy against threshold attacks is guaranteed for all individuals and query sequences. \qed
Adaptive Attacks

As before, adaptive attacks complicate things considerably, but we can nevertheless leverage the algorithmic idea developed for fixed-threshold attacks. In the case of adaptive attacks, recall that the privacy condition for each \( i \in B \) becomes

\[
\sum_{j \in Q_{t,1}} \Delta_{ij}^{(K)} \delta_j + \eta_i(Q_t) \geq \eta^{(K)}(Q_t),
\]

where we now make the dependence of \( \eta_i(Q_t) \) and \( \eta^{(K)}(Q_t) \) explicit. Note that we can still apply the Online Greedy Algorithm above, but with an important change: now, both \( \eta_i(Q_t) \) and \( \eta^{(K)}(Q_t) \) must be updated after receiving each query \( q \). Modulo this change, the algorithm, upon observing a query \( q \), checks whether \( \exists i \in D \) such that \( \sum_{j \in Q_{t,1}} \Delta_{ij}^{(K)} \delta_j + \eta_i(Q_t) < \eta^{(K)}(Q_t) \), flips \( q \) if this is true, and does not otherwise.

The crucial issue, however, is that we can no longer guarantee privacy in this setting, since flipping a query \( q \) may now actually cause the privacy condition for some other individual to be violated. However, in our experiments below we show that our populations remain well-mixed in terms of LRT statistics (for which the adaptive privacy condition is a proxy).

6.3.3 Unauthenticated Access

The key distinction between authenticated and unauthenticated access in our model is that in the latter case the Beacon does not know which queries have previously been made when it receives a new query \( q \) at any given point in time. We, therefore, model this setting by assuming that the query sequence (besides \( q \)) is adversarial. Specifically, the privacy constraint now takes the following form:

\[
\forall i, t, \min_{Q_i \subseteq Q_{t-1}} L_i(Q_i \cup q_t, x, \delta_t) - \theta(Q_i \cup q_t) \geq 0. \tag{6.7}
\]

We use \( Q_t \) to emphasize that since we do not know the past query sequence and wish to protect the privacy of every \( i \in D \), we are assuming that the sequence of queries is independently adversarial.
for each $i$. We now show that in the unauthenticated setting, the temporal aspect collapses, and the optimal decision about which queries to flip can be made at time $t = 0$.

**Proposition 6.3.4.** *In the unauthenticated access setting, if all $j \in S$ are queried by some finite time $t$ and there exists a solution to the Beacon-Privacy-Problem, then there is an optimal solution with the property that $F = F_0$ and $F_t = \emptyset$ for all $t > 0$.*

*Proof Sketch.* For a sufficiently large $t$, $Q_{t-1} = S$. As such, $Q_t \cup q_t = Q_t$. Then, it must be true that $F_t = \emptyset$ and $F = \cup_{t' < t} F_t$. Since $F$ must guarantee privacy for all queries at time $t' \geq t$, it must be a minimal set of queries to do so, and we can simply identify such a set at $t = 0$. □

This means that the unauthenticated online access setting is effectively a worst-case batch setting, where the worst-case set of queries is chosen independently for each $i \in D$. Note that this proposition appears to contradict Proposition 6.3.1, but in fact, it does not, as neither claims that the optimal solution it characterizes is unique. In this case, too, we can wait to implement the flips in $F$ until the associated queries are actually observed for the first time.

The consequence of Proposition 6.3.4 is that we can simplify somewhat the definition of privacy in the unauthenticated setting:

$$\forall i, \min_{Q_i \subseteq S} L_i(Q_i, x, \delta) - \theta(Q_i) \geq 0. \quad (6.8)$$

**Fixed-Threshold Attacks**

Recall that $P_i(Q) = \{ j \in Q_i | d_{ij} = 1 \}$. While we previously omitted the dependence of $P_i$ on $Q$, this must be explicit in the online setting. The next proposition shows that in the case of fixed-threshold attacks, the privacy condition reduces to a particularly simple form.

**Proposition 6.3.5.** *In the unauthenticated access setting with fixed-threshold attacks, the privacy condition (6.8) is equivalent to*

$$\forall i, \sum_{j \in P_i(S) \setminus F} A_j \geq \theta. \quad (6.9)$$
Proof. Fix \( i \in D \). We begin by unpacking the LRT score resulting from a flipping strategy \( \gamma \) in Equation (6.8) (since \( \theta \) is fixed, that is the only thing affected by the choices of queries):

\[
\min_{Q_i \subseteq S} L_i(Q_i, x, \delta) = \sum_{j \in Q_{i,1}} \Delta_{ij} \delta_j + \sum_{j \in Q_{i,1}} d_{ij} A_j + \sum_{j \in Q_{i,0}} d_{ij} B_j.
\]

First, observe that since \( B_j > 0 \) by our assumption that \( \gamma < 0.25 \) and from (the proof of the) Proposition 6.1.1, the query set \( S_0 \) (i.e., those with \( x_j = 0 \)) would not be included, since they can only increase the LRT statistic. Similarly, none of the queries with \( d_{ij} = 0 \) will be included since these do not contribute to the LRT statistic. Consequently, \( Q_i \subseteq P_i(S) \). Moreover, since \( B_j > A_j \) under the same assumptions, none of the terms with \( \delta_j = 1 \) are included. Consequently, \( Q_i \subseteq P_i(S) \setminus F \). Moreover, since \( A_j < 0 \), all queries in \( P_i(S) \setminus F \) will be included. \( \square \)

An important implication of Proposition 6.3.5 is that in this setting flipping queries is equivalent to masking them. The reason is that since flipping increases LRT statistics, the worst-case subset of queries will never include any queries that have been flipped, effectively masking all of them.

As a consequence of Proposition 6.3.5, we can represent the solution to the Beacon-Privacy-Problem in this setting as the following integer linear program:

\[
\min_{\delta} \sum_j \delta_j \quad \text{subject to:} \quad \sum_{j \in P_i} |A_j| \delta_j \geq \theta - \sum_{j \in P_i} A_j, \quad (6.10)
\]

where \( |A_j| \) refers to the absolute value of \( A_j \). Moreover, in the special case that the AAFs follow the beta distribution and we use their expectations, we can make direct use of the methods from Section 6.2.1, including the Greedy \( k \)-Cover algorithm (with \( k_i = \theta + |P_i| \), where \( |P_i| \) is the size of the set \( P_i \) (slightly overloading notation). Similarly, even in the general case, we can leverage the heuristic algorithm in Section 6.2.1, replacing \( \Delta_j \) with \( |A_j| \). Finally, we observe that in the special case \( \theta = 0 \), there is only one feasible solution, which is \( F = \bigcup_i P_i \). On the other hand, this solution is always feasible (but not necessarily optimal) if \( \theta \leq 0 \), and we can thus always guarantee privacy in such a setting for fixed-threshold attacks.
Adaptive Attacks

Recall that, even in the batch setting, since $\Delta_{ij}^{(K)}$ may be negative for some $i, j$, privacy constraints may be violated for some individuals if we flip certain queries $j$. Since in the unauthenticated setting, we are making decisions up-front, we only consider the subset of queries $j$ for which $\Delta_{ij}^{(K)} \geq 0$ for all $i$.

Unpacking the condition in Equation (6.8) and rearranging terms, we obtain the following privacy condition for adaptive attacks for each individual $i \in D$:

$$\min_{Q_{i,1} \subseteq S_1} \sum_{j \in Q_{i,1}} \left( \Delta_{ij}^{(K)} \delta_j + d_{ij}^{(K)} A_j \right) + \min_{Q_{i,0} \subseteq S_0} d_{ij}^{(K)} B_j \geq 0,$$

where $d_{ij}^{(K)} = d_{ij} - \sum_{k \in D(K)} \frac{d_{kj}}{K}$. Since the second term on the left-hand side doesn’t depend on $y$ (equivalently, $F$), we can pre-compute it, setting $k_i = -\min_{Q_{i,0}} d_{ij}^{(K)} B_j$. Consequently, we obtain the condition

$$\min_{Q_{i,1}} \sum_{j \in Q_{i,1}} \left( \Delta_{ij}^{(K)} \delta_j + d_{ij}^{(K)} A_j \right) \geq k_i.$$

We now use this expression to obtain a variant of the MI Greedy heuristic for this setting. The key idea behind this heuristic was to choose a query $j$ to flip that has the highest average marginal impact in each iteration (omitting individuals previously “covered” in the sense that their privacy is satisfied). Since we only consider flipping queries with $\Delta_{ij}^{(K)} \geq 0$, this will not have a detrimental impact on any such ”covered” individuals, as it can only increase their LRT statistics. For any $i$ not yet covered, define $\mu_{ij}$ to be the marginal impact of flipping a query $j$. If $\Delta_{ij}^{(K)} + d_{ij}^{(K)} A_j \geq 0$, this query will be omitted as a result of the flip, and the marginal contribution is thus $\mu_{ij} = |d_{ij}^{(K)} A_j|$. If, on the other hand, $\Delta_{ij}^{(K)} + d_{ij}^{(K)} A_j < 0$, this query will remain, but its contribution reduced by $\Delta_{ij}^{(K)}$ and the marginal contribution is therefore $\mu_{ij} = \Delta_{ij}^{(K)}$ as in the batch setting.
6.4 Experiments

6.4.1 Experiment Setup

Dataset

The dataset used in this paper was originally made available by the organizers of the 2016 iDash Privacy and Security Workshop [70] as part of their Practical Protection of Genomic Data Sharing through Beacon Services challenge. The goal of the challenge was for teams to develop computational approaches that release as many truthful responses as possible through a modified beacon before the Shringarpure-Bustamante attack [24] could be used to re-identify an individual. In this study, we use SNVs from Chromosome 10 for a subset of 400 individuals to construct the beacon, and another 400 individuals excluded from the beacon. Unless otherwise specified, we set the genomic sequencing error rate to $\gamma = 10^{-6}$, as in the iDash challenge.

Computational Environment

Experiments were carried out on a PC with an AMD Ryzen 7 3800x processor and 64 GB DDR4 3600 MHz-CL19 RAM running Ubuntu version 18.04.5, using Python version 3.6.12.

Baselines

We compare our approaches to three state-of-the-art baselines. The first is strategic flipping (SF) [69], which is the winning entry to the 2016 iDash Privacy Challenge, and uses a combination of greedy and local search. We compare two versions of this approach: first, the version as previously implemented (SF), and second, a variant, SF-M in the adaptive settings which uses our definition of privacy instead of setting a static threshold using a maximum allowable false positive rate. Note that SF and SF-M are equivalent in the fixed-threshold setting. The second baseline is random flipping (RF) proposed by Raisaro et al. [42]. RF randomly flips a subset of unique alleles in the Beacon dataset by sampling from a binomial distribution. The third baseline we consider is differential
privacy (DP) as proposed for this setting by Cho et al. [63]. These baselines are configured so that in the fixed-threshold batch setting they maximize utility within their respective parameter configuration space while guaranteeing privacy, which is defined as the fraction of individuals in the beacon for whom the privacy constraint under the respective threat model is satisfied. As privacy protection for all individuals was not always guaranteed by the baseline methods in the adaptive threshold case, we present two sets of results: 1) each baseline is tuned to be the best achievable privacy, and then additional results are presented for parameters that are less conservative in order to understand the relative impact on utility, and 2) tuning all methods to a similar level of utility to the best extent possible, and comparing the relative privacy achieved using ROC curves.

**Additional Adaptive Attacks**

While one of our threat models is explicitly adaptive, it does not consider the full scope of attack adaptivity that can be leveraged. We, therefore, consider two additional adaptive attacks in our evaluation: a) *allele-inference attack*, which attempts to infer which queries were flipped by considering cross-SNV correlations as in prior work [44, 116] and b) *mimicry attack*, which attempts to infer which queries were flipped by simulating our defense on synthetic data samples.

In the *allele-inference attack*, an adversary leverages correlations among SNVs to infer which SNVs might have been flipped by our approaches. A common measure of correlations between SNVs that is leveraged in past variations of this attack is Linkage Disequilibrium (LD) [117]. A bi-allelic SNV is a position on the genome for which two possible alleles are seen across the population. For a pair of bi-allelic SNVs (with alleles \{A,a\} and \{B,b\} respectively), the linkage disequilibrium is defined by:

\[
LD = P(AB) - P(A)P(B)
\]

(6.11)

where \(P(AB)\) is the frequency with which \(A\) and \(B\) occur together, and \(P(A)\) and \(P(B)\) are the individual allele frequencies for \(A\) and \(B\) respectively. The value of \(LD\) lies between \(-0.25\) and \(0.25\), with a positive value indicating a higher-than-random association of the two alleles. We note
that all SNVs considered in this study are bi-allelic. The allele-inference attack proceeds as follows. Upon receiving a ‘no’ response from the beacon for an SNV $j$, the attacker calculates LD values for SNV $j$ paired with $l$ neighboring SNVs on either side of $j$ on the genomic sequence, based on the fact that alleles close to each other on the genomic sequence are more likely to be correlated [118]. If the maximum LD thus calculated for any pair $(i, j)$ lies above a certain threshold $t_{LD}$, and the beacon response for SNV $i$ is ‘yes’, the attacker flips the beacon response for SNV $j$ to ‘yes’. In our experiments, we use $l = 5$ and $t_{LD} = 0.2$, which limits the impact of the attack to highly-correlated pairs. In the online setting, we assume that upon receiving a ‘no’ response for an SNV $j$, the attacker subsequently queries neighbors in order to calculate LD, if they have not already been queried.

In the mimicry attack, the attacker attempts to infer the SNVs that might have been flipped by the proposed approaches by simulating them either on 1) similar public datasets or 2) synthetic datasets constructed using knowledge of alternate allele frequencies (AAF). To simulate this, we generate ten synthetic populations, and corresponding simulated beacons using the AAFs from our original iDash dataset. Each synthetic population also consists of 400 individuals in the beacon, and 400 individuals not in the beacon. To construct the synthetic populations, we generate genomic sequences where SNVs are considered to be independent, and a minor allele exists at position $j$ for each individual with probability equal to the AAF for the $j^{th}$ position from the iDash dataset. The attack proceeds as follows. Our privacy-preserving algorithms corresponding to each threat model described above are used on these ten synthetic datasets to obtain a probability value $p_j$ that SNV $j$ will be flipped. In the batch setting, the attacker proceeds to flip beacon responses for the top $R$ queries in terms of $p_j$ from ‘no’ to ‘yes’ if the beacon response is indeed ‘no’ to begin with. In our experiments, we evaluate the effect of flipping the top 5% and top 10% of query responses in terms of $p_j$. In the online setting, as queries are made one at a time, computing the top $R$ queries in terms of $p_j$ is not possible until all SNVs are queried. Instead, the attacker flips the beacon response for an SNV $j$ if the original beacon response is ‘no’ and $p_j$ is above a chosen threshold $t_p$. We evaluate the effect of this attack for $t_p$ set to 0.5, 0.7 and 0.9.
6.4.2 The Batch Setting

Fixed-Threshold Attacks with a Small Sequencing Error

In our first set of experiments, we consider a setting where the sequencing error is negligible. *In these experiments only*, we set \( \gamma = 10^{-240} \), which is sufficiently small that it suffices to flip a single beacon response per individual to guarantee privacy in this setting. While a sequencing error this small is unrealistic in modern genomic sequencing, the associated results offer an instructive starting point. In this setting, because flipping queries when the minor allele is very frequent is likely to degrade the trust in the system, we consider the effect of a restriction on the rarity of occurrence of the alternate allele on the number of flips needed to secure privacy.

![Graph](image)

Figure 6.1: Number of SNVs flipped to guarantee privacy in a Beacon compared to baselines (\( n = 400 \) individuals).

Fig. 6.1A compares the number of flipped queries between the proposed Greedy Min Beacon Cover (GMBC) and the three baselines. We can see that GMBC allows us to guarantee privacy with significantly (more than an order of magnitude) fewer false beacon responses compared to the baselines. The suboptimality of strategic flipping stems from not accounting for how many individuals an SNV affects, and only looking at the average over the population, a limitation that GMBC overcomes.
**Fixed-Threshold Attacks with AAFs Drawn from Beta Distribution**

Next, we consider the setting with a static prediction threshold $\theta$, where the alternate allele frequencies (AAFs) are assumed to be drawn from a Beta-distribution. Recall that in this setting it suffices to flip $k_i$ SNVs per individual. Once again, we present results comparing to the three baselines, now varying the value of $\theta$. From here on, we limit ourselves to experiments where there are no restrictions on how frequently an alternate allele can be present in a population, due to the much larger compute times needed to handle multiple high-precision values arising from the AAFs for over 1.3 million SNVs. Henceforth, we also set $\gamma = 10^{-6}$. Fig. 6.1B presents the results comparing the proposed GKC approach to the baselines for $\theta \in [-2000, 2000]$. Again, we see that the proposed GKC algorithm again flips orders of magnitude fewer SNVs compared to the alternatives, while guaranteeing privacy.

**General Case - True AAFs**

![Figure 6.2: Comparing utility and LRT scores in the fixed threshold batch setting with general AAFs.](image)

(A) Utility comparison in the general AAF setting.

(B) LRT scores of individuals in and not in the Beacon.

Next, we look at the more general case with no assumptions on AAFs, a realistic $\gamma = 10^{-6}$, and an adversary who computes a static threshold $\theta$ based on some prior knowledge. This setting is much more representative of a real-world attack. Fig. 6.2A compares the number of queries
flipped by MI Greedy (MIG) to the baselines, over a range of prediction thresholds $\theta$. In this figure, also note that for the differential privacy and random flipping baselines, for each value of $\theta$, we present the performance with an empirically selected parameter ($\epsilon$ and $p$ respectively) which yields the highest utility while preserving privacy for all individuals, with the corresponding parameter denoted in the plot. We can observe that MIG again outperforms both RF and DP by several orders of magnitude in terms of utility (all approaches preserve privacy for all individuals in the beacon dataset). SF is closer to MIG, but still flips considerably more queries.

**Adaptive Attacks**

While the algorithms devised with a fixed $\theta$ in mind guarantee privacy given this assumption, adaptive attackers can defeat these approaches by taking the revised Beacon queries explicitly into account when determining the threshold. This is illustrated in Fig. 6.2B, which shows the LRT scores for 400 individuals in the beacon, and 400 others not in the beacon after flipping responses using MI-Greedy for $\theta = 0$. While the LRT scores for the individuals in the Beacon do end up above 0, they also remain below the scores computed on those not in the Beacon. Although the attacker does not in fact know who is in the Beacon, a clustering attack can separate the two populations, albeit with a high false positive rate. Specifically, using 1-dimensional $k$-means clustering of the LRT scores achieves 100% true positive rate at the somewhat substantial cost of a 30% false positive rate, averaged over 20 runs. Next, we evaluate the effectiveness of the proposed algorithms that aim to explicitly account for this more sophisticated attack.

Fig. 6.3A shows the number of flips that need to be flipped by our *adaptive attack* variant of MIG, as well as by the various baselines in this setting. The values of $\epsilon$ and $p$ used for DP and RF respectively are shown in parentheses in the plot legend. MIG again flips orders of magnitude fewer SNVs than either RF or DP, as well as SF-M for higher values of $K$ (the size of the non-Beacon LRT comparison group discussed in Section 6.2.2). SF is more competitive and actually flips fewer SNVs than MIG when $K$ is higher, but, as we will see presently, it offers very poor privacy in this setting.
Fig. 6.3B shows privacy (as a function of $K$ with respect to the definition of privacy in Section 6.2.2). While MIG and SF-M preserve privacy for all individuals in the Beacon in this setting, both DP (for $\epsilon \in \{0.1, 0.5, 1\}$) and SF fail to achieve privacy for all individuals. Fig. 6.3C illustrates that explicitly accounting for adaptive attacks, MIG yields LRT scores that are much more mixed between individuals in and not in the Beacon dataset than if we are to assume fixed-threshold attacks. Quantitatively, setting $K = 20$ increases the false positive rate for the clustering attack from 30% to over 50% on average.

**ROC Curves for Adaptive Attack, Batch Setting**

Fig. 6.4A presents ROC curves for the adaptive attack in the batch setting. Note that in this case, a lower area under the curve is better, as the ROC curve corresponds to attack success. Also note that the maximum false positive rate up until which the area under the ROC curve (AUC) remains zero for MIG and SF-M corresponds to the percentage of the total population for which the solution is computed (20% of individuals not in the beacon corresponds to 10% of the total population), beyond which the AUC is non-zero for all approaches (the plot line corresponding to SF-M has a slight non-zero slope between FPR=0.1 and FPR=0.2 in Fig. 6.4A).

Fig. 6.4B presents ROC curves for the adaptive batch setting, comparing MIG to the various
baselines when all methods are tuned to have similar utility in terms of the number of SNVs flipped. Parameters for RF and DP were chosen using the expected utility, computed over an average of 5 runs. Here, we see MIG significantly outperforms all baselines - illustrating that in order to achieve a similar level of privacy to MIG, the baselines must suffer a much greater utility loss.

6.4.3 The Online Setting

Authenticated Access

Recall that in the authenticated setting, the defender has access to each user’s query history, and thus a decision about whether to flip the beacon response for an SNV can be greedily made at runtime. Unlike previous settings where all methods were able to achieve perfect privacy for all individuals in the beacon, this will no longer always be the case in the online setting. Consequently, we also compare our methods with the baselines in terms of privacy, defined as the fraction of the individuals in the Beacon whose privacy is not violated.

First, we consider fixed-threshold attacks. Fig. 6.5 compares Online Greedy (OG) with the baselines when $\theta = 0$ (the worst case threshold in the online setting). Recall that OG provably
achieves privacy in such settings, but does flip more queries than SF (but far fewer than other baselines). In contrast, SF compromises privacy for a considerable fraction of individuals in the Beacon (as do other baselines) when few SNVs have been queried. In Fig. 6.6, we consider adaptive attacks. When $K = 1$, OG tends to have better privacy, but considerably lower utility than SF and SF-M (and dominates the other two baselines in both). For $K = 10$, however, OG has better utility than all baselines but SF, but slightly lower privacy than SF-M (and better than others). While SF achieves the highest utility, it has extremely poor privacy.

**ROC Curves for Adaptive Attack, Authenticated Online Setting**

We compare the performance of the various methods using ROC curves in the authenticated online setting with the adaptive threshold model. Figs. 6.7A, 6.7B and 6.7C present ROC curves when 100000, 500000 and 1.3 million SNVs are queried respectively, using $K = 1$, while Figs. 6.7D, 6.7E and 6.7F present corresponding results for $K = 10$.

When $K = 1$, OG outperforms all baselines except DP with $\epsilon = 0.1$ when 100000 SNVs are queried, and DP with $\epsilon = 0.5$ and $\epsilon = 0.1$, when the number of SNVs queried is increased to 500000 and 1.3 million. However, DP flips a significantly larger number of SNVs when compared to MIG, as can be observed in Fig. 6.6A. Also note that at $K = 1$, DP does violate privacy more often than
Figure 6.6: Comparing utility and privacy in the authenticated online setting with adaptive attacks.

MIG, as can be seen in Fig. 6.6C. When $K = 10$, OG outperforms all baselines, except DP with $\varepsilon = 0.1$, although the performance of the two approaches is closer in this case when compared to the case where $K = 1$. The performance of SF-M also shows significant improvement relative to the $K = 1$ case.

In contrast, Fig. 6.8 presents results in the authenticated online setting, where all methods are tuned to have similar utility. Here, as opposed to the results in Fig. 6.7, we observe that for both $K=1$ and $K=10$, OG significantly outperforms all baselines, achieving far greater privacy for the same utility loss.
Unauthenticated Access

Finally, we compare the proposed approaches to baselines in the unauthenticated online setting. Once again, we begin with fixed-threshold attacks. As shown in Fig. 6.9, OMIG (our algorithm variant for this setting) flips more queries than SF, but does guarantee privacy, whereas SF compromises the privacy of a subset of individuals, particularly as more SNVs can be queried. The other two baselines also achieve privacy but at a considerable loss in utility compared to OMIG.

Fig. 6.10 presents a similar comparison for adaptive attacks. In this setting, all methods including OMIG now compromise privacy, with DP doing so the least. However, OMIG is now again orders of magnitude better than most of the baselines in terms of utility (with SF-M now performing relatively poorly in terms of both utility and privacy). While SF performs similarly to OMIG in

Figure 6.7: ROC curves, authenticated online setting.
terms of privacy, it has slightly lower utility.

**ROC Curves for Adaptive Attack, Unauthenticated Online Setting**

Fig. 6.11 presents ROC curves comparing the performance of the various baselines to our variant (OMIG) in the unauthenticated online setting with an adaptive threshold attack, when 100000, 500000 and 1.3 million SNVs are queried. In all three cases, DP outperforms OMIG, and SF achieves very similar performance to OMIG. However, both DP and SF provide lower utility when compared to OMIG, as can be seen from Fig. 6.10. In this setting, RF is seen to perform better for lower values of the probability $p$, in contrast to other settings.
6.4.4 Allele-Inference Attack

Next, we evaluate the impact of the allele-inference attack on the proposed methods as well as the different baselines. Fig. 6.13A presents results in the fixed threshold batch setting, where all methods originally guaranteed privacy. It can be seen that the attack has very little impact on any of the methods, with privacy remaining above 94% in all cases. Empirically selected best parameters (in terms of utility while guaranteeing privacy for all) used for DP and RF for the various values of \( \theta \) are highlighted in the plot in the corresponding color.

Fig. 6.13B presents results for the adaptive threshold batch setting. The attack has no impact on
All methods tuned for similar utility. For SF, SFM and MIG, we present original performance as well as performance after the allele-inference attack, with plot lines corresponding to the latter denoted by (LD) in the legend. Yet again, the impact of the allele-inference attack is minimal for all methods, including the proposed approach.

Finally, we note that the allele-inference attack had no noticeable impact on any of the methods in the online settings. In the case of our online greedy algorithm, this can be attributed to the fact that the SNVs flipped in the online setting are different from the ones flipped in the batch setting. The decision to flip each SNV depends on the query sequence in case of authenticated access. In the case of unauthenticated access, the SNVs flipped do not have enough highly correlated neighbors to
impact privacy. The effect of allele inference on the different baselines is negligible when averaged over multiple random query sequences in the online settings.

### 6.4.5 Mimicry Attack

Finally, we present the impact of mimicry attacks using simulated datasets, starting with the batch setting using a fixed threshold threat model. Fig. 6.14A presents a privacy comparison in this setting, when the top 5% and 10% of query responses from the synthetic beacons in terms of flipping probability $p_j$ are flipped to ‘yes’. Much like allele inference, the mimicry attack has very little impact on any of the methods, with privacy remaining above 96%.
Fig. 6.14B similarly compares privacy achieved by the various methods after being subjected to the mimicry attack, when the top 5% and 10% of query responses are flipped. Only the proposed approach is affected, and while the associated reduction in privacy is now tangible, it is still quite small, with MIG remaining comparable to DP in terms of privacy, and competitive with SF-M.

Next, we present results for the online setting with authenticated access. Recall that in the online setting, the attack flips the beacon response for SNV $j$ to ‘yes’ if the original response is ‘no’ and the probability $p_j$ of SNV $j$ being flipped computed over the synthetic data is above threshold $t_p$.

For the fixed-threshold threat model, the attack is seen to have no impact on performance. Fig. 6.15 presents privacy comparison across the various methods for $t_p$ set to 0.5, 0.7 and 0.9 for the adaptive threshold setting when $K = 1$ and $K = 10$. Compared to the privacy originally achieved by these methods as presented in Fig. 6.6, we observe that the mimicry attack has a nominal impact on SF and RF, but other methods (including our approach) are essentially unaffected. The performance does not vary significantly across the various values of $t_p$. Finally, we note that the attack has no
impact on performance in the unauthenticated online setting.

### 6.5 Conclusion

In this chapter, we presented a novel framework for the privacy-preserving design of Beacons in the context of membership inference attacks leveraging a likelihood-ratio-test statistic. Our framework precisely dissects the many ways in which the Beacon service can be configured and used, such as allowing queries as a batch or in a sequence, allowing authenticated access to individuals whose identities can be verified, or simply opening the service to the public. We also consider two distinct threat models, one of which has been explicitly studied in prior literature, while the second involves a stronger adaptive attack and has not been formally defined or analyzed in prior work. We present polynomial-time, highly scalable, algorithms that exhibit privacy guarantees for some of these instantiations of our model, and in one special case, a provable approximation of optimal utility (while guaranteeing privacy). Moreover, the proposed algorithms typically outperform prior art in either privacy (against LRT-based membership inference attacks), utility, or both. Finally, we also show that our approach is largely unaffected by allele-inference and mimicry attacks which try to infer which SNVs are flipped to preserve privacy.

Our approach has several limitations. First, our privacy model is specific to the Beacon service, and the LRT-based attack; it is possible that other attacks can be devised that can defeat our approach, although we are not aware of any existing attacks that do. Second, flipping query responses, while common in prior art, is not always a viable means to protect the Beacon service (for example, it may degrade public trust in the service). An alternative framework of masking a subset of SNV queries may offer another practical solution without this limitation, which we explore in the next chapter, along with data sharing in the form of alternate allele frequencies. Finally, a far greater challenge is to determine practically plausible adversaries, in order to parameterize potential solutions for realistic adversarial models.
Chapter 7: Enabling Privacy-Utility Tradeoffs in Genomic Beacons and Summary Statistics

In Chapter 6, we studied the problem of protecting privacy for all individuals in a genomic data Beacon, demonstrating equivalence to the Min-Set-Cover Problem under certain assumptions, and leveraging problem structure to develop highly scalable greedy solutions. In this chapter we extend this line of work to algorithms that allow the data custodian to specify a desired privacy-utility balance exogenously, using tunable parameters for both Beacon services and alternate allele frequencies (AAF). We also extend the methods proposed to be able to apply more than one form of modification to the data release, namely noise injection and selective suppression.

7.1 Model

7.1.1 Defending Against MI Attacks

In this chapter, we make a significant departure from existing approaches to preserving the privacy of individuals when sharing genomic summary statistics in two ways. First, we explicitly enable trade-offs between privacy (in the sense of protection from membership inference attacks) and utility (in terms of the extent of modification of the summary statistics). Second, we consider two defensive strategies to mitigate the privacy risks presented by MI attacks - namely suppression (masking) of beacon responses, and the addition of noise to query responses or allele frequencies. In the case of Beacon responses, the addition of noise takes the form of falsifying responses to queries (claiming that a particular alternate allele is not present in the dataset, when in fact, it is). This query-flipping approach is standard in much of the prior literature, with various strategies applied to select the subset of SNVs to flip [42, 63, 69, 70]. By contrast, masking Beacon responses [37]
is a less explored strategy. This is likely because it has a considerably larger impact on utility. In the case of allele frequencies, we add real-valued noise to the published frequencies, which is a hallmark of methods based on differential privacy concepts; but we differ from prior literature in that ours is the first work that combines the addition of noise with suppression of SNVs. Formally, let $M$ be the subset of SNVs that are masked or suppressed. We assume that the data recipients only observe $Q \setminus M$, rather than $Q$ and $M$ separately; consequently, the choice of $M$ does not in itself reveal information about the individuals in the dataset. Let $\delta$ denote the noise added to SNVs in $Q \setminus M$. In the case of Beacon responses, let $\delta$ be a vector, with $\delta_j = -1$ indicating that the response for SNV $j$ is flipped, and $\delta = 0$ indicating otherwise. To flip a query $j$ is to return a response $1 - x_j$ for it, whereas masking $j$ implies that it cannot be queried at all. In the case of AAF releases, let $\delta$ denote the real-valued noise added to the AAFs $x$ for SNVs in $Q \setminus M$.

The LRT score for an individual $i$ when set $M$ of SNVs is masked, and noise $\delta$ is added to the remaining release is $L(Q \setminus M, d_i, x + \delta)$. Overloading notation, we use $L_i(M, \delta)$ to refer to either data release model henceforth - only making a distinction where mathematically necessary. Finally, we can also write the prediction threshold for whether individual $i$ is in the dataset $D$ as a function of the defense, $\theta(M, \delta)$. To ensure privacy is preserved for an individual $i \in D$ is to ensure $L_i(M, \delta) - \theta(M, \delta) \geq 0$. Given $M$ and $\delta$, we define $Z(M, \delta) \subseteq D$ to be the set of individuals for whom masking SNVs in $M$ and adding noise $\delta$ to the rest preserves privacy, i.e.,

$$Z(M, \delta) = \{i \in D \mid L(M, \delta) - \theta(M, \delta) \geq 0\} \quad (7.1)$$

Our goal is to solve the following Summary Stats Privacy Problem (SSPP):

$$\min_{M, \delta} \alpha ||\delta||_1 + (1 - \alpha)|M| - w|Z(M, \delta)|, \quad (7.2)$$

where $\alpha$ denotes the relative cost of adding noise as compared to masking SNVs, and $w$ captures the relative importance of preserving privacy for individuals over the utility of the released summary statistics. Preserving privacy for all individuals often comes at a prohibitive cost to the utility of the released aggregate data, and is not always desirable. This approach allows the data custodian
to explicitly trade off privacy and utility through a combination of masking and adding noise in a systematic way.

We remark that from a mathematical standpoint, a high-magnitude noise $\delta$ is equivalent to suppression, in the sense that we obtain no useful information about the associated SNV either way. Our distinction here takes the perspective of usability: while these may be equivalent information-theoretically, they are not in the way they are perceived. Specifically, adding a very high level of noise amounts to deception, since one appears to be presenting the information, but in fact, it has no value; in contrast, masking is transparent in that no information is actually provided. For this reason, we expect that the relative cost of adding high-magnitude noise is considerably higher than the cost of suppression.

Finally, we consider two threat models, differing in the choice of the prediction threshold, $\theta(M, \delta)$. The first model involves an attacker who computes the threshold \textit{a priori}, not accounting for the defense. We also consider the \textit{adaptive} attacker model introduced in Chapter 5. In this stronger adversarial model, the attacker attempts to identify individuals in the dataset, \textit{accounting for the defense} by relying on the separation of LRT scores.

**Fixed Threshold Attacks** As discussed in Chapters 5 and 6, a naive attacker, who does not account for our attempts to defend against such attacks, chooses a threshold $\theta$, typically to balance false positive and false negative rates with respect to some synthetic ground truth dataset. A maximum false positive rate is often used to tune $\theta$ [69, 70]. This may be accomplished by simulating Beacons on other publicly available datasets, genomic data that the adversary otherwise has access to, or data synthesized using knowledge of alternate allele frequencies. The practical implication of this assumption is that we can set the threshold to be a constant, i.e. $\theta(M, \delta) = \theta$, which does not depend on the subset of SNVs masked or the noise added. As a result, $Z(M, \delta) = \{ i \in D \mid L(M, \delta) - \theta \geq 0 \}$ for this threat model. This threat model is assumed by most prior work [24, 37, 42]. Note that this model does not require the defender to know the threshold $\theta$ used by the attacker; a conservative bound will do.
**Adaptive Threshold Attacks** On the contrary, the *adaptive* attack model attempts to separate the two populations (individuals in $D$ from those who are not) using the separation between their LRT scores *after* the defense has been implemented. Recall that $\bar{D}$ is a set of reference individuals not in the dataset $D$. Let $\bar{D}^{(K)} \subset \bar{D}$ be a set of $K$ individuals in $\bar{D}$ such that they have the lowest LRT scores. Recall that in the *adaptive* attack model, the prediction threshold is calculated as:

$$\theta(M, \delta) = \frac{1}{K} \sum_{k \in \bar{D}^{(K)}} L_k(M, \delta)$$  \hspace{1cm} (7.3)

As a result, the set of individuals for whom privacy is protected, $Z(M, \delta) = \{ i \in D \mid L_i(M, \delta) - \frac{1}{K} \sum_{k \in \bar{D}^{(K)}} L_k(M, \delta) \geq 0 \}$ under the adaptive attacker model.

### 7.1.2 Additional Preliminaries

**Linkage Disequilibrium**

So far, the attack models presented are based on the assumption that SNVs are independent. However, in practice, an adversary may be able to exploit correlations between SNVs to infer Beacon responses for SNVs that are modified. The linkage disequilibrium coefficient [44] is one formal measure of such correlations. Given two loci (a locus is the position of a certain gene on a chromosome) with alleles $\{A, a\}$ and $\{B, b\}$ respectively, the linkage disequilibrium coefficient is computed as $LD = P(AB) - P(A)P(B)$. The value of the linkage disequilibrium coefficient (henceforth called LD) lies between $-0.25$ and $0.25$, where larger values are indicative of higher-than-random association of the specified alleles.

We evaluate the effect of an adversary accounting for correlations as follows. In this situation, for each SNV that is either flipped or masked, the adversary identifies SNVs that are correlated with the target SNV. Due to the fact that computing the linkage disequilibrium coefficient (LD) for all pairs of 1.3 million SNVs is very computationally expensive, we model an adversary who computes correlations with a fixed number of SNVs on either side of the SNV of interest in the dataset. We consider two SNVs to be correlated if their LD is above a certain threshold, $t_{LD}$, limiting
our evaluation to highly correlated SNV pairs. Given a flipped or masked SNV \( j \), let \( N_{LD}(j) \) be the set of SNVs that are positively correlated with \( j \). If at least 75% of SNVs in \( N_{LD}(j) \) have a \textit{yes} Beacon response, then the Beacon response for SNV \( j \) is inferred to be \textit{yes} as well. We show in a subsequent section that only SNVs with \textit{yes} responses are masked/flipped. Having inferred responses for a subset of the flipped/masked SNVs, the adversary now recomputes LRT scores and makes membership inference claims using the fixed-threshold or adaptive-threshold models as described above. Observe that this evaluation is worst-case in the sense that we only perform the above correlation attack for SNVs that \textit{we know} have been flipped or masked, rather than \textit{all} SNVs, as would be done in an actual attack.

**Differential Privacy and the Laplace Mechanism**

Differential privacy (DP) is a popular privacy-preserving data-sharing technique based on the principle that the distribution of responses computed on two datasets that only differ in one entry should be similar.

Formally, a randomized algorithm \( f \) is \( \varepsilon \)-differentially private if for any two datasets \( D \) and \( D' \) differing in one entry (i.e., \( D' \) omits one record from \( D \)), and for all possible subsets \( F \) of the image of \( f \),

\[
\frac{P(f(D)) \in F}{P(f(D') \in F)} \leq \varepsilon
\]  

(7.4)

**Unbounded Risk** In practice, the value of \( \varepsilon \) is used to trade off utility and privacy, with a smaller value of \( \varepsilon \) corresponding to a greater degree of privacy, at the cost of utility. For aggregate queries such as means over columns, as is the case with AAFs, a simple mechanism to achieve differential privacy is to add random noise sampled from a Laplace distribution to each SNV’s AAF. Laplacian noise with a scale of \( \Delta g / \varepsilon \) where \( \Delta g \) is the sensitivity of the function \( g \) (in our case, \( g \) is the mean for each SNV) satisfies the requirements for \( \varepsilon \)-differential privacy [57]. The sensitivity of a function is defined as \( \max ||g(D) - g(D')||_1 \), where \( D \) and \( D' \) differ in one entry (i.e., \( D' \) omits one individual from \( D \)). For a given SNV in a dataset of \( n \) individuals, \( x_{ij} \) can be either 1 or 0. Therefore, the
maximum possible difference between means over columns differing in one entry is $1/n$. As the dataset has $m$ SNVs, the sensitivity, which is the $\ell_1$ norm of the vector of size $n$ with each entry being $1/n$, is $\Delta g = \frac{m}{n}$. Therefore, adding Laplacian noise centered at 0, and with a scale of $m/n\epsilon$ satisfies $\epsilon$-differential privacy.

**Bounded Risk** While the above measure of sensitivity provides theoretical worst-case privacy guarantees, two genomic sequences rarely - if ever - tend to be completely dissimilar in terms of alternate allele composition. The above measure of sensitivity, owing to the large number of SNVs considered (on the order of 1.3 million), forces the user to choose a very large value of $\epsilon$ (on the order of 100K or above when no SNVs are masked) to retain sufficient practical utility. Such large values of $\epsilon$, in turn, offer lower privacy guarantees. Therefore, we also consider a measure of sensitivity in the average-case scenario, where the numerator consisting of the total number of SNVs, $m$ when calculating sensitivity is replaced by the average number of bits by which a sequence in the dataset differs from each sequence in the reference population. We refer to this as Bounded Sensitivity, and on our data, this measure is an order of magnitude smaller than worst-case sensitivity (on the order of 150K when no SNVs are masked).

### 7.2 Our Approach

We now present our approach to solving SSPP, under the two threat models discussed above. Note that the noise $\delta$ that is added to summary statistics is qualitatively different in Beacons as compared to AAF summary releases. Recall that while in the former, $\delta$ is additive noise that codifies whether or not SNVs are flipped, in the latter case of AAFs, $\delta$ is real-valued. As a result, the two scenarios yield structurally different optimization problems but follow the same general framework as outlined in the Model Section. In both cases, we combine the addition of noise with selective suppression of a subset of SNVs.

We begin by rewriting the LRT scores for individual $i$ as follows. Let $Q_1 \subseteq S$ be the subset of SNVs for which Beacon response $x_j = 1$, and $Q_0 \subseteq S$ be the subset where $x_j = 0$. Then, the LRT
score for individual $i$ can be written as

$$L(Q, d_i, x) = \sum_{j \in Q_i} d_{ij} A_j + \sum_{j \in Q_o} d_{ij} B_j$$  \hspace{1cm} (7.5)$$

where $A_j = \log \frac{1 - R_n^j}{1 - \gamma R_n^j - 1}$ and $B_j = \log \frac{R_n^j}{\gamma R_n^j}$. In the case of AAFs, let $A(x_j) = \log \frac{\bar{p}_j}{x_j}$, and $B(x_j) = \log \frac{1 - \bar{p}_j}{1 - x_j}$. Note that in this scenario, $A$ and $B$ are functions of $x_j$, instead of constants for each $j$ as in the beacon service. Then the LRT score can be rewritten as:

$$L(Q, d_i, x) = \sum_{j} d_{ij} A(x_j) + (1 - d_{ij}) B(x_j)$$  \hspace{1cm} (7.6)$$

We note that in the case of Beacons, following Chapter 6, we assume that the alternate allele is the minor allele at a given position $j$. This, in turn, allows us to leverage a bound on the genomic sequencing error $\gamma$ to ensure our solution approach never violates privacy previously achieved using an iterative process, as we shortly explain. On the other hand, for AAF releases, we only assume that AAFs are bound by $[0.0001, 0.9999]$ in order to prevent division by zero, and any SNV may be masked in the case of AAF summary releases. We showed in the last chapter that flipping Beacon responses $x_j$ from 0 to 1 is counterproductive to defending against LRT-based attacks. We now make an analogous observation for masking queries where $x_j = 0$ for both Beacons and AAF summary statistics.

**Proposition 7.2.1.** In a Beacon service, for genomic sequencing error $\gamma < 0.25$, $B_j > 0$.

**Proof.** Let $\bar{p}_j$ be the AAF for SNV $j$ in the population. Recall that $R_n^j = (1 - \bar{p}_j)^{2n}$. As $\bar{p}_j < 0.5 \forall j$, $\frac{R_n^j}{R_{n-1}^j} = (1 - \bar{p}_j)^2 \geq 0.25$. Since $\gamma < 0.25$, $\frac{R_n^j}{\gamma R_{n-1}^j} > 1$, and consequently $B_j = \log \frac{R_n^j}{\gamma R_{n-1}^j} > 0$.  \hspace{1cm} \square

**Proposition 7.2.2.** Suppose Beacon response $x_j = 0$ for SNV $j$ given Beacon dataset $D$. Then masking the SNV can never increase the LRT score for an individual $i \in D$, provided $\gamma < \frac{R_n^j}{R_{n-1}^j} \forall j$.

**Proof.** Consider SNV $j$ and an individual $i \in D$. If $d_{ij} = 0$ (i.e., the individual does not have a minor allele at position $j$), masking the SNV makes no difference to the LRT score (contribution of $j$ to LRT score is 0 when $d_{ij} = 0$, refer to Eq. 5.1 for details). However, when $d_{ij} = 1$, suppressing
Beacon response $x_j$ changes the contribution of query $j$ to the LRT score from $\log \frac{R^j_n}{\gamma R^j_{n-1}}$ to 0. Based on Proposition 7.2.1, it can be seen that $\log \frac{R^j_n}{\gamma R^j_{n-1}} > 0$. Thus, if SNV $j$ is masked the LRT score can only decrease.

### 7.2.1 Masking SNVs

We begin by considering the impact of masking a single SNV on the LRT score. Let $S$ be the set of all SNVs, and $M \subseteq S$ be the subset of SNVs masked. Let $\Delta^M_{ij}$ represent the marginal contribution of masking SNV $j$ on the LRT score for individual $i$. In the case of the Beacon service, masking an SNV $j$ changes its LRT score contribution from $d_{ij}A_j$ to 0, as can be observed from Equation 5.1. Recall that we only mask SNVs where $x_j = 1$, therefore, if the individual does not have the alternate allele (i.e., $d_{ij} = 0$), masking the SNV makes no difference. Therefore, for Beacons, $\Delta^M_{ij} = -d_{ij}A_j$.

Similarly, in the case of an AAF summary release, masking an SNV $j$ changes its LRT contribution from $d_{ij}A(x_j) + (1 - d_{ij})B(x_j)$ to 0, as we can observe from Equation 5.2. Note that in this case, $\Delta^M_{ij}$ is also a function of the AAF $x_j$, and therefore, as real-valued noise may be added to SNV $j$ as part of our approach before the SNV is masked, $\Delta^M_{ij} = -d_{ij}A(x_j + \delta_j) - (1 - d_{ij})B(x_j + \delta_j)$. On the contrary, we assume that the subsets of SNVs flipped and masked in the case of Beacons are disjoint.

### 7.2.2 Adding Noise to Statistics

Next, we consider the addition of noise to the published statistics, for SNVs that are not masked. Let $\delta$ denote additive noise. In the case of the Beacon services, let $\delta_j = -1$ indicate that SNV $j$ is flipped, i.e. the Beacon response for SNV $j$ changes from 1 to 0. Note that following the observation in Chapter 6, we only flip SNVs where initially $x_j = 1$. The marginal impact of flipping beacon response for SNV $j$ on the LRT score for individual $i$ is $\Delta^F_{ij} = d_{ij}(B_j - A_j)$, as we can observe from Equation 5.1. In the case of AAFs, we use the Laplace mechanism defined in the Preliminaries section to add real-valued noise. Thus in this case, $\delta_j \in [0, 1]$, and frequencies after the addition of
noise are clipped to ensure they are still in the range $[0.0001, 0.9999]$. In both Beacons and AAFs, the $\ell_1$ norm of $\delta$ quantifies the total amount of noise added to the summary statistics.

### 7.2.3 Fixed Threshold Attacks

We begin by presenting our solution for the fixed-threshold attack model, where privacy is said to be preserved for an individual $i$ when their LRT score calculated after suppressing set $M$ of SNVs and adding noise $\delta$ lies above a constant prediction threshold $\theta$, specified exogenously.

Let $z_i \in \{0, 1\}$ be a binary variable corresponding to individual $i$, where $z_i = 1$ when privacy is preserved for $i$, and $z_i = 0$ otherwise, and define $y_j = 1$ if SNV $j$ is masked (i.e., $j \in M$), and $y_j = 0$ otherwise. Then the following optimization problem optimally solves SSPP for the fixed-threshold attacker:

\[
\min_{\delta, y, z} \alpha ||\delta||_1 + \sum_j (1 - \alpha)y_j - w \sum_i z_i
\]

subject to:

\[
(L_i(M, \delta) - \theta)z_i \leq 0 \quad \forall \ i \in D
\]

\[
y \in \{0, 1\}^m, z \in \{0, 1\}^n, \delta \in \begin{cases} 
\{-1, 0\}^m, & \text{Beacons} \\
\mathbb{R}^{|Q \setminus M|}, & \text{AAFs}
\end{cases}
\]

In the case of Beacons, $\delta$ is an integer vector, with entries being either $-1$ or $0$, and the above optimization problem assumes the form of an integer linear program (ILP). While the ILP optimally solves SSPP, it has an exponential worst-case running time with $\mathcal{O}(3^m)$ possible solutions (each SNV in $Q_1$ can be flipped, masked, or reported truthfully) which poses significant scalability challenges with larger populations over millions of SNVs. In the case of AAFs, $\delta$ is real-valued, and thus the above problem becomes a mixed-integer program (MIP). Much like the ILP, the MIP has difficulty scaling to large problem instances with over a million SNVs. To address these limitations, we now introduce heuristic algorithms which approximately solve SSPP for Beacons and AAFs.
Heuristic - Beacons

We now introduce a simple greedy algorithm to compute approximate solutions to SSPP for Beacon services. The driving idea behind our greedy heuristic is as follows: at each iteration, we choose an SNV for which flipping or masking achieves the highest average marginal contribution per unit cost (of flipping or masking the SNV) to the LRT scores for individuals in the Beacon. For each individual, let $P_i$ be the set of SNVs for which the Beacon response $x_j = 1$, and the individual’s genome has the associated minor allele; i.e., $d_{ij} = 1$. For $j \in P_i$, $\Delta_{ij}^F$ and $\Delta_{ij}^M$ are independent of $i$. Let $\Delta_j^F = (B_j - A_j)$ and $\Delta_j^M = -A_j$. Then, $\Delta_{ij}^F = d_{ij}\Delta_j^F$ and $\Delta_{ij}^M = d_{ij}\Delta_j^M$. For a chosen query $j$, and a subset of individuals $P \subseteq B$, let $T_j = \{i \in P | j \in P_i\}$ which is the set of individuals for which $j \in P_i$. The average marginal contribution of flipping the query response to SNV $j$ per unit cost is:

$$\hat{\Delta}_j^F(P) = \frac{|T_j|\Delta_j^F}{\alpha|P|}. \quad (7.8)$$

Similarly, the average marginal contribution of masking an SNV $j$ is:

$$\hat{\Delta}_j^M(P) = \frac{|T_j|\Delta_j^M}{(1 - \alpha)|P|}. \quad (7.9)$$

At each iteration, we calculate both $\hat{\Delta}_j^F(P)$ and $\hat{\Delta}_j^M(P)$ for every SNV $j$, and either flip or mask the SNV with the highest overall contribution, depending on whether flipping or masking led to it scoring the highest. The number of individuals for whom we thereby guarantee privacy is non-decreasing through each iteration of this algorithm, since flipping or masking SNVs for which $x_j = 1$ can only increase LRT scores (see Proposition 7.2.2). Each time privacy is assured for at least one additional individual, we compare this privacy-utility point to the current best solution (as measured by the objective function in Equation 7.7), and update it if it improves the objective. We also update $P$ to be the set of individuals for whom privacy is not yet assured. The algorithm iterates until privacy is protected for all individuals in the Beacon, or we cannot flip or mask any more SNVs, at which point we return the overall best solution. This idea is formalized in Algorithm 5, which we call Soft-Privacy-Greedy-Binary (SPG-B). For runtime analysis, refer to Appendix B.1.
**Algorithm 5: Soft-Privacy-Greedy-Binary (SPG-B)**

**Input:** A set of individuals $i \in D$, subset $P_i$ and LRT score $\eta_i$ for each individual, marginal contributions of flipping/masking $\Delta_j^F$ and $\Delta_j^M$ for each SNV, a set of queries $Q$ and threshold $\theta$, weight parameter $w$, relative cost of flipping $\alpha$.

**Output:** Subset of queries $F \subseteq S$ to flip, subset of queries $M \subseteq S$ to mask.

**Initialization:** $F = \emptyset$, $M = \emptyset$, $C = \emptyset$, $F_t = \emptyset$, $M_t = \emptyset$, $U = \infty$

**while** $(D \setminus C) \neq \emptyset$ **do**

Set $l = 0$, $N = -1$, $d = -1$.

**for** $j \in (Q \setminus (F \cup M))$ **do**

Set $T_j = \{i \in (D \setminus C) | j \in P_i\}$

Set $\tilde{\Delta}_j^F = \Delta_j^F \frac{|T_j|}{\alpha|D\setminus C|}$

Set $\tilde{\Delta}_j^M = \Delta_j^M \frac{|T_j|}{(1-\alpha)|D\setminus C|}$

**if** $\tilde{\Delta}_j^F > N$ **then**

Set $N = \tilde{\Delta}_j^F$

Set $l = j$

Set $d = 0$

**if** $\tilde{\Delta}_j^M > N$ **then**

Set $N = \tilde{\Delta}_j^M$

Set $l = j$

Set $d = 1$

**if** $d == 0$ **then**

Set $F_t = F_t \cup l$

**if** $d == 1$ **then**

Set $M_t = M_t \cup l$

**for** $i \in (D \setminus C)$ **do**

**if** $\sum_{j \in F} \Delta_{ij}^F + \sum_{j \in M} \Delta_{ij}^M + \eta_i \geq \theta$ **then**

Set $C = C \cup i$

Set $U_t = \alpha|F_t| + (1-\alpha)|M_t| - w|C|$

**if** $U_t \leq U$ **then**

Set $U = U_t$

Set $F = F_t$

Set $M = M_t$

**return** $F, M$
Heuristic - AAFs

We now introduce an alternating optimization algorithm that approximately solves SSPP, combining masking of alternate allele frequencies for a subset of SNVs with adding Laplacian noise to the rest.

The outline of the algorithm is as follows. At each step, we alternate between adding noise to SNVs that have not yet been masked, such that it minimizes the objective in Equation (7.7), and masking SNVs in order of their average marginal contribution to the population’s LR scores, in an attempt to increase utility. The average marginal contribution of masking an SNV \( j \) is simply the mean over the marginal contributions for all individuals, i.e., 
\[
\Delta_j^M = \frac{1}{n} \sum_{i \in D} \Delta_{ij}^M.
\]

Since in the fixed-threshold model, we require that each individual’s score lie above a specified threshold, we aim to increase LRT scores. Thus, we rank SNVs to mask in order of their average marginal contributions. For \( \alpha \gg (1 - \alpha) \), masking is preferred over adding noise—in essence, sharing a smaller subset of cleaner data, as opposed to sharing all SNVs with high obfuscation. Masking SNVs should necessarily continue to minimize the objective function until privacy is violated for an individual previously covered as a result of encountering large positive values of \( \Delta_j \).

However, continuing to mask in this manner is suboptimal because it does not allow us to explore possible intermediate solutions; for example, masking fewer SNVs and adding slightly higher noise may provide a better privacy-utility trade-off in many cases.

Our algorithm proceeds as follows. We first add noise to all SNVs by sampling from the Laplace distribution such that it best optimizes the objective in Equation (7.7). This may be done in one of two ways: 1) computing the objective over a pre-selected set of values of \( \epsilon \), or 2) a binary search over possible values of \( \epsilon \), assuming convergence when the difference between two considered values of \( \epsilon \) in the search is sufficiently small. While the latter is more systematic, it is also slower and performs poorly (refer to Appendix B.2), while the former approach is fully parallelizable and produces good results, as long as the choices for the set of candidate \( \epsilon \) values are reasonable. We, therefore, use the first approach in the rest of this work.

Having added Laplacian noise, we then mask a set of \( t \) SNVs in the order of their average
marginal contribution to LR scores, calculated after adding noise. The value of $t$ that we use is chosen to balance computation time and the near-optimality of the solution. Specifically, a smaller value of $t$ implies a larger number of candidate solutions explored, but with a runtime inversely proportional to $t$.

At the end of this cycle, we repeat the noise-addition and masking processes in an alternating fashion, each time adding noise to the SNVs that remain unmasked with the scale of the Laplacian distribution accordingly adjusted. Algorithm 6 - which we call the Soft-Privacy-Greedy-Real (SPG-R) approach - provides full details about the implementation of our method. In Appendix B.2, we present an alternate implementation of the SPG-R algorithm which leverages problem structure to reduce redundant computations, and parallel processing in order to significantly reduce runtime.

### 7.2.4 Adaptive Threshold Attacks

In the adaptive threshold scenario, the goal is to ensure that the LRT scores of individuals in $D$ and $\bar{D}$ (i.e., those not in the dataset) remain sufficiently well-mixed. Recall from the Preliminaries section that the prediction threshold in this setting is $\theta(M, \delta) = \frac{1}{K} \sum_{k \in \bar{D}(K)} L_i(M, \delta)$, which is the average LRT score for a set of $K$ individuals in $\bar{D}$ with the lowest LRT scores. Then similar to Eq. 7.7, we can formulate this as an optimization problem in the context of adaptive attacks.

$$
\min_{\delta, y \in \{0,1\}^m, z \in \{0,1\}^n} \alpha \|\delta\|_1 + \sum_j (1 - \alpha) y_j - w \sum_i z_i
$$

subject to:

$$(L_i(M, \delta) - \theta(M, \delta))z_i \leq 0 \quad \forall \ i \in D \quad (7.10)$$

$\delta \in \{-1,0\}^m$ (Beacons); $\delta \in \mathbb{R}^m$ (AAFs)

This structure allows us to extend our algorithms used for the fixed-threshold scenario, with one change - instead of sorting SNVs by $\Delta_j^M$ or $\Delta_j^F$, we now sort the SNVs by $\Delta_j^{M(K)} = \Delta_j^M - \frac{1}{K} \sum_{k \in D(K)} \Delta_k^M$ and $\Delta_j^{F(K)} = \Delta_j^F - \frac{1}{K} \sum_{k \in D(K)} \Delta_k^F$ respectively. In the adaptive threshold model,
with Beacons, $\Delta^M_j(K)$ and $\Delta^F_j(K)$ may be negative, and may be detrimental to privacy achieved in prior iterations of our greedy algorithms. As such, masking and flipping are respectively restricted to those SNVs where these quantities are strictly positive.

### 7.2.5 Linkage Disequilibrium

To defend against an attacker who leverages correlations to infer flipped/masked SNVs in a Beacon, we introduce a direct extension to our proposed Soft-Privacy-Greedy approach. Specifically, whenever an SNV $j$ with known correlations is flipped or masked, all SNVs correlated to it (the set $N_{LD}(j)$) are also flipped or masked, respectively. To capture the corresponding utility loss while deciding which SNV to flip or mask, we modify the Soft-Privacy-Greedy algorithm as follows. For each SNV $j$, we amend the marginal contribution of flipping $j$ to $\tilde{\Delta}^F_j(P) = \frac{T_j\Delta^F_j}{|P||N_{LD}(j)|}$, and the marginal contribution of masking SNV $j$ to be $\tilde{\Delta}^M_j(P) = \frac{T_j\Delta^M_j}{(1-\alpha)|P||N_{LD}(j)|}$. The algorithm then proceeds as before, with the added condition that, any time an SNV $j$ is picked such that $N_{LD}(j) \neq \emptyset$, all SNVs in $N_{LD}(j)$ are also correspondingly flipped or masked. We refer to this modified algorithm as Soft-Privacy-Greedy-LD or SPG-LD.

### 7.3 Results

#### 7.3.1 Experimental Design

**Dataset and Metrics** Our experiments were conducted on a dataset of 1338843 SNVs on Chromosome 10, made available by the 2016 iDash workshop on Privacy and Security [70]. The data consists of genomes of 400 individuals for whom summary statistics are to be released (i.e. the set $D$), and 400 individuals who are not part of this group (the set $\bar{D}$). This dataset was derived from the 1000 Genomes Project [36] and is sufficiently large to demonstrate the scalability of our approach, while using only Chromosome 10 makes it practical to work with in terms of memory footprint. All experiments were conducted on a PC with an AMD Threadripper 3960X CPU and 128 GB
RAM, running Ubuntu 22.04. We measure utility as $100\left[1 - (\alpha \cdot ||\delta||_1 + (1 - \alpha)\cdot|M|)/m\right]\%$, where $\alpha$ is the relative cost of adding noise to masking SNVs, $M$ is the set of SNVs masked, and $m$ is the number of SNVs. We define privacy as the percentage of individuals for whom privacy is preserved under each respective attacker model, as is the standard assumption in prior literature. The same measure of privacy was thus applied to all baselines when tuning for best utility for the sake of a fair comparison.

**Runtime Analysis**

Both SPG-B and SPG-R have a worst-case time complexity of $O(m^2n)$ (for details, we refer the reader to Appendix B.1). In practice, as the number of SNVs outpaces the number of individuals by several orders of magnitude, the impact of increasing the number of individuals is negligible. In the case of SPG-R, solutions that minimize the objective function are computed for a range of candidate values of $\epsilon$. While the number of candidate values for $\epsilon$ is small compared to the number of SNVs and individuals, the serial execution of operations for each candidate $\epsilon$ adds significant runtime overhead in practice, which can be greatly reduced by leveraging parallel processing, in combination with heuristics derived from problem structure. This provided motivation for the implementation of a parallel version of SPG-R, which is used for all experiments presented in this paper. The full details of the parallel implementation are provided in Appendix B.2.

**Baselines - SPG-B**

We compare our approach (SPG-B) to four state-of-the-art baselines. The first is strategic flipping (SF), where SNVs are flipped in decreasing order of their differential discriminative power as proposed by [69], followed by a local search. We also compare to a modified version of SF which we call SFM, where the adaptive threshold definition of privacy is used when applicable. The second is random flipping (RF) [42], where unique alleles in the dataset (i.e., only one individual has the allele) are randomly flipped by sampling from a Binomial distribution. The third is a differentially private (DP) mechanism proposed by [63], which offers plausible deniability for each Beacon
response. Fourth, we compare to the marginal-impact greedy (MIG) approach from the last chapter, noting that this approach guarantees privacy for all individuals. For each baseline, we consider a variation that selects SNVs as described by the method, but masks the SNVs instead of flipping them. None of the baselines allow us to combine flipping and masking into a single strategy.

**Baselines - SPG-R**

We compare the performance of the proposed SPG-R algorithm with three baselines: 1) only adding noise using the Laplace mechanism (standard DP), 2) *Masking* only, and 3) the *Linkage* approach proposed by [37], which greedily selects a subset of SNVs in linkage equilibrium in order of utility up to the maximum allowed power of the LR test. We note that we use the parallel version of SPG-R as detailed in Appendix B.2 throughout this work.

### 7.3.2 Fixed Threshold Attacks

**SPG-B**

We begin by considering fixed-threshold attacks on Beacons, with the prediction threshold $\theta = -250$. For SPG-B, we measure performance in the utility-privacy space by varying the relative weight of privacy to utility $w \in [0.01, 10]$. For DP and RF, we vary their respective parameters (noise parameter $\epsilon$ and probability $p$, respectively). Note that when approaches rely solely on masking SNVs, the only solution for threshold $\theta$ above a certain positive value is to shut the Beacon down (responses which are initially 0 are not masked) - as the maximum LRT score attainable for an individual $i$ is $\theta \geq \sum_{j \in Q_1} d_{ij}B_j$, where $Q_1$ is the subset of SNVs for which Beacon response is 1, and $d_{ij}B_j$ is the marginal contribution of flipping the response for SNV $j$ for individual $i$. Therefore, here we show results for a negative value of $\theta$ for which a solution is guaranteed. Results for $\theta = -750$ as well as special cases where all approaches including SPG-B are restricted to either flipping or masking SNVs are similar and provided in Appendix B.3.
Figure 7.1: Utility-privacy plots for the fixed threshold attack model for Beacons, compared to baselines. A) $\theta = -250$, baselines only flip SNVs. Zoomed-in portions shown in the top two subplots. B) $\theta = -250$, baselines only mask SNVs.

Fig. 7.1 compares the performance of SPG-B to the baseline approaches when a) the baselines only flip SNVs and b) the baselines only mask SNVs, respectively - as none of the baselines allow us to use a combination of the two. The key observation is that the proposed approach Pareto dominates the baselines, as the ability to both flip and mask SNVs provides an additional level of flexibility. The improvement over both DP and RF is particularly substantial in terms of utility. MIG and SF offer slightly lower utility than our approach when the privacy of all individuals is protected, but do not permit solutions that can explicitly trade off utility for privacy. Fig. 7.1B tells a similar story, where it can be seen that the difference in utility between SPG-B and MIG increases from about 0.001% (when MIG flipped SNVs) to around 0.05% (in the masking case where $\theta = -250$). This difference in utility corresponds to tens of thousands of more SNVs masked by MIG than SPG-B (as there are over 1.3 million SNVs in the dataset and $\alpha \gg (1 - \alpha)$).

**SPG-R**

In a similar fashion, we compare the parallel variant of SPG-R to the three baselines in the fixed threshold setting. We set the prediction threshold $\theta = 0$ in these experiments, as this was found to be the threshold that best separated the two populations before the defense was applied. In contrast to Beacons, we explore a wider range of values of $\alpha$, the relative cost of masking to adding noise -
as the $\ell_1$ norm of real-valued noise $\delta$ is orders of magnitude smaller on average than it is in Beacons. We vary the relative weight of privacy to utility $w$ in the range $[0.1, 10000]$. For DP, Linkage, and Masking, we choose the privacy-utility point that best optimizes the SSPP objective function for each value of $w$, ensuring fair comparisons. All results involving random noise are averaged over 5 runs. Here, we only present results for unbounded risk; those for bounded risk are similar and provided in Appendix B.5.

We select the differential privacy noise parameter $\epsilon$ for the Laplace mechanism from $\{10K, 50K, 100K, 500K, 1M, 5M, 10M\}$. While at first, it seems like these values are very large compared to parameters used in practice (on the order of 1-10), this is explained by the fact that the datasets used in practice have nowhere near the number of variables we are dealing with. With over 1.3 million SNVs, using a smaller value of $\epsilon$ would induce a prohibitive amount of noise (refer to the Preliminaries Section for details on how noise scales with the number of SNVs $m$ and $\epsilon$) that would essentially void the system of any utility.

![Utility-privacy plots](image)

Figure 7.2: Utility-privacy plots for the fixed threshold attack model for AAF releases, compared to baselines. A) $\delta = 0, \alpha = 0.5$. B) $\delta = 0, \alpha = 0.9$.

Fig. 7.2 considers the scenario where the relative cost of adding noise to masking $\alpha = 0.5$ and $\alpha = 0.9$. When $\alpha = 0.5$, SPG-R outperforms the Linkage and Masking methods, and has comparable performance to DP. When $\alpha = 0.9$, on the other hand, Masking outperforms other methods (since it has far lower cost than adding noise), with both SPG-R and DP having comparable performance (to Masking, and one another).
7.3.3 Adaptive Threshold Attacks

Next, we consider the adaptive threshold attacker, with $K = 10$ (here, this refers to the $K$ lowest percentile in terms of LRT scores). Yet again, we present results by varying the relative weight of privacy to utility $\omega$ as before, and the respective parameters for the considered baselines.

**SPG-B**

Fig. 7.3 presents results in this setting, where we observe that again SPG-B Pareto dominates the baselines, and is comparable to MIG, but now by a much larger margin than in the fixed threshold setting. While SF has better utility than the remaining baselines, it offers very low privacy. If we restrict the baselines to masking only, Fig. 7.3B shows that SPG-B once again outperforms all baselines. The reason is evident from the plot itself - a masking-only strategy is insufficient to guarantee privacy against an adaptive-threshold attacker. Results for $K = 5$, as well as settings where all approaches including SPG-B are restricted to flipping or masking, are similar and provided in Appendix B.4.

![Figure 7.3: Utility-privacy plots for the adaptive threshold attack model for Beacons, compared to baselines. A) $K = 10$, baselines only flip SNVs. B) $K = 10$, baselines only mask SNVs.](image)

**SPG-R**

Fig. 7.4 presents results for $K = 10$, for $\alpha = 0.75$ and $\alpha = 0.9$. In contrast to the fixed threshold setting, here SPG-R dominates all baselines. When $\alpha = 0.9$, i.e., adding noise is relatively expensive, Masking produces similar performance when $K = 5$ (see Appendix B.4); however, once $K$ is
increased to 10, the problem can no longer be solved using Masking alone, and SPG-R dominates it by a significant margin. DP and Linkage offer much lower utility on average compared to only masking SNVs or using SPG-R.

Figure 7.4: Utility-privacy plots for the adaptive threshold attack model for AAF releases, compared to baselines. A) $K = 10, \alpha = 0.75$. B) $K = 10, \alpha = 0.9$.

### 7.3.4 Linkage Disequilibrium

Finally, we consider attacks on Beacons which leverage correlations between SNVs. We assume that two SNVs are correlated if their LD coefficient is above 0.2. LD is measured within a span of 250 SNVs on either side of each target SNV. The attack was found to have no impact on DP and RF, so these are omitted from the following plots. Fig. 7.5 presents results in the fixed-threshold setting. Fig. 7.5A shows that the attack has a small impact on the privacy of MIG, and a significant impact on SF and SPG-B - limiting SF to $77 - 80\%$, and SPG-B to around $75\%$ when baselines flip SNVs for prediction threshold $\theta = 1000$. When baselines mask SNVs, the correlation attack has no impact on SF and MIG but reduces the maximum privacy achieved by SPG-B to around $73\%$ when $\theta = -250$. In both cases, SPG-LD successfully defends against the correlation attack, achieving $100\%$ privacy for large values of $w$, while dominating the baselines.

Fig. 7.6 presents results in the adaptive threshold case, when $K = 5$ and baselines flip SNVs. As before, the attack affects SPG-B, SF and MIG, though to a greater extent in this setting when compared to the fixed threshold model. The privacy achieved by SF drops to about $22\%$ and that of MIG drops to around $80\%$. The privacy achieved by SFM is unaffected by the attack, however, SFM yields much lower utility than our proposed methods. The privacy achieved by SPG-B is reduced
Figure 7.5: Fixed threshold attack model, where the attacker leverages correlation data, and baselines only flip SNVs. A) $\theta = 1000$, baselines only flip SNVs. B) $\theta = -250$, baselines only mask SNVs.

to about 63% when the attacker uses correlations. The modified approach, SPG-LD, dominates all approaches in terms of utility. In addition, it raises the privacy to about 88%. However, it fails to achieve privacy for all individuals - even with very large values of $w$. None of the baselines preserve privacy for any individuals solely by masking SNVs. Therefore, the performance of SPG-LD versus SPG-B is the same as in Fig. 7.6, such that we do not present new results for the adaptive threshold setting. For additional experiments with relaxed constraints on the choice of the subset of SNVs masked $M$, see Appendix B.6.

In the case of AAFs, adding Laplacian noise provides privacy protections under the assumption that SNVs are independent, though this may not be the case in practice. However, as indicated above, our results indicate that DP mechanisms are fairly robust to attacks that leverage correlation in the data in the Beacon case. Additionally, while correlations in the Beacon case may enable an attacker to infer the true Beacon response, in the case of AAFs, only noisy statistics are exposed for the set of SNVs that are not masked, and thus we expect SPG-R to also be robust to attacks leveraging correlation information.

**Conclusion**

In this chapter, we presented a formalization to the problem of finding the optimal privacy-utility tradeoff when defending against membership-inference attacks on genomic summary releases
Figure 7.6: Adaptive threshold attack model, where the attacker leverages correlation data, baselines flip SNVs. $K = 5$, baselines only flip SNVs.

( Beacon services and summary statistics), allowing - unlike prior studies - for the defense to combine masking of SNVs and the addition of noise to best balance the two while accounting for the relative cost of adding noise as compared to suppressing responses. In the case of Beacons, We further evaluate an extension of the proposed approach against a more powerful attacker model where correlations between SNVs are exploited to infer modified responses. We present a simple yet principled greedy algorithm for both release models to discover the best privacy-utility balance which outperforms prior art, evaluating it against powerful attacks from recent literature. In most cases, our approach outperforms existing methods in terms of privacy, particularly in the context of the adaptive threshold attacker where our approach protects privacy for nearly all individuals. Our approach preserves privacy in a manner that allows the data to retain the highest utility. Several thousand fewer SNVs are modified by our method in the case of Beacons. In the case of AAFs, masking a small number of SNVs allows us to get away with adding noise that is orders of magnitude smaller than traditional differential privacy based methods to achieve a comparable level of privacy. It should be recognized that our approach does have certain limitations, in that it is specific to the MI attacks that leverage an LRT score. More powerful attacks may be devised that defeat our approach.
**Algorithm 6: Soft-Privacy-Greedy-Real (SPG-R)**

**Input:** A set of individuals $i \in D$, marginal contributions of masking $\Delta_j^M$ for each SNV, a prediction threshold $\theta$, weight parameter $w$, relative cost of adding noise $\alpha$, number of SNVs to mask per iteration $t$, set $E$ of candidate DP parameters, AAFs $x$ for individuals in $D$ and $\bar{p}$ for individuals in reference set $\bar{D}$.

**Output:** Subset of SNVs $M \subseteq S$ to mask, real valued noise vector $\delta$.

**Initialization:** $M = \emptyset$, $C = \emptyset$, $U = \infty$, $\delta = 0$, $c = 0$, $\Delta^S = \text{Sort}(|\Delta^M|)$, $M_t = \emptyset$

**Function** $\text{GetLR}(d_i, \delta, M)$:

\[
\text{return} \sum_{j \in S \setminus M} d_{ij} \log \frac{\hat{p}_j}{x_j + \delta_j} + (1 - d_{ij}) \log \frac{1 - \hat{p}_j}{1 - (x_j + \delta)}
\]

**while** $S \setminus M_t \neq \emptyset$ **do**

Set $U_t = \infty$, $\delta_t = 0$, $C_t = \emptyset$

**for** $\epsilon \in E$ **do**

Set $\delta_\epsilon = \text{Laplacian}(0, \frac{|Q \setminus M_t|}{n\epsilon})$

Set $C_\epsilon = \emptyset$

**for** $i \in D$ **do**

if $\text{GetLR}(d_i, \delta_\epsilon, M_t) \leq \theta$ then

Set $C_\epsilon = C_\epsilon \cup i$

Set $U_\epsilon = \alpha||\delta_\epsilon||_1 + (1 - \alpha)|M_t| - w|C_\epsilon|$

if $U_\epsilon \leq U_t$ then

Set $U_t = U_\epsilon$

Set $\delta_t = \delta_\epsilon$

Set $C_t = C_\epsilon$

if $U_t \leq U$ then

Set $U = U_t$

Set $\delta = \delta_t$

Set $M = M_t$

Set $ct = 1$

**while** $ct \leq t$ **do**

Set $M_t = M_t \cup \Delta^S$

Set $c = c + 1$

Set $ct = ct + 1$

Set $U_t = \alpha||\delta_t||_1 + (1 - \alpha)|M_t| - w|C_t|$

if $U_t \leq U$ then

Set $U = U_t$

Set $\delta = \delta_t$

Set $M = M_t$

**return** $M, \delta$
Chapter 8: Yet More Powerful Attacks

Thus far, we have considered membership inference attacks that rely on likelihood ratio test (LRT) scores to make membership predictions. However, such attacks are sometimes limited by the need for external information such as requiring knowledge of allele frequencies to compute LRT scores in the case of Beacon services. In this chapter, we explore whether it is possible to leverage deep neural networks to learn an attack model for membership inference in an end-to-end fashion. Doing so has several ramifications for data custodians - a) such a model would represent an arbitrarily powerful attacker, as opposed to an LRT-based attacker for which precise bounds on its power may be obtained by leveraging problem structure, b) defenses tailored against LRT-based attacker models may not suffice to defend against a deep-learning based attack model and c) such deep learning models could potentially be trained to be robust to defensive measures using methods inspired by adversarial training. To this end, we explore the use of a deep generative neural network that takes a summary release such as a Beacon or alternate allele frequencies as input and produces predicted membership for each individual in the dataset as its output.

8.1 Generative Neural Networks for Membership-Inference

We model the attacker using a generative neural network that produces a distribution over dataset membership for a set of target individuals given the input summary data release. A generative model allows us to better capture uncertainty in the data [119, 120], which, in the case of genomic summary statistics means that we may be able to better account for data releases with some noise added to them. The attacker model architecture can also be easily used to develop privacy-protection measures such as by the joint training of a robust defender model with the attacker.

To train the neural network, consider a universe $U$ of $n$ individuals, and let $m$ be the number of
SNVs. Let $P$ be a probability distribution over $U$ such that it assigns subsets of $U$ to be in the data release $D$. Each individual in $U$ is associated with a binary vector $u$ of SNVs, where $u_j = 1$ iff the individual has the alternate allele at position $j$, and $u_j = 0$ otherwise. Let $b \in \{0, 1\}^n$ be a binary vector denoting membership in the data release, such that $b_i = 1$ iff $i \in D$ and $b_i = 0$ otherwise. Thus, $P$ in turn, induces a probability distribution over membership $b$.

We begin by considering attacks on Beacon services. Let $x \in \{+1, 0, −1\}^m$ represent the Beacon release, such that $x_j = +1$ indicates that the alternate allele at SNV $j$ is present for at least one individual in the dataset, $x_j = −1$ indicate that the alternate allele for SNV $j$ is not present in $D$ and $x_j = 0$ indicate that information about SNV $j$ has been suppressed using, for example, any of the defenses covered in the previous chapters. Similarly, note that the responses $x_j \in +1, −1$ may also be deceptive, in that they do not correspond to the true Beacon output. Let the true beacon responses be represented by $r = f(b)$, such that $r \in +1, −1^m$ where in fact, $\eta = −1$ iff for all $i \in D, u_{ij} = 0$ and $\eta = +1$ iff $\exists i \in D, u_{ij} = 1$. Finally, let $\mu \in [0, 1]$ be an auxiliary random noise input to a generative neural network $G_{\theta_A}(b, \mu)$ with parameters $\theta_A$. The attacker produces an output $s \in \{0, 1\}^n$ where $s_i = 1$ implies the attacker claims that individual $i$ is in the data release $D$.

The process is similar in the case of AAF releases. Let $x \in [0, 1]^m$ represent the AAF data release, where each entry $x_j$ may be modified by the injection of real-valued noise. Modeling SNV suppression in this setting presents an additional nontrivial challenge, as setting an AAF to 0 is not mathematically equivalent to providing no information at all, as for any SNVs that no individual has an alternate allele for, an entry of $x_j = 0$ is perfectly valid. We leave this problem to future work, considering only the modeling of noise injection using the previously described differentially-private Laplace mechanism (refer to Chapter 7) for AAF releases. Similar to Beacons, let $r \in [0, 1]^m$ represent the true AAFs ($r = f(b)$).

In practice, to train the neural network, we use a Sigmoid activation function such that $s \in [0, 1]^m$, and minimize the binary cross-entropy (BCE) loss [121]:

$$BCE(s, b) = -\frac{1}{n} \sum_{i=1}^{n} b_i \log s_i + (1 - b_i) \log(1 - s_i) \quad (8.1)$$
Fig. 8.1 presents a visualization of the model architecture used, with $m = 10000$ and the dimension of $\mu$ set to 50.

8.2 Experiments

Under the memory constraints of available GPU hardware, we restrict the experiments in this chapter to a set of 10000 SNVs from the dataset used in the last two chapters. Specifically, we construct the universe $U$ by combining the data for all individuals in the dataset. We then sample membership ($b$)
randomly and compute the Beacon release or alternate allele frequencies as a function of $b$ at each training iteration. We use the Adam optimizer [112] in PyTorch [111], using Python version 3.9.13 on Ubuntu 22.04 running on a machine with an AMD Threadripper 3960X CPU, 128GB RAM and two Nvidia 3070 GPUs. For both Beacons and AAF releases, we trained the neural network for 150,000 iterations, with a starting learning rate of $10^{-5}$, and an exponential decay of $l_{r_{t+1}} = 0.985l_{r_t}$ applied every 1000 iterations.

We begin, once again, with Beacon services, where given a universe $U$ of individuals, training proceeds by first sampling a subset $D$ of individuals from $U$ at random - this is the set of individuals considered in the Beacon - and the remaining individuals are treated as not in the Beacon. For individuals in $D$, we construct the true Beacon (the vector $r \in \{+1, -1\}^m$) that indicates the presence of an alternate allele for any individual $i \in D$, which serves as the input to our neural network. Fig. 8.2 compares the performance of the neural-network-based attacker (which we call NNA henceforth) to the likelihood-ratio-based attackers (LRT) described in Chapter 5 in the form of receiving operator characteristic (ROC) curves. We can observe that the learned NNA model performs well above random, approaching the performance of the LRT attack overall. However, given that a reasonable attacker may be assumed to have bounds on the tolerance of false positives (as is standard in prior literature pertaining to membership-inference attacks), we focus on the low FPR region of the ROC curves (zoomed-in portion showed in Fig. 8.2(b)), where the NNA model outperforms the LRT attack. Results presented show averages over 5 unique runs for the LRT attack, and 5 batches of 5 random samples each for the NNA model.

Next, we consider the case of AAF releases, where given a universe $U$ of individuals, we once again sample a subset $D$ for which summary statistics are released. The true AAFs (the vector $r \in [0, 1]^m$) are computed as the fraction of individuals in $D$ who have an alternate allele, for each of the 10000 positions considered on the genome. Fig 8.3 compares the performance of the NNA model to the LRT attack, once again using ROC curves. In this setting, the NNA model outperforms the LRT attack throughout the entire range for a potential bound on false positives. Once again,
Figure 8.2: Comparison of the performance of the neural-network-based attacker (NNA) to the likelihood-ratio-based attacker (LRT) for Beacon services.

Figure 8.3: Comparison of the performance of the neural-network-based attacker (NNA) to the likelihood-ratio-based attacker (LRT) for AAF releases.

results presented are averaged over 5 unique runs for the LRT attack, and 5 batches of 5 random samples each for the NNA model.
8.2.1 Attacker Robustness

In the previous two chapters, we explored various defensive mechanisms against membership-inference attacks. We now investigate whether such measures, in fact, suffice to defend against the proposed NNA model. In the case of Beacons, we employ the Marginal Impact Greedy (MIG) solution proposed in Chapter 6, aiming to protect privacy for all individuals in the Beacon. We compute the defender’s strategy in the adaptive threshold setting, with the percentile $k$ of individuals not in the Beacon for threshold computation set to 5. Fig. 8.4 compares the impact of MIG on the performance of the NNA and LRT models, focusing on $\text{FPR} \in [0, 0.05]$. Recall that the prediction threshold is computed as the mean of the lowest 5% of LRT scores for individuals not in the Beacon. This, however, does not equate to a maximum allowable false positive rate of 5%, but is in fact, lower. Depending on the distribution of the LRT scores of these individuals, a strategy produced by MIG for $k = 5$ may only guarantee privacy against an LRT-based attacker with an FPR tolerance that is well below 5%, as is the case in Fig. 8.4. In contrast, we observe that MIG has minimal impact on the performance of the neural network based attack model, with NNA performing better on average than the LRT attacker even after MIG is implemented.

A natural subsequent question is whether these results are only a consequence of MIG being
designed to defend against LRT-based attackers. To gain further insight into this, we investigate the impact of flipping and masking subsets of SNVs at random. Note that we do not restrict flipping or masking to rare or unique alleles alone, as doing so was demonstrated to be a largely ineffective approach in the last two chapters. Instead, we sample the probability of flipping and masking each SNV from a Beta distribution \( \beta(\mu, a, b) \) centered at the mean \( \mu \), with the parameters \( a \) and \( b \) controlling the shape of the distribution. For each SNV, if the probability of flipping or masking the SNV as drawn is above 0.5, we correspondingly modify the Beacon. If the probability for both flipping and masking are above 0.5 for a given SNV, we consider the SNV masked.

Fig. 8.5 shows the impact of increasing levels of perturbation on the NNA and LRT attack models, relative to their original performance using ROC curves. In Fig. 8.5(b), we see that the performance of the LRT attack declines steadily as the number of SNVs flipped and masked increases, with performance dropping to near-random when the parameters for the Beta distribution are set to \( a = 4, b = 8 \). In contrast, the NNA approach is relatively robust to random perturbations, with performance remaining well above random for the same Beta distribution parameter choices. We attribute this to the fact that the NNA model is a generative neural network, which is better suited to account for out-of-distribution samples at evaluation time.
Finally, we study the impact of adding noise to AAFs using the differentially private (DP) Laplace mechanism from Chapter 7. In contrast to MIG, the DP approach is not tailored specifically to LRT attacks, and is a more general framework aimed at defending against arbitrary attacks. Fig. 8.6 shows the impact of adding increasing magnitudes of noise on the NNA and LRT models.
As expected, adding more noise leads to a correspondingly lower attack performance in both models. For ease of comparison, we overlay the original performance of both methods on clean data and their performance with injected noise using the DP parameter $\epsilon = 2500$ in Fig. 8.7. Here, we observe that the impact of DP on the LRT model is significantly higher than on the NNA model. The zoomed in portion of the ROC corresponding to $\text{FPR} \in [0, 0.1]$ in Fig. 8.7(b) paints an even more worrying picture: despite adding a fairly high amount of noise to the data, the performance of the NNA model remains comparable to the performance of the LRT model on clean data.

8.3 Takeaways

The results presented in this chapter may be summarized into three key ideas:

- Optimization-based approaches targeted at attackers with a well-defined underlying mathematical model that is known a priori may not generalize well to arbitrary attackers.
- More powerful attacks than the state of the art may be constructed with the use of deep neural networks. Doing so overcomes several limitations of prior art - chief of which is that correlations between SNVs may be naturally captured by the model, which makes them relatively robust against defenses that suppress or modify a subset of SNVs.
- Defending against such an attacker in a manner that guarantees privacy for all individuals in a dataset may not be possible without a prohibitive loss in the utility of the data, and solutions must instead focus on achieving a desired privacy-utility tradeoff. However, computing such optimal tradeoffs using static methods such as the ones proposed in Chapters 6 and 7 may not be sufficient.
Chapter 9: Future Directions and Conclusion

9.1 Future Directions

Armed with these observations and lessons from previous chapters, we now describe some potential future directions for the privacy-preserving sharing of genomic data. I propose to consider the three following broad goals:

9.1.1 Defending Against Black Box Attackers

In this thesis, we observed in multiple instances that leveraging knowledge of the adversary’s process leads to significant utility gains over methods that do not. For example, in Chapter 4, knowledge of the matching process between genomic sequences and face images allowed us to construct adversarial examples with noise that is practically imperceptible to the human eye and yet they reduce the attacker’s performance to below random guessing. Similarly, in Chapters 6 and 7, we use the knowledge of how the attacker computes LRT scores, and use the underlying problem structure to calculate the marginal contributions of changing or suppressing the data release at each position. In this case, we saw significant utility gains from having to affect thousands of fewer SNVs, compared to other state-of-the-art methods.

However, a data custodian may not always be able to obtain information about the attacker’s method. Worse still, optimizing against a specific attack model may lead to defenses that do not generalize well against other potential threats, as evidenced by the experiments above. To this end, I propose to consider the question: how can we construct defenses against membership-inference attacks, where the adversary is considered a black box? A fortunate consequence of the demonstrated success of the NNA attack model above is that a data custodian may now easily
model an arbitrary attacker using a generative deep neural network. Training a defender’s strategy against such an attacker would take the form of a 2-player game, where the attacker and defender train against each other in a GAN-like fashion. Recent literature [122–124] suggests that this is an effective approach, and to the best of my knowledge, such a framework has not been proposed in the context of membership-inference attacks on genomic data.

In this context, a defender and an attacker may be seen as interacting strategically in a game. Both the defender and the attacker may be modeled using generative deep neural networks, representing their respective strategies as functions of the true membership of individuals in the universe. The defender incurs a cost for adding noise to the data (different forms of noise may be modeled as having different costs). The attacker incurs a cost for making claims that an individual is in the data release - this captures the fact that a trivial attacker would simply claim that everyone is in the data release, which is not realistic. Finally, the defender incurs a cost, and the attacker a utility from each correct positive prediction the attacker makes. For the defender, this captures the costs of privacy violations, such as payouts in case of a breach of contract.

The second question I propose to consider in pursuit of defending against black-box attackers is: *how can we construct attacker and defender models that allow for variable input dimensions?* Doing so would allow us to naturally capture selective suppression of data, which remains a challenge in AAF data releases when using a neural-network-based attacker, as well as potentially reduce training time for models transferred between datasets with a varying number of individuals.

### 9.1.2 Adding Noise to Genomes, and Evolving Datasets

In Chapter 4, we studied the addition of noise to images as a viable defense against an image-genome matching attack. It should be possible to also construct a defense where the noise is added to the genotypes (in this setting, adding noise to genomes involves changing the allele occurring at various SNVs) instead of the face images. While in our study, we restricted ourselves to manipulating images, considering that the genotypes were *already public* without any modifications to them.
However, this need not be the case in future settings. This approach presents algorithmic challenges arising from the high dimensionality of genotypes, potential restrictions on which SNVs the data custodian may or may not add noise to, and the binary nature of the noise itself.

A related problem has to do with the release of summary information with datasets that are changing over time - either through the addition or deletion of records. The effect of such changes to the dataset on the summary information released has been shown [116] to leak membership information. The authors of the same study propose updating Beacon responses considering the size of the Beacon and the update, the diversity within the Beacon members and the potential risks to privacy as quantified by their proposed attack model. However, the effectiveness of the proposed measures remains to be studied, and must also consider alternate allele frequency releases, and the more powerful adaptive attack model presented in this thesis.

9.1.3 Privacy and Utility Preserving Synthetic Data Generation

Finally, I propose to consider the following question: can we leverage joint neural network training to construct synthetic datasets that are both privacy-preserving and also hold utility in downstream tasks? In this scenario, a data generator neural network (autoencoder) would be trained against an ensemble of task networks - each representing an application that the data release may be used for, such as phenotype association studies - and an attacker network trying to predict membership of a target genome in the original dataset while using the generated synthetic data as input.

9.2 Conclusion

This thesis studied the problem of membership inference attacks on various forms of genomic data release. In the first part of the thesis, we considered attacks where an adversarial actor attempts to link publicly posted face images to genomic sequences. Chapter 3 formalizes this attack, which proceeds by predicting phenotypes from DNA, as well as extracting phenotypes from face images using deep neural networks, and assigning a likelihood score to each genome-face pair. Chapter 4
proposed a technique to defend against such attacks inspired by adversarial machine learning literature. We inject noise designed to maximize the misclassification error of phenotype-predicting neural networks into face images and show that the approach is fairly robust to classifiers that may be trained to account for such adversarial examples.

In Part II, we considered the problem of membership inference attacks on genomic summary information, in the form of Beacon services that expose the presence or absence of alternate alleles, or data releases sharing alternate allele frequencies in a dataset. Chapter 5 presented an overview of state-of-the-art attack models, as well as a novel attack model that leverages the separation between LRT scores of individuals in a data release from those of a set of individuals who are not. In Chapter 6, we developed highly scalable heuristics, taking inspiration from the greedy solution to the Min-Set-Cover problem, combined with computing the marginal contribution of perturbing each SNV by either adding noise or suppressing it to the likelihood ratio test score used by the attacker. We also presented a formal model of the online Beacon, as well as Beacons with or without access control through user authentication.

In Chapter 6, we extend our methods to alternate allele frequency releases as well. We also extend our methods to permit the data custodian to optimize the privacy-utility balance of a data release as desired, instead of necessarily trying to protect privacy for all individuals. We combined the use of noise injection and SNV suppression in a significant departure from prior art and demonstrated the utility gains in doing so. In Chapter 8, we proposed the use of generative neural networks to model a black box attacker that learns to perform membership inference in an end-to-end fashion. We showed that this model is fairly robust to the various defensive measures proposed both in this thesis and in prior literature, sparking further privacy concerns. Finally, I proposed directions for future work in four key areas: a) defending against black box attackers in a game-theoretic framework, b) modeling attackers and defenders with variable input dimensions, c) adding noise to genomes and accounting for datasets that change over time and d) the use of joint-training frameworks for privacy and utility preserving synthetic genomic data generation.
References


16. Church, G. M. *The personal genome project* 2005 (cited on pp. 2, 8).


49. Reardon, S. Geneticists pan paper that claims to predict a person’s face from their DNA. *Nature* **549** (2017) (cited on p. 11).


Appendix A: Supplement for Chapter 3

A.1 Matching accuracy with varying prediction thresholds - Receiver Operating Characteristic (ROC) curves

When predicting that an image matches a particular genome, we can use thresholds to make predictions in two different ways. The first method is by thresholding with an integral value $k$, where we predict a positive match when a selected genome is in the top-$k$ potential matches by likelihood, given an image. Alternatively, we can predict a positive match for each image-DNA pair if the matching likelihood is above a real-valued threshold $\theta$. These approaches complement each other, in the sense that in the former case, we understand the false positive vs. true positive rate tradeoff in terms of narrowing potential matches down to a likely subpopulation, whereas the latter allows us to understand the performance of making independent predictions for image-DNA pairs. Fig. A.1 (A)-(L) show the precision-recall curves for various population sizes when $k$ is increased from 1, where a single match is predicted, to the population size where everyone is predicted to be a match. Similarly, Fig. A.1 (M)-(X) show ROC curves for various population sizes, when the threshold $\theta$ is increased from 0 to 1 in real-valued increments. In both cases, we observe that the classifier performs better than a random-guessing baseline, although the area under the curve remains relatively low, around 0.7 for the top-$k$ method, and 0.6 when making independent predictions.
A.2 SVM - Matching as Binary Classification

For the sake of completeness, we also study the use of classical machine learning methods to predict matches between images and DNA, given a vector representing the likelihood of phenotypes detected in an image corresponding to a selected genome. We train a linear SVM with an equal number of true and false matches, selected from subsets of the 126 individuals (10-fold cross-validated). From the results in Fig. A.2 (A)-(C), we see that this approach does not contribute much to matching accuracy, likely arising from limited signal present in the small dataset. Linear SVM was found to outperform SVM with a non-linear (rbf) kernel, as well as several other learning methods.

A.3 Addressing Challenges in Phenotype Predictions

Next, we seek to explain the significantly larger gap between the accuracy of predicted matches and the accuracy using ground-truth phenotypes in the synthetic datasets. Experiments reveal that this behavior is rooted in the poor performance of our eye-color classification model, owing to both sparsity and low-quality of data available for training, and the fact that eye-color prediction remains an open problem in computer vision - an especially hard one at that. Having evaluated multiple approaches (including segmentation of the eyes and classic machine learning methods with color histograms), we did not have much success in improving our predictions for this particular phenotype. The following are its implications to matching accuracy. We present as evidence two complementary sets of results - matching accuracy when eye color is entirely ignored, and matching accuracy when predicted variants from images are used for all phenotypes but for eye color, where we swap in favor of the ground-truth variants instead.

Results in Fig. A.3 (A)-(D) confirm our suspicions. Ground truth accuracy drops significantly when eye-color is entirely disregarded, signaling the importance of the phenotype in matching, while our predicted accuracy slightly increases upon disregarding eye-color, signaling a high volume of noise in our predictions. Subsequently, replacing our eye-color predictions with ground truth values
produces accuracies that are nearly the ground-truth upper bound, strongly indicating our eye-color prediction models act as a bottleneck in the matching pipeline.

We consider a number of techniques to address this issue, ranging from conventional machine learning to neural networks. Because eyes are a small fraction of a face image, it could be very likely that the rest of the image makes eye color prediction harder. To understand the impact of this, we segment the eyes from each image using Multitask Cascade CNNs [94], and use each approach on the segmented eyes (with the exception of VGGFace, which was designed specifically to work on full face images). From conventional machine learning, we report results using k-Nearest-Neighbors and a Support Vector Machine (SVM) with the rbf kernel, which outperform others with test accuracies of 57.89% and 51.75% respectively. We also build a shallow convolutional neural network, whose accuracy seems to be high at 60%. Unfortunately, this CNN converges to always predicting the majority class (Brown eyes), leading to its seemingly high performance. By contrast, the VGGFace classifier achieves a test accuracy of 59%.

Fig. A.3 (E)-(G) show re-identification performance when using each approach to predict eye color. None of the approaches are particularly different from the VGGFace classifier. Future advances in computer vision could solve this problem, proposing an increased risk of re-identification in the wild.
Figure A.1: Receiver Operating Characteristic curves for various population sizes. (A)-(L): Top \( k \) entries in the sorted list of DNA sequences per image are predicted to be matches. (M)-(X): Predictions are made independently for each image-DNA pair.

[150]
Figure A.2: **Using SVMs to fine-tune matching.** (A)-(C): Matching accuracy when SVMs are used on the 126 individuals to fine-tune. From left to right, (A) Top 1, (B) Top 3, (C) Top 5 matching accuracy with SVMs. The input to the SVM for an image-DNA pair is a vector of probabilities of phenotypes, where phenotypes are extracted from the image and conditional probabilities are calculated from the DNA sequence, and the labels are binary indicating a match or otherwise. For each image in the test set, we rank all DNA sequences in order of their distance from the separating hyperplane, from most likely match to least likely match, and consider the top $k$ as true matches. We note that this approach does not significantly improve matching accuracy, which we believe to be the result of low signal-noise ratio.
Figure A.3: **Effect of eye-color prediction accuracy.** (A)-(B): Matching accuracies with and without considering the eye-color phenotype, both predicted and ground-truth for (A) Ideal and (B) Realistic synthetic datasets. Notice the significant drop in ground-truth accuracy when eye-color is disregarded entirely. This points to the high importance of eye-color in matching. At the same time, notice that in the ideal dataset, disregarding eye-color entirely produces better matching accuracy than when including it, although it does not make a significant difference to the realistic dataset. This points to the presence of significant noise in our eye-color predictions. (C)-(D): Matching accuracies when the eye-color predictions from images are replaced by ground-truth values, for (C) Ideal and (D) Realistic synthetic datasets. The significant increase in matching accuracy (nearly the upper bound) in both synthetic datasets strongly suggests that we are limited in our matching ability by the poor performance of eye-color prediction. (E)-(G): Top-1, Top-3 and Top-5 matching accuracies with various eye-color prediction techniques. None of them are particularly effective.
Appendix B: Supplement for Chapter 7

This supplement provides additional results for the various threat models described in Chapter 7, runtime analysis for the SPG-B algorithm, and a parallel implementation of SPG-R which leverages problem structure to speed up computation.

B.1 SPG Runtime Analysis

We show that the running time of the proposed SPG-B and SPG-R algorithms is quadratic in the number of SNV queries $m$ and linear in the number of individuals $m$ in the Beacon.

Theorem B.1.1. The worst-case running time of the Soft-Privacy-Greedy-Binary algorithm is $O(m^2n)$.

Proof. All operations inside the first for loop are constant time, and the loop executes $m$ times, where $m$ is the number of SNVs - therefore yielding a complexity of $O(m)$. The if condition inside the second for loop involves a sum over $m$ SNVs, and is therefore an $O(m)$ operation, and this for loop executes at most $n$ times in the worst case, with a total time complexity of $O(mn)$. The remaining statements within the while loop are constant time operations. The outer while loop executes at most $m$ times, as all SNVs are flipped or masked in the worst case, and therefore the overall time complexity of SPG-B is $O(m(m + mn + 1)) = O(m^2n)$. $\square$

Theorem B.1.2. The worst-case running time of the Soft-Privacy-Greedy-Real algorithm is $O(m^2n)$.

Proof. The GetLR module is linear in the number of SNVs, i.e. $O(m)$. Statements inside the first for loop execute for each candidate value of $\epsilon$, and the number of candidate values, i.e., $|E|$ is a constant ($k$). Statements inside the inner for loop execute once for each individual, i.e. exactly
\( n \) times. All `Set` commands are constant time. Thus the inner for loop has time complexity \( O(kmn) = O(mn) \). The while loop executes \( t \) times, where \( t \) is the number of SNVs to mask per iteration, and has time complexity \( O(t) \). The outer while loop iterates over all SNVs in batches of size \( t \), which is essentially linear in \( m \). The overall time complexity of SPG-R, therefore, is 
\[
O(m(mn + t)) = O(m^2n + mt) \approx O(m^2n),
\]
given that \( m \gg t \).

**Remark.** In practice, \( m \gg n \), therefore we can treat \( n \) as a small constant, and the runtime is approximately \( O(m^2) \). We further note that the worst-case running time complexity of MIG and SPG-B is of the same magnitude, as they proceed almost identically, except that in SPG-B we consider marginal impacts of masking, as well as flipping each SNV, and compare the two which are constant time operations inside a loop that iterates over all SNVs.

### B.2 Improving Runtimes for SPG-R

When optimizing the real-valued noise using the Laplace mechanism, we can utilize some structural observations to avoid redundant computations over large matrices which, in turn, can cut down runtimes by orders of magnitude. Our first observation is that the scale of the Laplacian distribution from which random noise is drawn is directly proportional to the number of SNVs that are not yet masked. Recall that the scale of the Laplacian when no SNVs are masked is \( m/n\epsilon \). If a \( k^{th} \) fraction of SNVs remains unmasked, the scale of the Laplacian, accordingly, is \( m/kn\epsilon \). Consider the noise added per SNV for a given value of \( \epsilon \), when no SNVs are masked. This same amount of noise is achieved per SNV when a \( k^{th} \) fraction of SNVs is unmasked, with noise added corresponding to \( \epsilon/k \), and herein lies our first runtime improvement. When a noise sample is drawn from a Laplacian distribution with scale \( s \) for a given value of \( \epsilon \), the same noise can be used at scale \( s/k \) for a corresponding DP parameter \( \epsilon/k \). While drawing a noise sample is not an expensive operation in itself, this observation allows us to re-use previously computed values for \( \Delta^M_{ij} \) and \( \Delta^{M(K)}_{ij} \) at different (scaled) values of \( \epsilon \) as more SNVs are masked. As both \( \Delta^M \) and \( \Delta^{M(K)} \) contain one
Algorithm 7: The SPG-R (parallel) Algorithm

**Input:** A set of individuals $i \in D$, a prediction threshold $\theta$, weight parameter $\omega$, marginal contributions of masking $\Delta^M$, relative cost of adding noise $\alpha$, number of SNVs to mask per iteration $t$, set $E$ of candidate DP parameters, AAFs $x$ for individuals in $D$ and $\bar{p}$ for individuals in reference set $\bar{D}$.

**Output:** Subset of SNVs $M \subseteq Q$ to mask, real-valued noise vector $\delta$.

**Initialization:** $M = \emptyset$, $C = \emptyset$, $U = \infty$, $\delta = 0$

**Function** GetLR($x_i, \Delta, M$):

```latex
\text{return } \sum_{j \in Q \setminus M} -\Delta_{ij}
```

for $\epsilon \in E$, in parallel do

```latex
\text{Set } \delta^\epsilon = \text{Laplacian}(0, \frac{|Q|}{n^\epsilon}), M_\epsilon = \emptyset, c_\epsilon = 0
\Delta_{ij}^\epsilon = -d_{ij} \log \frac{p_j}{x_j + \delta_j^\epsilon} - (1 - d_{ij}) \log \frac{1 - p_j}{1 - (x_j + \delta_j^\epsilon)}
\Delta_j^\epsilon = \frac{1}{|D|} \sum_{i \in D} \Delta_{ij}^\epsilon
\Delta_{S\epsilon} = \text{Sort}(\Delta_j^\epsilon)
```

while $Q \setminus M_\epsilon \neq \emptyset$ do

```latex
\text{Set } C_\epsilon = \emptyset
\text{for } i \in D \text{ do}
\text{if GetLR($x_i, \Delta^\epsilon, M_\epsilon$) \leq \theta then}
\text{Set } C_\epsilon = C_\epsilon \cup i
\text{Set } U_\epsilon = \alpha\|\delta_\epsilon\|_1 + (1 - \alpha)|M_\epsilon| - \omega|C_\epsilon|
\text{if } U_\epsilon \leq U \text{ then}
\text{AcquireLock($U, \delta, M$)}
\text{Set } U = U_\epsilon
\text{Set } \delta = \delta^\epsilon
\text{Set } M = M_\epsilon
\text{ReleaseLock($U, \delta, M$)}
\text{Set } ct = 1
\text{while } ct \leq t \text{ do}
\text{Set } M_\epsilon = M_\epsilon \cup \Delta_{S\epsilon}^c
\text{Set } c = c + 1
\text{Set } ct = ct + 1
```

**return** $M, \delta$

entry per individual per SNV, re-computing values for a fresh noise sample each time contributes significantly to overall runtime. As long as the choices of $\epsilon$ at the beginning of the algorithm are
well-spread out (over multiple orders of magnitude, as the best solutions may involve masking a significant fraction of SNVs), we can avoid re-computing $\Delta^M$ and $\Delta^{M(K)}$ as we change the scale of the Laplacian, instead assuming the noise to be generated for a correspondingly scaled value of $\varepsilon$. We note that the $\ell_1$ norm of the noise for the objective function would still have to be recomputed over only the SNVs that remain unmasked, but this is a relatively inexpensive operation.

The second structural observation about our solution approach is that the privacy-utility points explored for a given set of candidate $\varepsilon$ values are independent of $w$, the relative importance of guaranteeing privacy over preserving utility. In Algorithm 6, threads for parallel processing are initialized after masking every $t$ SNVs. Even if we use global variables (one instance of noise $\delta$ and $\Delta^M_{ij}$ or $\Delta^{M(K)}_{ij}$, depending on the attacker model, for each value of $\varepsilon$), the repeated creation and synchronization of threads before masking the next set of SNVs can add significant overhead. To deal with this, we re-formulate our search on a per-$\varepsilon$ basis, where each thread masks SNVs locally. Threads still share access to the matrix $\mathbf{x}$, but avoid repeated function calls and synchronization wait times. Moreover, we can save all candidate solutions explored by recording $||\delta||$, $\sum_j y_j$, and $\sum_i z_i$ under either attack model, and for any value of $\alpha$ and $w$, and find the best solution in linear time over the saved candidate solutions, which are in turn linear in the number of SNVs. The algorithm, SPG-R (parallel), that takes advantage of these improvements, is provided in Algorithm 7.

**Optimizing Over $\varepsilon$**

We compare the two variants of SPG-R (binary and parallel) under the harder of the two attacker models - the adaptive threshold setting. Fig. B.1 shows the relative performance of the two variants

![Figure B.1: Relative performance of SPG-R (binary) and SPG-R (parallel) under the adaptive threshold model](image)
when $K = 5$ and $K = 10$. Binary search was initialized with $\varepsilon \in \{10^K, 10^M\}$, and candidate values for SPG-R (parallel) were selected from $\varepsilon \in \{10^K, 50^K, 100^K, 500^K, 1^M, 5^M, 10^M\}$. We can observe that SPG-R (parallel) significantly outperforms SPG-R (binary) in utility-privacy tradeoffs, likely because the latter does not explore useful tradeoff points.

**Empirical Runtime Comparison** Next, we compare the runtimes for the various methods used in Fig. B.2. The number of SNVs masked in one iteration ($t$) used for each approach is indicated in the plot. We also compare the runtimes to a projected estimate of a non-parallel naive implementation of SPG-R, where solutions over the various candidates for $\varepsilon$ are sequentially computed. Runtime for DP is omitted because it is too small compared to the rest. The runtime for SPG-R (binary) is an order of magnitude larger than SPG-R (parallel), even when masking 10 times the number of SNVs in each iteration ($t = 10^K$), taking about 5.5 hours in practice. SPG-R (binary) with $t = 1000$ can therefore be expected to take in excess of 55 hours. The estimated runtime for naive implementation of SPG-R is calculated by multiplying the average runtime of SPG-R (parallel) with the number of threads and adding some marginal overhead for thread creation and synchronization. Runtime for Linkage includes the time taken to compute linkage disequilibrium coefficients for pairs of SNVs using a sliding window of 500 SNVs (250 on either side of each SNV), which takes about 15 hours in practice, although we note that this is a parallelizable problem with scope for shared data structures, and the computation only needs to be done once.
B.3 Additional Results - Fixed Threshold

First, we present additional results where SPG-B combines flipping and masking, while baselines solely flip or mask SNVs for $\theta = -750$ in Fig B.3. Performance, in this case, is similar to the results for $\theta = -250$, in that SPG-B Pareto dominates all baselines, and shows significant improvements in utility over DP and RF - both when baselines only flip SNVs (Fig. B.3A) or mask SNVs (Fig. B.3B).

![Utility-privacy plots for the fixed threshold attack model for Beacons, compared to baselines.](image)

Figure B.3: Utility-privacy plots for the fixed threshold attack model for Beacons, compared to baselines.

Special Case: Only Flipping SNVs

We now consider the special case where all approaches, including SPG-B, are restricted to flipping SNVs. This represents a scenario where suppressing Beacon responses may be impractical. In this setting, in addition to SPG-B and the various baseline methods shown in the more general setting, we also present the optimal solution computed using CPLEX [125], an ILP-solving toolkit. Since the original ILP in (7.10) is unable to scale to a search space consisting of 1.3 million SNVs, we restrict the ILP to search for an optimal utility-privacy balance over the SNVs identified by MIG in this setting (on the order of $10^2$). In this scenario, the higher the value of $\theta$, the more SNVs the defender has to flip to guarantee privacy. A higher value of $\theta$ thus more clearly demonstrates the differences in utility across the methods, and therefore we present results for $\theta = 0$ and $\theta = 1000$.

Fig. B.4 shows results in this setting when $\theta = 0$ and $\theta = 1000$. SPG-B again Pareto dominates
DP, RF, and SF. SF guarantees privacy for all individuals while offering much lower utility. MIG guarantees privacy for all individuals with a slightly lower utility when \( \theta = 1000 \), dropping further when \( \theta = 0 \). In practice, the difference between the performance of MIG and SPG-B arises from flipping about 10 additional SNVs to guarantee privacy for only a single individual in the dataset. For a very large value of the weight parameter \( w \), SPG-B produces the same solution as MIG in this setting.

**Special Case: Only Masking SNVs** We now consider the alternative case where SNVs are only masked. This setting also serves to demonstrate the greater loss of utility that must be tolerated to achieve privacy for all using just masking. Note that the impact of masking a Beacon response is smaller than that of flipping it.

Fig. B.5 presents the results when the \( \theta \) prediction threshold is set to -250 and -750. Yet again, we observe that SPG-B Pareto dominates all baselines, with MIG offering comparable utility when the privacy of all individuals is necessarily guaranteed. Comparing the performance of SPG-B
between Fig. B.5 and Fig. 7.1 for $\theta = -250$, it can be seen that choosing to flip a small number of SNVs and masking the remaining greatly improves utility.

### B.4 Additional Results - Adaptive Threshold

Here, we present some additional results in the adaptive threshold setting for both SPG-B and SPG-R. First, we compare the performance of SPG-B to various baselines, when the threshold is set to the mean of $K = 5$ lowest percentile of LRT scores, where baselines either flip or mask SNVs, while SPG-B combines both. From Fig. B.6, we can observe that SPG-B once again dominates all baselines. None of the baselines offer any privacy when they are restricted to masking SNVs in this case.

![Figure B.6: Utility-privacy plots for the adaptive threshold attack model for AAF releases, compared to baselines.](image)

(A) $K = 5$, baselines only flip SNVs  
(B) $K = 5$, baselines only mask SNVs

**Special Case: Only Flipping SNVs**

Finally, we present results in the case where all methods, including SPG-B, only flip SNVs. Fig. B.7 compares SPG-B to the baselines. It can be seen that SPG-B offers a better privacy-utility balance than all methods, except for MIG when privacy for all individuals is to be guaranteed. In comparison to Figs. 7.3 and B.6, it can be seen that SPG-B has a slightly lower utility. This further illustrates the value of applying a method that uses both flipping and masking.
B.5 Additional Results - Bounded Risk

Here, we compare the performance of SPG-R (parallel) to the various baselines, under the assumption of bounded risk, where the sensitivity of the mean query is calculated in the average case instead of the worst case, and correspondingly the scale of the Laplacian depends on the average number of bits by which a genome in the dataset differs from those not in the dataset. On our data, the average sensitivity is $148515$, which is an order of magnitude smaller than the number of SNVs.

This has no qualitative impact on the best solutions found by our approach, except that a correspondingly smaller value of DP parameter $\varepsilon$ is now used to generate the same amount of noise. Because our candidate $\varepsilon$ values were well spread out ($\varepsilon \in \{10K, 50K, 100K, 500K, 1M, 5M, 10M\}$), our approach works well without any modifications.

**Fixed Threshold Attacks**

Fig. B.8 compares SPG-R to the various baselines under the fixed threshold attack model, when $\theta = 0$ for $\alpha = 0.5$ and $\alpha = 0.9$. As is the case with unbounded risk, SPG-R dominates all baselines when $\alpha = 0.5$, but as masking gets relatively cheaper compared to adding noise, SPG-R is dominated by a pure-masking strategy, with the difference being more pronounced in the bounded risk scenario.

**Adaptive Threshold Attacks**

The performance with bounded risk in the adaptive threshold setting yet again qualitatively mirrors the results in the unbounded risk setting, with SPG-R dominating all baselines, as we can observe from Fig. B.9.
Figure B.8: Performance of SPG-R compared to baselines in the fixed threshold setting (bounded risk) $\theta = 0$.

Figure B.9: Performance of SPG-R compared to baselines in the adaptive threshold setting (bounded risk) for $K = 5$ and $K = 10$.

Figure B.10: Adaptive threshold attack model, where the attacker leverages correlation data, and baselines only flip SNVs. The Greedy approach is permitted to flip or mask all SNVs. $K = 5$
B.6 Additional Results - LD

In the adaptive threshold case, recall that flipping or masking is restricted to SNVs where $\Delta_{ij}^{F(K)} \geq 0$ and $\Delta_{ij}^{M(K)} \geq 0$ respectively for all individuals in the Beacon. When an SNV $j$ is flipped or masked by $SPG-LD$, there may be SNVs in $N_{LD}(j)$ for which these inequalities may not hold true and are therefore not flipped or masked. These SNVs however may still be used to infer flipped or masked Beacon responses by measuring correlations.

The correlation attack had no impact on SPG-BB when $K = 10$. None of the SNVs picked by SPG-B in this setting to either flip or mask had correlations with other SNVs within a sliding window of 500 SNVs (250 on each side). Therefore, we ran a second set of experiments that neglects these inequality constraints, essentially allowing Beacon responses for all SNVs to be flipped or masked, with the consequence that privacy achieved in early iterations of the greedy algorithms may be reduced by later flips or masks. Fig. B.10 presents the results. While the performance of SF, SFM and MIG are unchanged compared to the previous setting in Fig. 7.6, notice that for both SPG-B and SPG-LD, the privacy starts decreasing after a point, as more and more SNVs are flipped or masked. The maximum privacy achieved by the modified greedy algorithm accounting for correlations, SPG-LD, is around 90%. This is a marginal increase compared to the previous setting.

B.7 Impact of Increasing Population Sizes

Here, we present additional results comparing the relative performance of DP to SPG-B and SPG-R for Beacons and summary statistics as we increase the number of individuals in the dataset ($n$). With the available hardware, we were able to experiment with summary statistics for datasets consisting of up to 600 individuals. Fig. B.11 shows that neither approach is significantly affected by population size in the case of Beacons. Fig. B.12 suggests that, while neither approach is affected in the fixed-threshold setting for summary statistics, the relative performance of DP deteriorates in
Figure B.11: Relative performance of DP and SPG-B with increasing dataset size ($n$). Zoomed-in portions are shown on top.

terms of utility as the number of individuals increases in the adaptive threshold case. The slight
difference in performance trends between these figures and the results presented in the main paper
is due to randomly splitting the universe of all individuals ($D \cup \bar{D}$) into two sets of varying sizes
here, as compared to working with a particular 400-400 split in previous settings.

Figure B.12: Relative performance of DP and SPG-R with increasing dataset size ($n$)
B.8 Empirical Runtime - ILP vs SPG-B

Finally, we present empirical results comparing the running times of the ILP and SPG-B (refer Section B.3 for details). Because the ILP was only feasible over a small set of SNVs chosen by MIG, we restrict SPG-B to the same set of SNVs to ensure a fair comparison. For both approaches, running time was only measured for the steps involved in solving the optimization, ignoring all initialization and pre-computation steps. This analysis was performed on a 2018 MacBook Pro with an Intel i7 processor and 16 GB of RAM.

Fig. B.13 depicts that the running time for the ILP is between 2 to 6 orders of magnitudes larger, depending on the weight parameter. For small weights, the optimal solution is to do nothing, and for large enough weights, the optimal solution is to guarantee privacy for everyone - these are solutions that a branch-and-bound approach is expected to arrive quickly at - which is what IBM CPLEX uses in order to compute optimal solutions. For weight parameters between these two extremes, we expect that the branching would need to continue to a greater depth, leading to the spike in running time at $\omega = 0.2$. On the other hand, SPG-B continues flipping or masking SNVs until all individuals are covered, regardless of $\omega$, the weight is only used to update the current best solution which is eventually returned. Therefore the running time is constant across all weights.

Figure B.13: Empirical Comparison of running times for ILP and SPG-B (using only SNVs identified by MIG)