

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/384210371>

Bisection Grover's Search Algorithm and Its Application in Analyzing CITE-seq Data

Article in *Journal of the American Statistical Association* · September 2024

DOI: 10.1080/01621459.2024.2404259

CITATION

1

READS

237

4 authors, including:



Ping Ma

University of Georgia

101 PUBLICATIONS 2,202 CITATIONS

[SEE PROFILE](#)



Yongkai Chen

University of Georgia

16 PUBLICATIONS 79 CITATIONS

[SEE PROFILE](#)



Wenxuan Zhong

University of Illinois Urbana-Champaign

64 PUBLICATIONS 1,278 CITATIONS

[SEE PROFILE](#)

Bisection Grover's Search Algorithm and Its Application in Analyzing CITE-seq Data

Ping Ma^{*†}

Department of Statistics, University of Georgia
and

Yongkai Chen[†]

Department of Statistics, University of Georgia
and

Haoran Lu

Department of Statistics, University of Georgia
and

Wenxuan Zhong^{*}

Department of Statistics, University of Georgia

December 13, 2024

Abstract

With the rapid development of quantum computers, researchers have shown quantum advantages in physics-oriented problems. Quantum algorithms tackling computational biology problems are still lacking. In this paper, we demonstrate the quantum advantage in analyzing CITE-seq data. CITE-seq, a single-cell technology, enables researchers to simultaneously measure expressions of RNA and surface protein detected by antibody-derived tags (ADTs) in the same cells. CITE-seq data hold tremendous potential for identifying the associated ADTs for the targeted genes and identifying cell types effectively. However, both tasks are challenging since the best subset of ADTs needs to be identified from enormous candidate subsets. To surmount the challenge, we develop a quantum algorithm named bisection Grover's search (BGS) for the best subset selection of ADT markers in CITE-seq data. BGS takes advantage of quantum parallelism by integrating binary search and Grover's algorithm to enable fast computation. Theoretical results are provided to show the privilege of BGS in the estimation error and computational complexity. The empirical performance of the BGS algorithm is demonstrated on both the IBM quantum computer and simulator.

Keywords: Single-cell; variable selection; quantum computing; Grover's algorithm; binary search; curse of dimensionality

^{*}Corresponding authors.

[†]Equal contributions.

1 Introduction

Recent breakthroughs in quantum computers have shown quantum advantage (aka quantum supremacy), i.e., quantum computers outperform the classic computers for solving specific problems ([Shor, 1999](#); [Arute et al., 2019](#); [Zhong et al., 2020](#); [Wu et al., 2021](#)). A more relevant fact is that there are already general-purpose programmable quantum computing devices available to the public, e.g., IBM Quantum Experience, Microsoft Quantum, and Amazon Braket. Such quantum devices are commonly known as noisy intermediate-scale quantum (NISQ) devices ([Preskill, 2018](#)). It is reported that NISQ with a thousand qubits has been released ([Castelvecchi, 2023](#)). Despite the success of quantum computers, the investigated problems are highly physics-oriented and may not necessarily appeal to researchers in other fields, e.g., statistics and data science ([Wang, 2022](#)). The key challenge in leveraging quantum advantages in these fields is identifying practical applications where the integration of statistical analysis and quantum computing can effectively overcome computational bottlenecks.

In this paper, we develop a highly versatile quantum algorithm named Bisection Grover’s search (BGS) for best model selection, which exhibits excellent performance and computational efficiency in its applications, specifically in the analysis of CITE-seq data. BGS takes advantage of quantum parallelism by integrating binary search and Grover’s algorithm to enable fast computation. We show that the proposed BGS is consistent as long as sufficient quantum bits are available. We also show that the BGS algorithm is nearly quadratic speed-up of classic algorithms in terms of time complexity. In addition, we provide the code that implements the proposed algorithm in the IBM quantum computer and the IBM quantum simulator with practical guidance.

Brief context and background. The advantages of quantum computing rely on the

fact that quantum bits carry more information than classical bits. Different from classical computers built on classical bits having a *state* of either 0 or 1, quantum computers operate on quantum processing units, quantum bits (or qubits), which can be in a state 0, 1, or both simultaneously due to the superposition property (Nielsen and Chuang, 2010). The p qubits create 2^p different states for the system that are superposed on each other. The superposition enables researchers to perform computations using all of those 2^p states simultaneously, which is also called quantum parallelism. Quantum parallelism circumvents the time/space trade-off of classical parallel computing through its ability to hold exponentially many units of information in a linear amount of physical space. In addition, a quantum computer has some logic gates, which a classic computer does not have, enabling faster computation than a classic computer (Nielsen and Chuang, 2010).

Despite the impressive achievements made possible by quantum computers, quantum algorithms for solving statistical or data science problems are still lacking. This deficiency primarily stems from the fact that many classic optimal algorithms are the culmination of intellectual efforts by generations of scientists. Developing a quantum algorithm that outperforms the classic optimal algorithms is intellectually challenging. In particular, there are significant technical obstacles in developing such quantum algorithms. First, even though quantum computing enjoys quantum parallelism, accessing the result is not straightforward. Taking measurements of a quantum system induces a superposition of quantum states collapsing into one state in a certain probability and permanently changing the state of the system. Second, many quantum algorithms often depend on an oracle function to recognize if an outcome is a solution or not. For example, Grover's algorithm requires an oracle function that can recognize solutions. However, such an oracle function is usually not available in many statistics or data science problems. Third, extensive efforts have

been devoted on analyzing the algorithmic aspect of quantum algorithms. In spite of these impressive algorithmic results, the theoretical analysis that addresses statistical aspects of quantum algorithms is still lacking. Without theoretical statistics insight, researchers may not effectively develop new quantum algorithms to tackle data science problems in a principal statistics framework.

To overcome these obstacles, we develop a BGS algorithm for the analysis of CITE-seq data. CITE-seq (Cellular Indexing of Transcriptomes and Epitopes by sequencing) is an innovative technology for studying single-cell biology ([Stoeckius et al., 2017](#)). Different from single-cell RNA-seq measuring RNA expression alone, CITE-seq enables researchers to simultaneously measure RNA and surface protein expression in the same cells. In particular, in CITE-seq, surface proteins detected by antibody-derived tags (ADTs) are transcriptomically profiled alongside RNA transcripts using single-cell RNA-seq. By utilizing CITE-seq data of cells, we can study the following two biological problems: (1) identifying the associated ADTs for the targeted genes and (2) designing panels of ADT markers for cell type identifications. Successfully tackling these two problems hinges on the effective selection of the best subset of ADTs. The best ADT subset selection problem is generically hard with classical computing since it is computationally demanding and time-consuming due to the need to evaluate an extensive number of candidate subsets. Even for 30 ADTs, i.e., $p = 30$, we have more than one billion candidate subsets, which poses significant challenges for classic algorithms to select the best subset.

Identification of ADTs associated with marker genes of interest. Despite the fact that RNAs and proteins are produced from the same genes, they provide some complementary information on cell states due to post-transcriptional and post-translational gene regulation. Since the CITE-seq data is often of high dimension and contains lots of

redundant information across different RNAs and ADTs, selecting a parsimonious set of RNAs and ADTs is crucial to leverage the information of the CITE-seq data. We aim to develop a method to identify a set of ADTs that best explains the variations of the expressions of an RNA in a regression model.

Panel design for cell type identification. Efficiently identifying immune cell types is a key objective in immunological research and clinical diagnostics. A recent study finds that CITE-seq can effectively identify compact sets of immunophenotypic markers (ADTs) for characterizing different cell types ([Hao et al., 2021](#)). Such a set of markers is referred to as a panel. Including too many markers in a panel can result in a decreased signal-to-noise ratio and increased background noise in downstream experimental validation, e.g., the spillover-spreading error inherent to the fluorescence in flow cytometry ([Ferrer-Font et al., 2021](#)). Here, we identify the best subset of ADTs for each cell type using logistic regressions by setting the cell type of interest as one and other cell types as zero. The resulting panel of ADTs can improve accuracy in identifying immune cell populations while optimizing resource allocation.

Statistically, our problem can be described as follows: CITE-seq yields a sample with RNA expressions and p ADT expressions $x_{i,1}, x_{i,2}, \dots, x_{i,p}$ in the i th cell, where $i = 1, \dots, n$ and n is the number of cells. In the problem of identifying the ADTs associated with the targeted gene, we model the expressions of RNA and ADTs through the linear regression model. Given a specific gene of interest, we represent its expression as y_i in the i th cell, serving as the response variable in the regression model. We assume that only a subset, denoted by \mathcal{A} , of ADTs co-express with the gene of interest and are utilized as the predictor variables. In the panel design for the cell identification problem, we consider the logistic regression model. The cell type of the i th cell is denoted by z_i , which is either one if it is

a cell of interest, or zero otherwise. Once again, we assume that only a subset, denoted by \mathcal{A} , of ADTs is the marker for the cell type of interest. The primary research interest is identifying the subset $\mathcal{A} \subseteq \{1, \dots, p\}$ effectively and efficiently. Here, we choose the subset \mathcal{A} using the Bayesian information criterion (BIC) (Schwarz, 1978), which is given by

$$\text{BIC}(\mathcal{A}) = |\mathcal{A}| \log n - 2 \log L(\mathcal{A}), \quad (1)$$

where $|\mathcal{A}|$ is the subset size, n is the number of cells in the training dataset, $L(\mathcal{A})$ is the maximized value of the likelihood function of the fitted model for the corresponding subset.

In the literature, many quantum-based feature and model selection methods have been explored. He et al. (2018) leverages the searching abilities of Grover's algorithm and proposes the quantum versions of the forward selection and backward elimination algorithm, achieving quadratic speedup for each step of addition or deletion. Chakraborty et al. (2020) and Li et al. (2022) explore the quantum benefits of graph theory to solve the graph-theoretic feature selection problems. Recently, the promise of high efficiency for solving the quadratic unconstrained binary optimization (QUBO) with quantum optimizers has sparked widespread interest in its study, leading to a surge in research investigating quantum-based feature selection within the QUBO model (Von Dollen et al., 2021; Turati et al., 2022; Mücke et al., 2023; Turati et al., 2022). Nonetheless, the quantum algorithm specifically designed for the aforementioned global optimization searching problems is still lacking. Furthermore, most existing researches focus on conceptual and theoretical exploration. The practical implications of leveraging quantum computing to enhance our understanding of single-cell multi-omics studies remain largely unexplored.

To overcome the aforementioned limitations, we propose BGS to select the best subset. In this method, we randomly choose a subset as our benchmark subset. This benchmark subset bisects all subsets into two partitions: an oracle set consisting of subsets having

smaller BICs than that of the benchmark subset, and a non-oracle set consisting of rest subsets. BGS starts with an initial superposition where all candidate subsets are encoded with equal weights. BGS then iteratively updates the superposition toward the oracle set and yields a new superposition. The new superposition is then measured and collapses to a new subset. The BIC of the new subset is compared with that of the benchmark set. If the new subset is better than the benchmark set, the new one replaces the benchmark one. We then repeat the above procedures until no more replacements can be made. Different from Grover’s algorithm, our BGS algorithm does not require any oracle function to recognize the best subset.

Our methodological contributions are as follows. First, we take advantage of quantum parallelism by integrating binary search and Grover’s algorithm to design a new iteration algorithm and only access the outcome at the end of each iteration. In this way, we can greatly harness the power of quantum computing. Second, our proposed algorithm does not require an oracle function to recognize the best subset. This relaxation expands the scope of applications and overcomes the obstacle of the impractical assumption that the best subset needs to be known *a priori* in Grover’s algorithm. Third, we integrate Grover’s algorithm with the quantum counting algorithm to get a high probability of identifying the best subset.

Our theoretical contribution is that we derive the error bound for the proposed BGS algorithm and show that the error can be controlled to be arbitrarily small as long as we have a sufficient number of qubits. It is the consequence of powerful quantum parallelism and accurate quantum counting. Moreover, we derive the time and space complexity of the proposed algorithm.

Our empirical contribution is that we conduct the empirical analysis in an IBM quantum

computer and IBM quantum simulator. Our empirical results are consistent with our theoretical analysis and demonstrate the quantum advantage of the proposed algorithm over classic algorithms. Our exploration facilitates future quantum algorithm developments targeting other biological problems.

2 Methods

2.1 Overview

To select the best subset \mathcal{A} that minimizes Eq.(1), one possible quantum computing method we could use is Grover’s algorithm ([Grover, 1996](#)). Grover’s algorithm is a quantum search algorithm. The intuition behind Grover’s algorithm lies in its ability to leverage quantum superposition to enhance the search process. Grover’s algorithm initializes the qubits in a superposition of all possible states, effectively exploring multiple search paths simultaneously. Grover’s algorithm then employs a technique known as amplitude amplification ([Brassard et al., 2002](#)) to enhance the probability of finding the desired solution. This technique involves iteratively applying a sequence of quantum operations to amplify the amplitude of the target state while suppressing the amplitudes of other states. This approach leads to a quadratic speedup compared to classical search algorithms, making it highly efficient for large-scale search problems. However, there are three key difficulties in applying Grover’s algorithm to our problem. First, in our problem, we do not know the oracle state *priori*. Consequently, it is not clear how to design Grover’s operation. Second, even if we know the oracle state in Grover’s algorithm, the number of Grover’s rotations needs to be determined accurately. Otherwise, Grover’s algorithm may not be as effective as what algorithmic analysis prescribes. Third, theoretical analysis, including error quan-

tification and algorithmic complexity, for taking into account the above two difficulties and uncertainties is still lacking. The details of Grover's algorithm are relegated to Section A.2 in the Supplementary Material.

We shall now develop a novel quantum search algorithm named bisection Grover's search (BGS) to address these difficulties. Recall that all subsets of $\{1, \dots, p\}$ naturally correspond to a state in the orthonormal basis $\mathcal{B} = \{|b_0\rangle, \dots, |b_{D-1}\rangle\}$ of a Hilbert space \mathcal{H} . Notice that only p qubits are sufficient to represent these states. We define a state loss function $g(\cdot) : \mathcal{H} \rightarrow \mathbb{R}$,

$$g(|b_j\rangle) \equiv \text{BIC}(\mathcal{A}_j), \quad (2)$$

where state $|b_j\rangle \in \mathcal{B}$ is a vector in \mathcal{H} that corresponds to the subset \mathcal{A}_j . Hence, the best model corresponds to the state that minimizes the state loss function. We assume there exists a sole oracle state in \mathcal{B} that minimizes $g(\cdot)$. Suppose the oracle state denoted by $|b^*\rangle$ is unique and satisfies

$$g(|b^*\rangle) < g(|b_j\rangle) \quad \text{for } |b_j\rangle \in \mathcal{B} \text{ and } |b_j\rangle \neq |b^*\rangle. \quad (3)$$

The idea of the proposed BGS algorithm can be sketched as follows. Since the oracle state is unknown, the BGS algorithm randomly selects a state and regards all the states with smaller BICs than that of the selected state as oracle states. In particular, BGS randomly selects a state in \mathcal{B} and calculates its BIC. We name this state the *benchmark state*. This benchmark state bisects the set of all states into two subsets: a subset of *oracle states* and a subset of *non-oracle states*. Oracle states are those states having smaller BICs than the benchmark state, and non-oracle states are all other states. BGS then rotates the uniform superposition of all states via Grover's operations toward the superposition of the oracle states. Once taking the measurement, the rotated superposition collapses to one of the oracle states. If the collapsed state's BIC is smaller than that of the benchmark state,

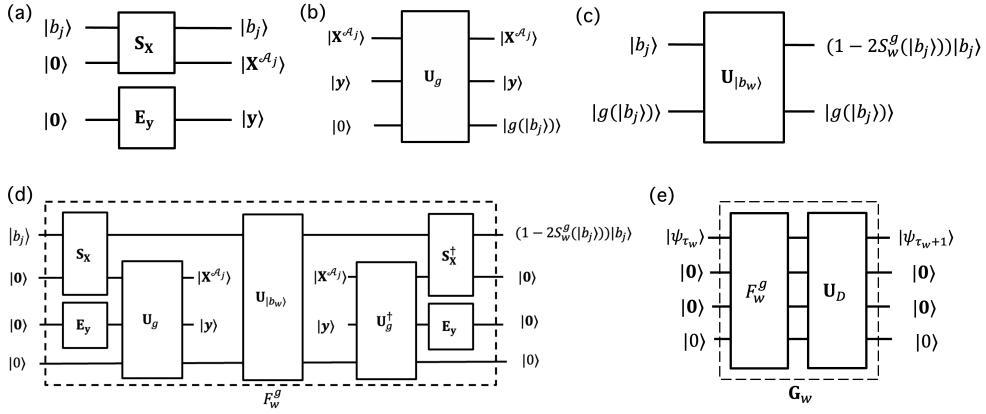


Figure 1: The quantum circuits for illustration. (a) The circuit encodes the ADT subset. (b) The circuit to calculate the state loss function $g(\cdot)$. (c) The circuit to flip the sign of oracle states. (d) The assembled circuit for the operation F_w^g . (e) The circuit for the operation \mathbf{G}_w in the BGS algorithm.

the benchmark state is replaced by it. BGS then iterates the above steps. Otherwise, we output this collapsed state.

The quantum circuits for evaluation of the state loss function $g(\cdot)$ in all D states are illustrated in Fig. 1 (a) and (b). The operation \mathbf{S}_X encodes the ADTs \mathcal{A}_j when the input state is $|b_j\rangle$. The operation \mathbf{S}_y encodes the response \mathbf{y} . The state $|\mathbf{X}^{\mathcal{A}_j}\rangle$ encodes the matrix $\mathbf{X}^{\mathcal{A}_j}$ as suggested by Schuld et al. (2016). The operation \mathbf{U}_g computes the BIC value $g(|b_j\rangle)$ with the input $\mathbf{X}^{\mathcal{A}_j}$ and the encoding of the response $|\mathbf{y}\rangle$. The detailed description of these quantum circuits is relegated to Section B in Supplementary Material.

Remark. \mathbf{U}_g can be implemented with quantum speedup. Notice that quantum computing methods for regression and classification offer exponential or polynomial speed-up compared to their classical counterparts (Biamonte et al., 2017), e.g., $O(\log(n))$ complexity for fitting linear regression models compared to $O(n)$ complexity in classical computing methods (Schuld et al., 2016) for linear regression.

Remark. The inherent advantages of quantum parallelism enable the efficient evaluation of the state loss function $g(\cdot)$ in all D states. That is, all D values can be computed simultaneously in a single operation by leveraging the power of quantum computing.

We shall now present the proposed method in detail.

2.2 Bisection and Grover's operations

BGS randomly chooses an initial benchmark state $|b_w\rangle$ from the set \mathcal{B} . Let us define a local evaluation function $S_w^g(|b_j\rangle)$ as

$$S_w^g(|b_j\rangle) = \begin{cases} 1, & \text{if } g(|b_j\rangle) < g(|b_w\rangle), \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Using this local evaluation function, $|b_w\rangle$ bisects basis set \mathcal{B} into two subsets: the subset of oracle states: $\mathcal{B}_w = \{|b_j\rangle : S_w^g(|b_j\rangle) = 1\}$ and the subset of non-oracle states: $\mathcal{B}_w^c = \{|b_j\rangle : S_w^g(|b_j\rangle) = 0\}$. If \mathcal{B}_w is a null set, i.e., $|b_w\rangle$ happens to be the unique oracle state $|b^*\rangle$, BGS can output $|b_w\rangle$ as the final result. If \mathcal{B}_w is not empty, BGS proceeds as follows. Let D_w be the cardinality of \mathcal{B}_w , i.e., the number of states in \mathcal{B}_w . BGS initializes a uniform superposition as

$$|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |b_j\rangle \equiv \sqrt{\frac{D_w}{D}} |\phi_w\rangle + \sqrt{\frac{D - D_w}{D}} |\zeta_w\rangle, \quad (5)$$

where $|\phi_w\rangle = \frac{1}{\sqrt{D_w}} \sum_{j \in \mathcal{B}_w} |b_j\rangle$ and $|\zeta_w\rangle = \frac{1}{\sqrt{D - D_w}} \sum_{j \in \mathcal{B}_w^c} |b_j\rangle$. Notice that $|\phi_w\rangle$ is the linear combination of the oracle states, whereas $|\zeta_w\rangle$ is the linear combination of the non-oracle states. Define as θ_w the angle between $|\psi_0\rangle$ and $|\zeta_w\rangle$. By the definition, we have

$$\sin(\theta_w) = \sqrt{D_w/D}. \quad (6)$$

Since $|\phi_w\rangle$ and $|\zeta_w\rangle$ are orthonormal, these two superpositions form a two-dimensional Hilbert space. This two-dimensional Hilbert space can be represented as the column space of a $2^p \times 2$ matrix $\mathbf{\Gamma}_w$, where the two columns are the corresponding vector representations of $|\phi_w\rangle$ and $|\zeta_w\rangle$. The initial state vector then can be written as $|\psi_0\rangle = \sin \theta_w |\phi_w\rangle + \cos \theta_w |\zeta_w\rangle$. We now rotate this state vector toward $|\phi_w\rangle$. Analogous to Grover's algorithm, we define a

flip operation F_w^g , where $F_w^g |b_j\rangle = (1 - 2S_w^g(|b_j\rangle))|b_j\rangle$. The implementation details of the flip operation are illustrated in Fig.1 (c) and (d). The operation $|\mathbf{U}_{|b_w}\rangle$ flips the sign of the input state $|b_j\rangle$ if $g(|b_j\rangle) < g(|b_w\rangle)$, i.e., $S_w^g(|b_j\rangle) = 1$. Consequently, the flip operation can be constructed by assembling the operations for evaluation of the state loss function $g(\cdot)$ and $|\mathbf{U}_{|b_w}\rangle$. Notice that to avoid the unnecessary entanglement created by qubits encoding $|\mathbf{X}^{A_j}\rangle$ s and $|g(|b_j\rangle)\rangle$, we use a standard approach by uncomputing the evaluation of the state loss function with the conjugate transpose of the operations $\mathbf{S}_\mathbf{X}$ and \mathbf{U}_g , i.e., $\mathbf{S}_\mathbf{X}^\dagger$ and \mathbf{U}_g^\dagger respectively.

The diffusion operation \mathbf{U}_D remains the same as the one defined in Eq.(A.8). We thus have Grover's operation (with respect to $|\phi_w\rangle$) $\mathbf{G}_w = \mathbf{U}_D F_w^g$. Fig.2 provides a visualization of the two operations in the first rotation of BGS. It is easy to derive that \mathbf{G}_w can be decomposed as follows,

$$\mathbf{G}_w = \Gamma_w \begin{pmatrix} \cos 2\theta_w & \sin 2\theta_w \\ -\sin 2\theta_w & \cos 2\theta_w \end{pmatrix} \Gamma_w^\top + \tilde{\Gamma}_w \tilde{\Gamma}_w^\top, \quad (7)$$

where $\tilde{\Gamma}_w$ is a $2^p \times (2^p - 2)$ matrix of which columns are orthonormal and orthogonal with Γ . The detailed derivation can be found in Section 6 of [Nielsen and Chuang \(2010\)](#).

After we apply Grover's operation τ_w times to it, $|\psi_0\rangle$ becomes

$$|\psi_{\tau_w}\rangle = \sin((2\tau_w + 1)\theta_w)|\phi_w\rangle + \cos((2\tau_w + 1)\theta_w)|\zeta_w\rangle. \quad (8)$$

As long as $(2\tau_w + 1)\theta_w \leq \pi/2$, BGS gradually amplifies the amplitudes of the oracle states and suppresses the amplitudes of non-oracle states. Notice that all the states' amplitudes are updated simultaneously in this procedure, resulting in highly efficient searching.

Once a measurement is taken, $|\psi_{\tau_w}\rangle$ collapses to a state $|b_{w^{new}}\rangle \in \mathcal{B}_w$. The probability

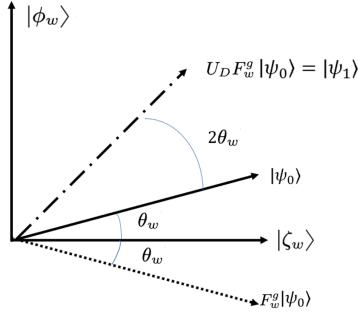


Figure 2: Geometrical illustration of the two operations in the first rotation step of the BGS algorithm. After taking the diffusion operation \mathbf{U}_D and the flip operation F_w^g , the initial state vector $|\psi_0\rangle$ moves to $|\psi_1\rangle$, which is closer to the oracle states $|\phi_w\rangle$.

that $|b_{w^{new}}\rangle = |b_j\rangle$ is

$$\mathbb{P}(|b_{w^{new}}\rangle = |b_j\rangle) = \begin{cases} \frac{\sin^2((2\tau_w+1)\theta_w)}{D_w}, & \text{if } S_w^g(|b_j\rangle) = 1, \\ \frac{\cos^2((2\tau_w+1)\theta_w)}{D-D_w}, & \text{otherwise.} \end{cases} \quad (9)$$

Now the output is $|b_{w^{new}}\rangle = |b_j\rangle$. If $S_w^g(|b_j\rangle) = 0$, BGS does not update benchmark state $|b_w\rangle$. Otherwise, i.e., $S_w^g(|b_j\rangle) = 1$, BGS replaces benchmark state $|b_w\rangle$ by $|b_{w^{new}}\rangle$, i.e., updates $|b_w\rangle$ to an oracle state.

The quantum circuit for implementing Grover's operation \mathbf{G}_w is visualized in Fig.1 (e).

2.3 Estimating the number of Grover's operations

BGS only updates the benchmark state if the superposition collapses to an oracle state. By Eq.(9), the probability that the superposition collapses to an oracle state is the highest if $(2\tau_w + 1)\theta_w = \pi/2$. Thus the efficiency of BGS highly depends on the choice of τ_w , the number of Grover's operations. Since θ_w is also unknown, a proper choice of τ_w relies on an accurate estimate of θ_w . Note that we cannot estimate θ_w from Eq.(6) since \mathcal{B}_w is also unknown.

We propose to estimate θ_w using the quantum counting algorithm (Brassard et al., 1998). Since the presentation of the quantum counting algorithm is lengthy, we relegate

the details to Section A.3 in Supplementary Material. The key idea of the quantum counting algorithm is sketched as follows. Since $0 \leq \theta_w < \pi$, the T -digit binary approximation of θ_w is $\theta_w/\pi \approx 0.e_1e_2\cdots e_T = \sum_{k=1}^T e_k 2^{-k}$, where e_k is either 0 or 1. The quantum counting algorithm employs a divide-and-conquer strategy by dividing the problem of estimating θ_w into T sub-problems of estimating each digit e_k separately for $k = 1, \dots, T$ with T additional (auxiliary) qubits. The estimates of e_k for $k = 1, \dots, T$ are then summed together yielding the estimate of θ_w . We refer to the T auxiliary qubits as the *rotation register*. Technically, the information of θ_w is first passed on to each auxiliary qubit in the rotation register. The amplitude of the uniform superposition in each auxiliary qubit is then varied with respect to θ_w so that the probability of measuring the corresponding e_k is amplified.

Passing the information of θ_w to an auxiliary qubit in the rotation register can be achieved via the *controlled Grover operation*. The controlled Grover operation acts on the joint system of one auxiliary qubit in the rotation register and the *BGS register* that consists of the qubits for running BGS. In particular, the auxiliary qubit in the rotation register is initialized at the uniform superposition, i.e., $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Applying the controlled Grover operation is equivalent to applying the Grover operation to the BGS register only if the auxiliary qubit in the rotation register is at the state $|1\rangle$. In this way, a single controlled Grover operation rotates the auxiliary qubit by angle $2\theta_w$ (see Section A.1 in Supplementary Material for the technical details). Repeatedly applying different numbers of the controlled Grover operations is needed to solve different sub-problems. For example, applying 2^{k-1} controlled Grover operations is needed to estimate e_k . The inverse quantum Fourier transform (QFT) is then applied to each auxiliary qubit, the measurement of which is the solution of the corresponding sub-problem with a high probability. Finally, the measurement of the joint system of the rotation register and the BGS register is used to

estimate θ_w .

Using the quantum counting algorithm, we get the estimate of θ_w , denoted by $\hat{\theta}_w$. By Eq.(6), we estimate the number of oracle states via $\hat{D}_w = [D \sin^2(\hat{\theta}_w)]$, where $[.]$ denotes rounding to the nearest integer. Hence, the estimate of the number of iterations is

$$\hat{\tau}_w = \left\lceil \frac{\pi}{4 \arcsin(\sqrt{\hat{D}_w/D})} - \frac{1}{2} \right\rceil. \quad (10)$$

The estimation error of the quantum counting algorithm is established in the following lemma. The proof of this lemma is relegated in Section A.3 in Supplementary Material.

Lemma 2.1. *Let $\hat{\theta}_w$ be the estimate of the θ_w using the quantum counting algorithm with T qubits in the rotation register. Assuming that $\theta_w \in (0, \frac{\pi}{4})$. For any $\varepsilon \in (0, \frac{1}{4})$, we have*

$$\mathbb{P}\left(\frac{1}{\pi} |\hat{\theta}_w - \theta_w| > \varepsilon + \frac{1}{2^T}\right) < \frac{1}{2^{T+1}\varepsilon}. \quad (11)$$

The computational complexity of this quantum counting algorithm is $O(p2^T)$.

Remark. The condition of $\theta_w \in (0, \frac{\pi}{4})$ implies that there are less than half of the states have smaller BICs than the benchmark state. To satisfy this condition, we can take a simple approach to choose the benchmark state. Instead of randomly selecting one state as the benchmark state, we randomly select several states and choose the one with the smallest BIC as the benchmark state. For example, if m states are randomly selected, then the condition is satisfied with the probability $1 - \frac{1}{2^m}$, which is close to 1 for a moderate m .

Notice that the computational complexity of the quantum counting algorithm is of linear order in p and exponential order in T . Moreover, we will show that the computational cost of the quantum counting algorithm is of the lower order compared to the overall computational cost of the BGS algorithm.

2.4 BGS Algorithm

The details of BGS are summarized in Algorithm 1.

Algorithm 1 Bisection Grover's search

Input: An orthonormal basis set \mathcal{B} of size $D = 2^p$, a state loss function $g(\cdot)$ that maps a state in \mathcal{B} defined in Eq.(2), initial benchmark state $|b_w\rangle$, and error tolerance δ .

Initialization: Define a local evaluation function $S_w^g(\cdot)$ such that $S_w^g(|b_j\rangle) = 1$ if $g(|b_j\rangle) < g(|b_w\rangle)$ and $S_w^g(|b_j\rangle) = 0$ if $g(|b_j\rangle) \geq g(|b_w\rangle)$.

(a). Use the quantum counting algorithm to get the initial estimates $\hat{\theta}_w$ and $\hat{\tau}_w$.

repeat

repeat

(b1). Repeatedly apply Grover's operations defined in Eq.(7) for $\hat{\tau}_w$ times.

(b2). Measure the quantum register and denote the measurement result by $|b_{w^{new}}\rangle$.

until $S_w^g(|b_{w^{new}}\rangle) = 1$.

(c). Set $|b_w\rangle = |b_{w^{new}}\rangle$ and update $S_w^g(\cdot)$ accordingly.

(d). Use the quantum counting algorithm to get the updated estimates $\hat{\theta}_w$ and $\hat{\tau}_w$.

until $\sin(\hat{\theta}_w) \leq \delta$.

Output: The final benchmark state $|b_w\rangle$.

Note that at the beginning of each iteration, i.e., (b1) in Algorithm 1, the quantum state is always initialized at the uniform superposition of all states specified in Eq.(5). This is due to the no-cloning theorem (Wootters and Zurek, 1982), which states the impossibility of creating an independent and identical copy of an arbitrary unknown quantum state. This feature distinguishes the iterative algorithms in quantum computers from those on classic computers.

Furthermore, iterations in the BGS algorithm are terminated if $\sin(\hat{\theta}_w)$ is less than an

error tolerance δ . By Eq.(6), the value of $\sin(\theta_w)$ dictates the number of oracle states D_w . If $\sin(\theta_w) = 0$, there is no oracle state, i.e., $D_w = 0$. Hence, the benchmark state $|b_w\rangle$ must be the solution state $|b^*\rangle$ in the BGS algorithm. If $\sin(\theta_w) = \sqrt{1/D}$, there is one oracle state, i.e., $D_w = 1$. In that case, the BGS algorithm may yield a state with the second-best BIC value. The larger value of $\sin(\theta_w)$ is, the higher probability that the BGS algorithm yields an inaccurate estimate. However, smaller $\sin(\theta_w)$ implies more iteration steps and longer computation time. Thus one may set appropriate error tolerance δ in the iteration stopping rule to strike a balance between estimation accuracy and computational cost.

3 Theoretical Analysis

In this section, we present the theoretical analysis for the proposed BGS method. Theoretical results are established for estimation error and computational cost. The theorems also provide guidance for selecting the parameters, i.e., error tolerance δ and the rotation register's number of qubits T in Algorithm 1. All proofs for this section are relegated to Section C in Supplementary Material.

The following theorem is on the estimation error of BGS.

Theorem 3.1 (Estimation error of BGS). *Assume all the conditions in Lemma 2.1 are satisfied. If the error tolerance $\delta < 1/\sqrt{D}$, and the rotation register's number of qubits $T = \Omega\left(\log \frac{\log D}{(1/\sqrt{D}-\delta)}\right)$, the final output state $|b_w\rangle$ has the following error bound,*

$$\mathbb{P}(|b_w\rangle \neq |b^*\rangle) \lesssim \frac{\log D}{2^T (1/\sqrt{D} - \delta)}. \quad (12)$$

Remark. If the error tolerance violates $\delta \geq 1/\sqrt{D}$, Eq.(12) no longer holds, and the error probability $\mathbb{P}(|b_w\rangle \neq |b^*\rangle)$ is lower bounded by a constant close to $\frac{1}{2}$ (see Proposition C.5 in Supplementary Material).

Under a slightly stronger condition on δ , we can establish a more explicit relationship between the error bound and D as well as T .

Corollary 3.2. *Suppose all the conditions in Theorem 3.1 are satisfied. If we further require $\sqrt{D}\delta$ to be upper bounded away from 1, i.e., $\delta \leq (1 - c)/\sqrt{D}$ for some positive constant $c < 1$, we have*

$$\mathbb{P}(|b_w\rangle \neq |b^*\rangle) \lesssim \frac{\sqrt{D} \log D}{c 2^T}. \quad (13)$$

Remark. Since $D = 2^p$, we can rewrite Eq.(13) as $\mathbb{P}(|b_w\rangle \neq |b^*\rangle) \lesssim p 2^{\frac{p}{2}-T}/c$.

Note that the conditions $T = \Omega\left(\log \frac{\log D}{(1/\sqrt{D}-\delta)}\right)$ and $\delta \leq (1 - c)/\sqrt{D}$ together imply that $T = \Omega\left(\log(\sqrt{D} \log D)\right)$. Thus, Eq.(13) indicates that we can control the error probability $\mathbb{P}(|b_w\rangle \neq |b^*\rangle)$ to be arbitrarily small with a proper T of the order $\Theta(\log(\sqrt{D} \log D))$.

According to Theorem 3.1 and Corollary 3.2, we provide guidance for selecting the error tolerance δ and the number of qubits T in the rotation register. Note that the estimation error Eq.(13) diverges if T is too small, while the computation is too expensive if T is too large. Thus, we suggest setting $T = [\log(\sqrt{D} \log D) + 5]$ in practice. As for δ , intuitively, small δ needs more iterations for the algorithm to converge. Hence, we set $\delta = \frac{1}{2\sqrt{D}}$, which is bounded away from $\frac{1}{\sqrt{D}}$ but not too close to zero.

We further establish the following theorem regarding the computational cost of BGS.

Theorem 3.3 (Time and space complexity). *Under all the conditions of Lemma 2.1, if $T = \Theta(\log(\sqrt{D} \log D))$ and $\delta = \frac{1}{2\sqrt{D}}$, we have (1) the expected time complexity of Algorithm 1 is $O(\sqrt{D}(\log D)^3)$; (2) the space complexity of Algorithm 1 is $O(\log(\sqrt{D} \log D))$.*

Remark. Given the stochastic nature of quantum measurement outcomes and the heuristic-based iterative approach of BGS, the time complexity is random. Therefore, we are reporting the upper bound of the expected time complexity. On a quantum computer, the

time complexity of one rotation operation is $O(\log D)$ (Koike and Okudaira, 2009). Our proposed BGS algorithm needs $O(\sqrt{D}(\log D)^2)$ rotations in expectation.

Note that search algorithms in classic computers usually have a time complexity of $O(D)$. Thus, our BGS method provides a nearly quadratic speed-up. When D is large, this improvement over classic algorithms is very significant. Moreover, Grover's algorithm has a time complexity of $O(\sqrt{D} \log D)$. The time complexity of the BGS has a moderate increase over that of Grover's algorithm while without requiring the oracle state as input.

4 Empirical Performance

We assess the empirical performance of the BGS algorithm through simulation studies using IBM Quantum Experience. Since IBM Quantum Experience only offers seven qubits for public access, we evaluate the performance of BGS for varying the number of predictors up to the maximum in the real quantum computing environment. In addition, IBM provides a quantum simulator to mimic the real quantum computing environment. We thus evaluate the performance of BGS for a relatively large number of predictors using the simulator. We implemented the proposed BGS algorithm using a Qiskit Python development kit¹ provided by IBM Quantum Experience. The performance of the BGS algorithm is assessed in both linear regression models and weighted logistic regression models. We present the results of linear regression models here. The results of weighted logistic regression models are relegated to Section D in Supplementary Material.

For the linear regression model, replicated samples are generated according to

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \quad i = 1, \dots, n, \tag{14}$$

where $y_i \in \mathbb{R}$, $\mathbf{x}_i \in \mathbb{R}^p$, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top \in \mathbb{R}^p$, and $\epsilon_i \in \mathbb{R}$. We set $n = 1,000$, $\{\mathbf{x}_i\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim}$

¹<https://qiskit.org/>

$N_p(\mathbf{0}, \Sigma)$, the (i, j) th entry of Σ equals $0.7^{|i-j|}$. The first $[p/2]$ entries in β are set to 1, while the remaining entries are set to 0. We set $\sigma^2 = \frac{1}{3}\beta^\top \Sigma \beta$, and $\{\epsilon_i\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma^2)$. We generate 10 replicated samples for each $p \in \{3, 4, 5, 6, 7\}$ for real quantum computing, and 100 replicated samples for each $p = \{6, 7, \dots, 15\}$ for quantum simulating .

We apply our BGS algorithm to these samples and compared its performance with the best subset selection (BSS) via an exhaustive search.

4.1 Computational complexity

Recall that Theorem 3.3 states the time complexity of our proposed BGS algorithm is $O(\sqrt{D}(\log D)^3)$. In particular, the time complexity of Grover's rotation is $O(\log D)$, and BGS needs $O(\sqrt{D}(\log D)^2)$ times of rotations.

In Fig.3 (a), we plot the actual computation time of the BGS algorithm for a limited range of D conducted in the quantum computer. Next, we examine the empirical computational complexity of the proposed BGS for a larger range of D in the quantum simulator. Due to the fact that the quantum simulator is realized by a classical computer, some quantum steps are implemented through classic computational methods. For example, Grover's rotation is implemented via multiplying a $D \times D$ rotation matrix with a D -dimensional vector, which gives rise to the fact that the actual computational complexity of one Grover's rotation in the simulator is $O(D^2)$. Therefore, the real computation time of the BGS algorithm in a simulator does not faithfully represent its computation time in a quantum computer. For a fair comparison, we plot the number of rotations of the BGS algorithm as a surrogate of its computation time in Fig.3 (b).

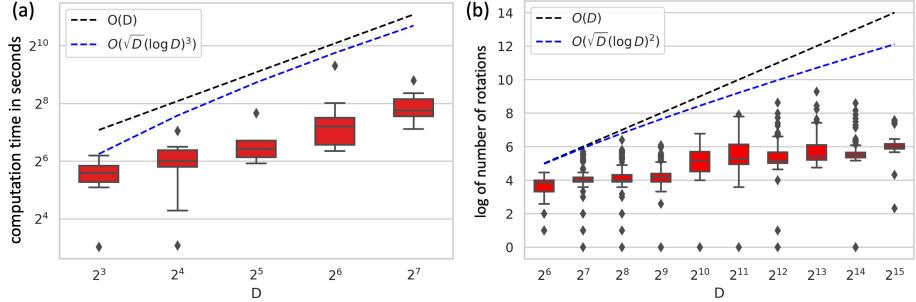


Figure 3: (a) The computation time of the BGS algorithm for the experiments conducted in the quantum computing environment. The boxplots of the actual computation time of the replicated experiments are plotted for varying D . The dashed black line denotes the order of the computation time of the exhaustive search $O(D)$. The dashed blue line represents the order of the theoretical computation time upper-bound $O(\sqrt{D}(\log D)^3)$. (b) The results of the experiments conducted in a quantum simulator. The boxplots of the log-transformed number of rotations of the replicated experiments are plotted for varying D . The dashed black line denotes the order $O(D)$. The dashed blue line represents the order of the theoretical upper bound of the number of rotations $O(\sqrt{D}(\log D)^2)$.

Both the results in Fig.3 (a) and (b) show that the computational cost of our proposed BGS gradually increases as D gets larger. However, we note that both the growth rate of the computation time and the growth rate of the numbers of rotations are indeed upper bounded as shown in our theoretical analysis. In Fig.3 (a), we notice that the computation time of running BGS is large even when the number of predictor sets D is small. This computational cost is primarily spent on the quantum machine warm-up.

We also notice that the computational cost has a large variability, which is attributed to the fact that BGS is a stochastic algorithm. Nevertheless, the advantage of the proposed BGS over the classic algorithm becomes more significant as D gets larger.

4.2 Performance of selection

To evaluate the subset selection accuracy of the BGS algorithm, we compare it with the best subset selection (BSS) via emulation in a classic computer. For the BSS, the subset with the smallest BIC is selected. One hundred replicated samples of size 1,000 are generated according to model Eq.(14) for each $p = \{6, 7, \dots, 15\}$. We report the frequency

of the true subset being selected among 100 replicates in Table 1.

We observe that the frequencies in Table 1 are close or equal to 100% for both BGS and BSS under all settings. The frequencies of the true subset being selected by the BGS are almost as good as those by the BSS.

Table 1: The percentage of the true subset being selected among 100 replicates by BSS and proposed BGS.

method	$p = 6$	$p = 7$	$p = 8$	$p = 9$	$p = 10$	$p = 11$	$p = 12$	$p = 13$	$p = 14$	$p = 15$
BSS	96%	100%	99%	98%	99%	99%	99%	99%	98%	99%
BGS	98%	99%	98%	98%	98%	98%	97%	99%	97%	98%

5 Application

We evaluate the performance of the BGS algorithm in two CITE-seq studies. In each study, we randomly divide the whole dataset into a training dataset, consisting of CITE-seq expressions of 75% of the cells, and a testing dataset consisting of expressions of 25% of the cells. We apply BGS to the training dataset to fit models and select the best subsets, and then conduct predictions in the testing dataset. We repeat this experiment 100 times. In each experiment, we also apply the differential expression methods in the Seurat package V5 (Hao et al., 2023) with six options: Wilcoxon Rank Sum test (Seurat-wilcox), Student’s t-test (Seurat-t), likelihood-ratio test, (Seurat-bimod), ROC analysis (Seurat-roc), a logistic regression-based method (Seurat-LR) and a hurdle model-based method (Seurat-MAST), and the differential expression methods in the SCANPY package V1.9.2 (Wolf et al., 2018) with three options: Wilcoxon Rank Sum test (SCANPY-wilcox), Student’s t-test (SCANPY-t), and a logistic regression-based method (SCANPY-LR). Further details can be found in Sections E and F in Supplementary Material.

5.1 Identification of ADTs associated with the marker gene of interest

In a study of the human cord blood mononuclear cells (CBMC), Stoeckius et al. (2017) profiled a total of 8,617 cells, including 1,727 CD14+ Monocytes cells, using CITE-seq. For each cell, 13 cell-surface protein markers are quantified via sequencing their corresponding antibody-derived tags (ADTs), and RNA expression levels are measured. Among the sequenced RNA, CD14 RNA is an important immune response gene and has a high expression level in some monocytes (Rawat et al., 2021). We are interested in the association between the expression level of the CD14 RNA and these 13 ADTs in the CD14+ Monocytes cells. We pre-process the expression data using the Seurat package V5 (details are provided in Section E in Supplementary Material).

We identify the ADTs associated with CD14 RNA through a regression approach (Zhong et al., 2005; Zamdborg and Ma, 2009; Liu et al., 2024). In particular, we regress the expression level of CD14 RNA on the expression levels of 13 ADTs. We aim to identify the best subset of ADTs for explaining the variation of CD14 RNA. For each candidate subset of the 13 ADTs, we fit the corresponding linear regression model to the training dataset and calculate the BIC. We then apply the BGS algorithm to select the best subsets.

We also select the subset of ADTs using the differential expression methods of the Seurat and SCANPY packages. Using the subset of ADTs, we fit a linear regression for CD14 RNA on the training set, and we evaluate the methods by the mean square error between the predicted and observed values of CD14 RNA expression level on the testing dataset. For a fair comparison, we ensure that all methods have the same number of selected ADTs, ranging from 1 to 12. The natural logarithms of the ratios of the mean squared errors (MSEs) and BIC of Seurat and SCANPY relative to BGS are computed.

The resulting log MSE ratios and BIC ratios are presented in Fig.4 (a) and (b) through boxplots. Notably, for all numbers of selected ADTs, most boxplots are situated above zero. This observation indicates that BGS consistently outperforms the other two methods across various scenarios.

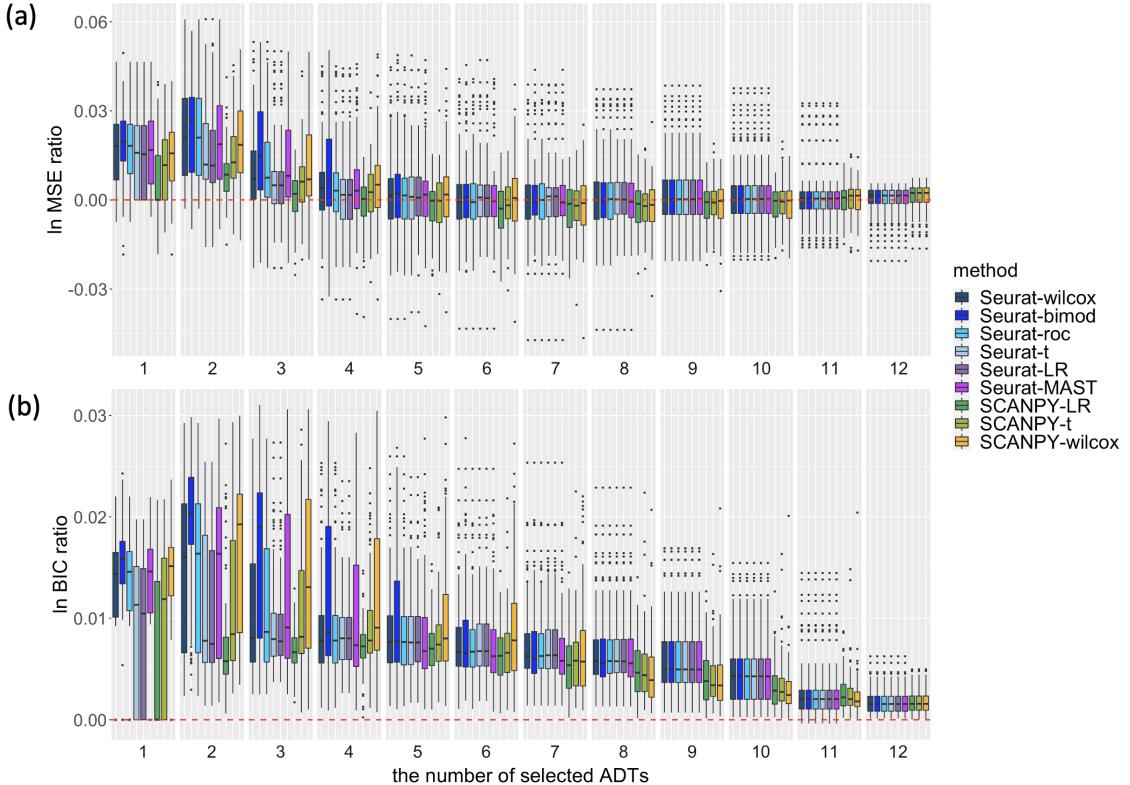


Figure 4: Comparison of the performance of BGS, Seurat, and SCANPY on the CD14+ Monocytes cells from CBMC data. The evaluation is based on the mean squared error (MSE) and BIC between the predicted and observed values of CD14 RNA expression level across 100 testing datasets. (a) The x-axis represents the number of selected ADTs, while the y-axis represents the log ratio of the MSE for each method compared to the MSE of the proposed BGS algorithm. The red dashed line indicates a ratio of zero. (b) The x-axis represents the number of selected ADTs, while the y-axis represents the log ratio of the BIC for each method compared to the BIC of the proposed BGS algorithm.

We now examine the identified ADTs associated with CD14 RNA by the BGS algorithm. The best subset of ADTs is selected among 2^{13} possible candidate subsets. We identify the two most significant ADTs associated with CD14 RNA. By fitting the linear regression model using CD4 and CD14 as the predictors on all CD14+ Monocytes cells, the coefficient

of CD4 is -0.1388 , and the coefficient of CD14 is 0.1333 . Notice that CD14 ADT is positively associated with CD14 RNA, which is consistent with the fact that CD14 RNA is the coding gene for the CD14 protein. Additionally, we observe negative relationships between CD4 and CD14 RNA. This outcome is consistent with the existing literature. A study on the clinical outcomes of dedifferentiated liposarcoma patients ([Schroeder et al., 2021](#)) has shown a negative relationship between CD4 and CD14.

5.2 Panel design for immune cell type identification

In a CITE-seq experiment, [Hao et al. \(2021\)](#) reported the measurement of expression levels for 228 ADTs and 33,538 RNAs in 161,764 peripheral blood mononuclear cells (PBMCs). These cells encompassed 57 distinct cell types, which were identified and annotated based on differentially expressed RNAs. Additionally, 10 ADT markers were identified for each cell type.

Our study aims to evaluate the performance of our proposed BGS algorithm in identifying panels of ADT markers for classifying various cell types. To ensure a fair comparison with the results reported in [Hao et al. \(2021\)](#), we restrict the panel size to be smaller than 10. We focus on classifying cells for the major cell types, specifically those with more than 500 sequenced cells, resulting in 41 cell types of interest. For each cell type of interest, we assign that particular cell type a label of one, while labeling all other cell types as zero. Consequently, we encounter highly unbalanced cell samples, with significantly fewer cells labeled as one compared to those labeled as zero. To mitigate potential biases arising from unbalanced samples, we employ weighted logistic regression models ([King and Zeng, 2001](#)) on the training dataset for each candidate panel, accounting for the sample weights. We calculate the BIC and employ our BGS algorithm to identify the best panel with the opti-

mal BIC. For each of the 41 cell types, we also utilize the Seurat and SCANPY methods to identify the differentially expressed ADTs specific to that cell type, using the training dataset. These selected ADTs are then used to fit weighted logistic regression models. Finally, we assess the performance of the fitted models on the testing dataset by calculating metrics such as the area under the ROC curve (AUC), sensitivity, and specificity.

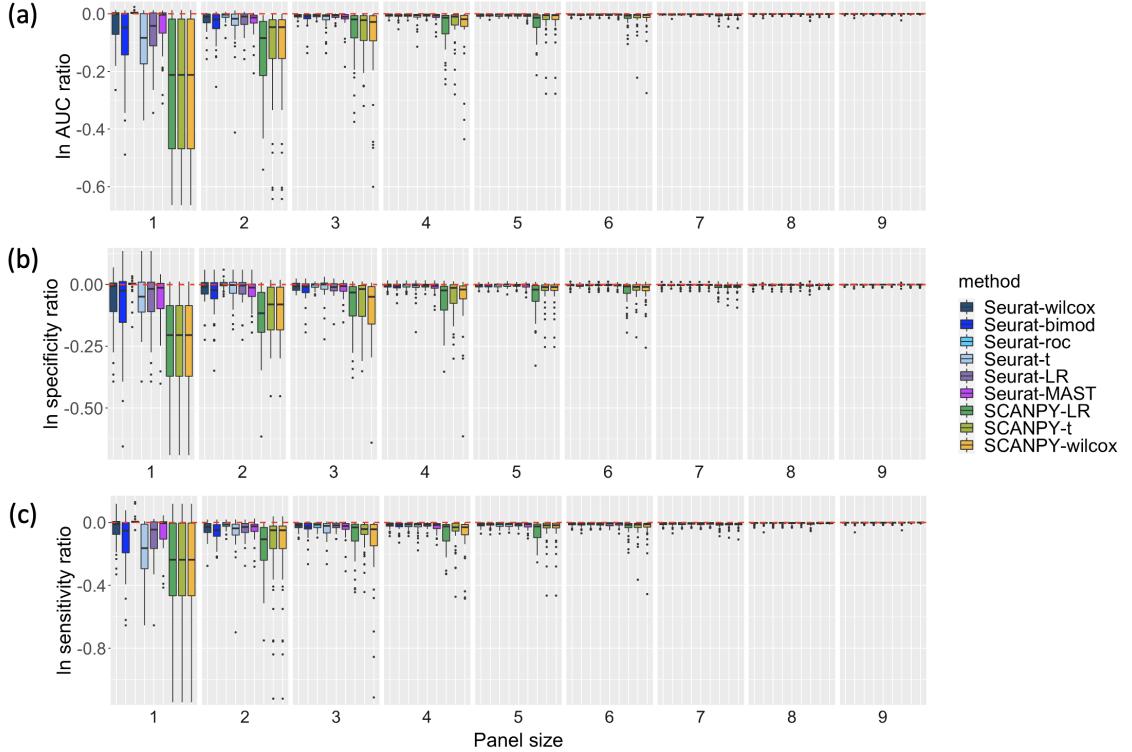


Figure 5: The prediction performance of the 41 cell types is compared for the Seurat and SCANPY methods with respect to the BGS algorithm. The panel size is shown on the X-axis, while the Y-axis represents the log ratio of the evaluation metric of each method compared to that of the proposed BGS algorithm. The evaluation metrics reported are AUC in (a), specificity in (b), and sensitivity in (c). The red dashed line in each plot indicates a ratio of zero, serving as a reference point for comparison.

For each of the 41 cell types, we calculate the log ratios of the evaluation metrics for the Seurat and SCANPY methods compared to those of the proposed BGS algorithm. A log ratio less than zero indicates that BGS outperforms the corresponding method. The results are visualized through boxplots in Fig.5 (a), (b), and (c), for AUC, sensitivity, and specificity, respectively. To maintain a reasonable scale of the figure, the outliers below

the lower bound of the Y-axis are omitted. Across all three metrics and considering the four different panel sizes, the majority of the boxplots are positioned below zero. This observation indicates that BGS consistently outperforms the other methods across a wide range of scenarios.

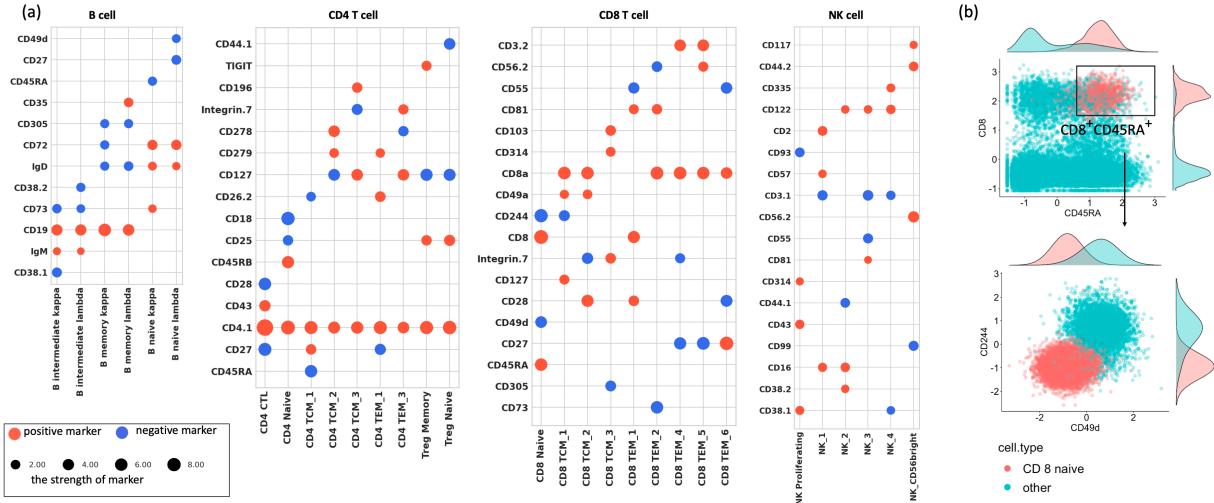


Figure 6: The best four-marker panels selected by the BGS algorithm. (a) Dot plots of the best panels are presented. Each cell type is represented by a column, with the cell type name displayed in the bottom margin. Within each column, four dots are positioned to represent the best four markers, and their corresponding names are indicated in the left margin. The dots are color-coded as red or blue, representing the positive or negative coefficients of the markers in the corresponding logistic regression model. The size of each dot reflects the strength of the corresponding marker, determined by the absolute value of the coefficient. (b) The best four-marker panel for CD8 naive cells (CD8, CD45RA, CD49d, CD244) is shown. The marginal distributions of these markers' expressions are displayed on the top and right margins. Each dot represents a cell, and the color of the dot indicates the cell type. The upper subfigure illustrates the scatter plot of CD8 and CD45RA for all PBMC cells, while the lower subfigure shows the scatter plot of CD49d and CD244 specifically for the screened CD8⁺CD45RA⁺ cells.

Fig.6 (a) showcases dot plots depicting the best four-marker panels selected by the BGS algorithm for 30 different cell types, which can be categorized into four coarse cell types: B cells, CD4 T cells, CD8 T cells, and NK cells. Upon examining the panels within each coarse cell type, we observe that several best panels share common markers. For instance, in the case of CD4 T cells, all best panels include CD4.1 as a marker with a large strength. However, when comparing panels across different coarse cell types, we find that the best

panels exhibit distinct marker compositions. In addition, the panels identified by our BGS algorithm are consistent with the literature. For instance, $CD19^+IgD^-$ are well-known markers for B memory cells; IgD^+ is a marker for B naive cells ([Kaminski et al., 2012](#)); $CD4^+CD25^+CD127^-$ are markers for Treg cells ([Liu et al., 2006](#)), and $CD3^-CD122^+$ are markers for NK cells ([Farag and Caligiuri, 2006](#)).

Moreover, we make an intriguing discovery regarding the combined use of CD49d and CD244 as highly informative markers, in addition to the typical markers $CD8^+CD45RA^+$ ([Nguyen et al., 2016](#)), for identifying CD8 naive T cells. This novel finding has not been previously reported. Although previous studies consistently demonstrate that CD8 naive cells typically exhibit low expression levels of both CD49d and CD244, which may increase upon activation, the specific roles and implications of CD49d and CD244 in immune responses have not been actively studied until very recently ([White et al., 2017](#); [Berard and Tough, 2002](#); [Agresta et al., 2018](#)). The up-regulation of CD49d facilitates efficient access to inflamed peripheral tissues and enhances responsiveness to inflammatory signals, while up-regulated CD244 signaling activates CD8 naive T cells. However, using CD244 or CD49d alone may not be sufficient as informative markers, as illustrated by the marginal distributions of CD49d and CD244 for the $CD8^+CD45RA^+$ cell population in Fig.6 (b). In contrast, our selected panel ($CD8^+CD45RA^+CD244^-CD49d^-$) demonstrates high accuracy in screening CD8 naive cells, as evidenced by an average AUC of 0.998 in the test datasets across 100 replications. Some additional results for this study are reported in Section H in Supplementary Material.

Note that the design of the experimental setup is constrained by the hardware currently available. Consequently, the experiments are conducted on relatively low-dimensional problems. Nonetheless, we expect that our algorithm will scale effectively and yield more sig-

nificant insights as quantum computing technology advances.

6 Conclusion

Given that many statistical problems involve computationally intensive tasks, statisticians are particularly intrigued by the potential of quantum computers (Wang and Song, 2020; Wang and Liu, 2022). Consequently, a natural question is whether these computers will benefit the statisticians in solving some statistics or data science problems. If the answer is affirmative, what kind of statistics problems should statisticians resort to quantum computers? Unfortunately, the general answer to this question remains elusive.

In this paper, we answer this question by showing the benefit of quantum computing in single-cell biology problems that statisticians have been working on extensively (Agarwal et al., 2020). In particular, we developed the bisection Grover’s search algorithm for selecting the best subset and demonstrated its advantages in identifying the ADTs associated with targeted genes and designing panels for cell type identifications. We established the theoretical properties of our proposed algorithm. We also demonstrated the empirical performance of the proposed algorithm in a NISQ device and a simulator.

It is worth noting that the BGS algorithm is highly versatile and flexible. It has the potential to be seamlessly integrated into other machine learning methods, such as deep neural networks, which are known for their remarkable fitting or expressive power and predictive capabilities (Beer et al., 2020; Abbas et al., 2021). This integration enables researchers to harness the expressive power offered by machine learning models while capitalizing on the computational efficiencies provided by quantum computing to tackle more complex problems.

One key feature that distinguishes quantum algorithms from classical algorithms is

quantum parallelism, which enables us to develop a unique approach to addressing multi-modal biological problems. The potential application of quantum algorithms in the realm of biological problems extends far beyond the scope presented here. For instance, quantum algorithms hold promise for more effectively analyzing spatial transcriptomics data, unlocking new insights and capabilities in the field.

7 Data, Materials, and Software Availability

The human cord blood mononuclear cells dataset (Stoeckius et al., 2017) is available at the NCBI Gene Expression Omnibus (GEO; <https://www.ncbi.nlm.nih.gov/geo/>) with accession no.GSE100866. The peripheral blood mononuclear cells dataset (Hao et al., 2021) is available at New York Genome Center (<https://atlas.fredhutch.org/nygc/multimodal-pbmc/>). Source code for the BGS algorithm is freely available on GitHub (<https://github.com/StatCYK/BGS-for-CITEseq/>).

8 Acknowledgements

The authors would like to thank the editor, associate editor, and the three committed reviewers for their careful and constructive review. This work was partially supported by the U.S. National Science Foundation DMS-1925066, DMS-1903226, DMS-2124493, DMS-2311297, DMS-2319279, DMS-2318809 and the National Institutes of Health NIH R01GM152814.

9 Disclosure Statement

The authors report there are no competing interests to declare.

References

- Abbas, A., D. Sutter, C. Zoufal, A. Lucchi, A. Figalli and S. Woerner (2021). The power of quantum neural networks. *Nature Computational Science* 1(6), 403–409.
- Agarwal, D., J. Wang and N. R. Zhang (2020). Data denoising and post-denoising corrections in single cell rna sequencing. *Statistical Science* 35(1), 112–128.
- Agresta, L., K. H. Hoebe and E. M. Janssen (2018). The emerging role of cd244 signaling in immune cells of the tumor microenvironment. *Frontiers in immunology* 9, 2809.
- Arute, F., K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature* 574(7779), 505–510.
- Beer, K., D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann et al. (2020). Training deep quantum neural networks. *Nature communications* 11(1), 808.
- Bennett, C. H., E. Bernstein, G. Brassard and U. Vazirani (1997). Strengths and weaknesses of quantum computing. *SIAM Journal on Computing* 26(5), 1510–1523.
- Berard, M. and D. F. Tough (2002). Qualitative differences between naive and memory t cells. *Immunology* 106(2), 127.
- Biamonte, J., P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe and S. Lloyd (2017). Quantum machine learning. *Nature* 549(7671), 195–202.
- Brassard, G., P. Hoyer, M. Mosca and A. Tapp (2002). Quantum amplitude amplification and estimation. *Contemporary Mathematics* 305, 53–74.
- Brassard, G., P. Høyer and A. Tapp (1998). Quantum counting. In *International Colloquium on Automata, Languages, and Programming*, pp. 820–831. Springer.
- Buhrman, H., R. Cleve, J. Watrous and R. De Wolf (2001). Quantum fingerprinting. *Physical review letters* 87(16), 167902.

Castelvecchi, D. (2023). IBM releases first-ever 1,000-qubit quantum chip. *Nature* 624(7991), 238–238.

Chakraborty, S., S. H. Shaikh, A. Chakrabarti and R. Ghosh (2020). A hybrid quantum feature selection algorithm using a quantum inspired graph theoretic approach. *Applied Intelligence* 50(6), 1775–1793.

Farag, S. S. and M. A. Caligiuri (2006). Human natural killer cell development and biology. *Blood reviews* 20(3), 123–137.

Ferrer-Font, L., S. J. Small, B. Lewer, K. R. Pilkington, L. K. Johnston, L. M. Park et al. (2021). Panel optimization for high-dimensional immunophenotyping assays using full-spectrum flow cytometry. *Current Protocols* 1(9), e222.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, New York, NY, USA, pp. 212–219. Association for Computing Machinery.

Hao, Y., S. Hao, E. Andersen-Nissen, W. M. Mauck, S. Zheng, A. Butler et al. (2021). Integrated analysis of multimodal single-cell data. *Cell* 184(13), 3573–3587.

Hao, Y., T. Stuart, M. H. Kowalski, S. Choudhary, P. Hoffman, A. Hartman et al. (2023). Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, 1–12.

Harrow, A. W., A. Hassidim and S. Lloyd (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters* 103(15), 150502.

He, Z., L. Li, Z. Huang and H. Situ (2018). Quantum-enhanced feature selection with forward selection and backward elimination. *Quantum Information Processing* 17, 1–11.

Hughston, L. P., R. Jozsa and W. K. Wootters (1993). A complete classification of quantum ensembles having a given density matrix. *Physics Letters A* 183(1), 14–18.

- Kaminski, D. A., C. Wei, Y. Qian, A. F. Rosenberg and I. Sanz (2012). Advances in human b cell phenotypic profiling. *Frontiers in immunology* 3, 302.
- King, G. and L. Zeng (2001). Logistic regression in rare events data. *Political Analysis* 9(2), 137–163.
- Koike, T. and Y. Okudaira (2009, 10). Time complexity and gate complexity. *Physical Review. A* 82.
- Li, Y., R.-G. Zhou, R. Xu, J. Luo, W. Hu and P. Fan (2022). Implementing graph-theoretic feature selection by quantum approximate optimization algorithm. *IEEE Transactions on Neural Networks and Learning Systems*.
- Liu, W., A. L. Putnam, Z. Xu-Yu, G. L. Szot, M. R. Lee, S. Zhu et al. (2006). Cd127 expression inversely correlates with foxp3 and suppressive function of human cd4+ t reg cells. *The Journal of experimental medicine* 203(7), 1701–1711.
- Liu, Y., Y. Chen, H. Lu, W. Zhong, G.-C. Yuan and P. Ma (2024). Orthogonal multi-modality integration and clustering in single-cell data. *BMC bioinformatics* 25(1), 164.
- Martínez, C. and S. Roura (1998). Randomized binary search trees. *Journal of the ACM (JACM)* 45(2), 288–323.
- Mücke, S., R. Heese, S. Müller, M. Wolter and N. Piatkowski (2023). Feature selection on quantum computers. *Quantum Machine Intelligence* 5(1), 11.
- Nguyen, H. H., T. Kim, S. Y. Song, S. Park, H. H. Cho, S.-H. Jung et al. (2016). Naïve cd8+ t cell derived tumor-specific cytotoxic effectors as a potential remedy for overcoming tgf- β immunosuppression in the tumor microenvironment. *Scientific reports* 6(1), 1–10.
- Nielsen, M. A. and I. L. Chuang (2010). *Quantum Computation and Quantum Information*. Cambridge University Press.
- Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum* 2, 79.

- Rawat, K., A. Pal, S. Banerjee, A. Pal, S. C. Mandal and S. Batabyal (2021). Ovine cd14—an immune response gene has a role against gastrointestinal nematode haemonchus contortus—a novel report. *Frontiers in immunology* 12.
- Schrödinger, E. (1935). Discussion of probability relations between separated systems. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Volume 31, pp. 555–563. Cambridge University Press.
- Schroeder, B. A., N. A. LaFranzo, B. J. LaFleur, R. M. Gittelman, M. Vignali, S. Zhang et al. (2021). Cd4+ t cell and m2 macrophage infiltration predict dedifferentiated liposarcoma patient outcomes. *Journal for immunotherapy of cancer* 9(8).
- Schuld, M., I. Sinayskiy and F. Petruccione (2016). Prediction by linear regression on a quantum computer. *Physical Review A* 94(2), 022342.
- Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 461–464.
- Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review* 41(2), 303–332.
- Stoeckius, M., C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow et al. (2017). Large-scale simultaneous measurement of epitopes and transcriptomes in single cells. *Nature methods* 14(9), 865.
- Turati, G., M. F. Dacrema and P. Cremonesi (2022). Feature selection for classification with qaoa. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 782–785. IEEE.
- Von Dollen, D., F. Neukart, D. Weimer and T. Bäck (2021). Quantum-assisted feature selection for vehicle price prediction modeling. *arXiv preprint arXiv:2104.04049*.
- Wang, Y. (2022). When quantum computation meets data science: Making data science quantum. *Harvard Data Science Review* 4(1).

Wang, Y. and H. Liu (2022). Quantum computing in a statistical context. *Annual Review of Statistics and Its Application* 9, 479–504.

Wang, Y. and X. Song (2020). Quantum science and quantum technology. *Statistical Science* 35(1), 51–74.

Weigold, M., J. Barzen, F. Leymann and M. Salm (2020). Data encoding patterns for quantum computing. In *Proceedings of the 27th Conference on Pattern Languages of Programs*, pp. 1–11.

White, J. T., E. W. Cross and R. M. Kedl (2017). Antigen-inexperienced memory cd8+ t cells: where they come from and why we need them. *Nature Reviews Immunology* 17(6), 391–400.

Wolf, F. A., P. Angerer and F. J. Theis (2018). Scanpy: large-scale single-cell gene expression data analysis. *Genome biology* 19, 1–5.

Wootters, W. K. and W. H. Zurek (1982). A single quantum cannot be cloned. *Nature* 299(5886), 802–803.

Wu, Y., W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen et al. (2021). Strong quantum computational advantage using a superconducting quantum processor. *Physical review letters* 127(18), 180501.

Zamdborg, L. and P. Ma (2009). Discovery of protein–dna interactions by penalized multivariate regression. *Nucleic acids research* 37(16), 5246–5254.

Zhong, H.-S., H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo et al. (2020). Quantum computational advantage using photons. *Science* 370(6523), 1460–1463.

Zhong, W., P. Zeng, P. Ma, J. S. Liu and Y. Zhu (2005, 09). RSIR: regularized sliced inverse regression for motif discovery. *Bioinformatics* 21(22), 4169–4175.

Supplementary Material for “Bisection Grover’s Search Algorithm and Its Application in Analyzing CITE-seq Data”

A Notation and Technical Preliminaries

We present some essential notations and definitions, including the preliminaries of quantum computing that are minimally sufficient to serve our needs. More comprehensive treatment of quantum computing can be found in [Nielsen and Chuang \(2010\)](#).

A.1 Notations and definitions

Asymptotic inequality notations. Let (a_n) and (b_n) be two sequences of real numbers, and at least one sequence is not always zero. We say $a_n \sim b_n$ if $\lim_{n \rightarrow \infty} a_n/b_n = 1$. We say $a_n \gtrsim b_n$ if $a_n \sim \max\{a_n, b_n\}$, and $a_n \lesssim b_n$ if $a_n \sim \min\{a_n, b_n\}$.

O , Ω and Θ notations. The notation $A_n = O(a_n)$ means that for some constant c and n_0 , $A_n \leq c a_n$ for all $n \geq n_0$. The notation $A_n = \Omega(a_n)$ means that for some constant c and n_0 , $A_n \geq c a_n$ for all $n \geq n_0$. The notation $A_n = \Theta(a_n)$ means that $A_n = O(a_n)$ and $A_n = \Omega(a_n)$.

Qubit and superposition. Qubit is the fundamental unit of information in quantum computing. A qubit has two possible states denoted by $|0\rangle$ and $|1\rangle$ using the customary Dirac notation. Different from a classic bit that can be in one state only, a qubit can be in both $|0\rangle$ and $|1\rangle$ simultaneously. Mathematically, the state of a qubit can be described by a linear combination of $|0\rangle$ and $|1\rangle$, also called a superposition,

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle, \quad (\text{A.1})$$

where complex coefficients α_0 and α_1 (called *amplitudes*) satisfy $|\alpha_0|^2 + |\alpha_1|^2 = 1$, and $|\cdot|$ is the norm of a complex number. These two special states $|0\rangle$ and $|1\rangle$ are called the computational basis states.

Quantum register. A quantum register is a quantum system comprising multiple qubits. In general, the p -qubit register has $D \triangleq 2^p$ basis states, $|b_j\rangle$, $j = 0, \dots, D-1$, where b_j is the binary representation of the integer j . Every superposition $|\psi\rangle$ can be represented as

$$|\psi\rangle = \sum_{j=0}^{D-1} \alpha_j |b_j\rangle, \quad (\text{A.2})$$

where amplitudes $\alpha_0, \dots, \alpha_{D-1}$ satisfy $\sum_{j=0}^{D-1} |\alpha_j|^2 = 1$.

Measurement and collapse. By Schrödinger-HJW theorem ([Schrödinger, 1935](#); [Hughston et al., 1993](#)), if a superposition $|\psi\rangle$ in Eq. (A.2) is measured, it randomly collapses to one of the basis states in $\{|b_0\rangle, |b_1\rangle, \dots, |b_{D-1}\rangle\}$. The probability that we observe $|b_j\rangle$ is $|\alpha_j|^2$ for $j = 0, \dots, D-1$.

Quantum gates and quantum circuits. The building blocks of quantum algorithms are quantum gates. Mathematically, a quantum gate for a p -qubit quantum system can be represented by a $2^p \times 2^p$ unitary matrix, and the quantum operation can be represented by multiplying the corresponding unitary matrix with the state vector of the qubits. For example, the Hadamard gate acts on a single qubit and can be represented by the Hadamard matrix,

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (\text{A.3})$$

which enables the rotations of the input states. Quantum circuits consist of quantum gates with associated connections to perform quantum computations.

Hilbert space. For a p -qubit quantum system (register), the basis states correspond to the standard basis vectors in the $D = 2^p$ -dimensional vector spaces. For example, $|0\rangle$

can be represented by $(1, 0)^\top$ and $|1\rangle$ can be represented by $(0, 1)^\top$. The superposition can also be described by a 2^p -dimensional unit vector. Thus, the mathematical foundation of quantum computing is built upon a state-space postulate, which describes the state of a system by a unit vector in a Hilbert space \mathcal{H} . For the Hilbert space \mathcal{H} , the dual Hilbert space \mathcal{H}^* is defined as the set of linear maps $\mathcal{H} \rightarrow \mathbb{C}$, where \mathbb{C} is the complex space. Then, for a vector $|a\rangle \in \mathcal{H}$, we define $\langle a| \in \mathcal{H}^*$ as the dual vector of $|a\rangle$ by $\langle a| (|b\rangle) \equiv (\langle a|, |b\rangle)$ for any $|b\rangle \in \mathcal{H}$, where (\cdot, \cdot) is the inner product of two vectors. Thus, \mathcal{H} and \mathcal{H}^* together naturally induce an inner product for $|a\rangle$ and $|b\rangle$ as $\langle a|b\rangle \equiv (\langle a|, |b\rangle)$.

\mathbf{U}^m operation. Let \mathbf{U} be a unitary operation in Hilbert space \mathcal{H} . For a positive integer m , we define a \mathbf{U}^m operation, which is repeatedly applying \mathbf{U} operation m times. We have

$$\mathbf{U}^m(|\psi\rangle) = \underbrace{\mathbf{U}(\cdots(\mathbf{U}(|\psi\rangle))\cdots)}_m,$$

where $|\psi\rangle \in \mathcal{H}$.

Controlled unitary operation. Let \mathbf{U} be a unitary operation in a Hilbert space \mathcal{H} . Let \mathcal{H}_0 be the Hilbert space of a one-qubit quantum system. A controlled- \mathbf{U} operation is defined in Hilbert space $\mathcal{H}_0 \otimes \mathcal{H}$ as

$$\text{controlled-}\mathbf{U} ([\alpha_0 |0\rangle + \alpha_1 |1\rangle] \otimes |\psi\rangle) = \alpha_0 |0\rangle \otimes |\psi\rangle + \alpha_1 |1\rangle \otimes \mathbf{U}(|\psi\rangle), \quad (\text{A.4})$$

where $|\psi\rangle \in \mathcal{H}$. For this controlled- \mathbf{U} operation, the unitary operation \mathbf{U} applies to $|\psi\rangle$ only when the one-qubit system is in state $|1\rangle$. We refer to the one-qubit quantum system as the control qubit and the register of $|\psi\rangle$ as the target register.

Density matrix. The density matrix is a mathematical representation of a quantum state, including both pure and mixed states. It is a Hermitian, positive-semidefinite operation that contains all the statistical information about a quantum system. Mathematically, for a quantum system associated with a Hilbert space \mathcal{H} , the density matrix ρ is defined

as:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$$

Where p_i represents the probability of the system being in state $|\psi_i\rangle$, $|\psi_i\rangle$ is a quantum state vector in \mathcal{H} , and $|\psi_i\rangle\langle\psi_i|$ is an outer product of the state vector.

The density matrix allows for the representation of both pure states (when ρ is a projector onto a single state) and mixed states (when ρ is a convex combination of pure states).

Partial trace. The partial trace is a mathematical operation used when dealing with composite quantum systems, where each subsystem is described by its own Hilbert space. Given a composite system A, B with associated Hilbert spaces \mathcal{H}_A and \mathcal{H}_B , the partial trace operation allows researchers to extract information about one subsystem while tracing out (ignoring) the other subsystem. Mathematically, for a density matrix ρ_{AB} describing the composite system, the partial trace over subsystem B is denoted as $\text{Tr}_B(\rho_{AB})$ and is defined as:

$$\text{Tr}_B [\rho_{AB}] := \sum_j (\mathbf{I}_A \otimes \langle j |_B) \rho_{AB} (\mathbf{I}_A \otimes |j\rangle_B)$$

where $|j\rangle$ spans the basis of \mathcal{H}_B , \mathbf{I}_A is the identity operation on \mathcal{H}_B , and the trace operation sums over all possible basis states of subsystem B . The result, $\text{Tr}_B(\rho_{AB})$, is a density matrix that describes the reduced state of subsystem A , taking into account the correlations and entanglement between A and B . It effectively “traces out” the degrees of freedom of subsystem B , allowing researchers to analyze subsystem A in isolation.

A.2 Review of Grover’s algorithm

The most popular quantum searching algorithm is Grover’s algorithm, which is quadratically faster than classic algorithms. We shall now present it in detail within the context of our single-cell problems. In our problems, we aim to identify a model that minimizes Eq.(1)

with respect to all possible choices of the subset \mathcal{A} , which are denoted by $\mathcal{A}_0, \dots, \mathcal{A}_{D-1}$.

All subsets are indexed by $0, 1, \dots, D - 1$. In quantum computing, each index is regarded as a state, which can be represented by a unit vector in an orthonormal basis $\mathcal{B} = \{|b_j\rangle\}_{j=0}^{D-1}$ of a D -dimensional Hilbert space \mathcal{H} . Thus, each basis state corresponds to a subset. We call the state corresponding to the best subset as the oracle state.

Geometric interpretation of Grover's algorithm. Grover's algorithm can be more easily understood from a geometric perspective. Denote the oracle state by $|b^*\rangle$ and the average of all non-oracle states by $|\zeta\rangle = \frac{1}{\sqrt{D-1}}(-|b^*\rangle + \sum_{j=0}^{D-1} |b_j\rangle)$. Since there is no prior information on which state is the oracle state, Grover's algorithm initializes a uniform superposition $|\psi_0\rangle$ that is the equally weighted average of all states $\{|b_j\rangle\}_{j=0}^{D-1}$, i.e.,

$$|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |b_j\rangle. \quad (\text{A.5})$$

Denote the angle between the initial superposition $|\psi_0\rangle$ and $|\zeta\rangle$ by θ . It is easy to see that $\sin \theta = \frac{1}{\sqrt{D}}$.

Grover's algorithm rotates the superposition until it coincides with $|b^*\rangle$. The rotation of Grover's algorithm contains two operations. The first operation reflects $|\psi_0\rangle$ with respect to $|\zeta\rangle$ to get $F|\psi_0\rangle$. The second operation reflects $F|\psi_0\rangle$ with respect to $|\psi_0\rangle$ to get $GF|\psi_0\rangle$. After the two operations, the resulting new superposition is denoted by $|\psi_1\rangle$. Consequently, the angle between $|\psi_1\rangle$ and $|\zeta\rangle$ is 3θ . Grover's algorithm then repeats the above two operations for the superposition, i.e., a reflection with respect to $|\zeta\rangle$ and then a reflection with respect to $|\psi_0\rangle$. After the m th rotation, the angle between the current superposition $|\psi_m\rangle$ and $|\zeta\rangle$ is $(2m + 1)\theta$. The rotation terminates if $|\psi_m\rangle$ coincides with the oracle state $|b^*\rangle$, i.e., $|\psi_m\rangle = |b^*\rangle$, which implies that

$$(2m + 1)\theta = \pi/2. \quad (\text{A.6})$$

When the number of candidate sets D is large, i.e., θ is small, we can approximate the

angle by

$$\theta \approx \sin \theta = \frac{1}{\sqrt{D}}. \quad (\text{A.7})$$

Substituting Eq. (A.7) into Eq. (A.6), we have $(2m + 1)/\sqrt{D} = \pi/2$, which yields m is approximately $\sqrt{D}\pi/4$. That is, the number of rotations for Grover's algorithm to find the best subset is in the order of $O(\sqrt{D})$. In contrast, the classic search algorithms need $O(D)$ operations. Thus Grover's algorithm substantially improves the computational complexity of classic algorithms. Moreover, it is shown that the computational complexity of Grover's algorithm is optimal, up to a constant, for all possible quantum search algorithms (Bennett et al., 1997).

Mathematical formulation of Grover's algorithm. Mathematically, Grover's algorithm iteratively amplifies the amplitude of the oracle state (Nielsen and Chuang, 2010). In Grover's algorithm, we have a quantum oracle evaluation function $S^*(\cdot)$, which can recognize the oracle state $|b^*\rangle$ to the search problem. That is,

$$S^*(|b_j\rangle) = \begin{cases} 1 & \text{if } |b_j\rangle = |b^*\rangle, \\ 0, & \text{otherwise.} \end{cases}$$

Recall that, Grover's algorithm is initialized at the uniform superposition,

$$|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |b_j\rangle \equiv c_0 |b^*\rangle + d_0 |\zeta\rangle,$$

where $c_0 = \frac{1}{\sqrt{D}}$, $d_0 = \frac{\sqrt{D-1}}{\sqrt{D}}$. Then, Grover's algorithm updates c_0 and d_0 in an iterative manner. In the m th iteration (rotation) of Grover's algorithm, the following two operations are applied to the current superposition $|\psi_{m-1}\rangle = c_{m-1}|b^*\rangle + d_{m-1}|\zeta\rangle$,

1. Flip operation F on $|\psi_{m-1}\rangle$, where $F|b^*\rangle = -|b^*\rangle$ and $F|b_j\rangle = |b_j\rangle$ for $|b_j\rangle \neq |b^*\rangle$. We can write it as $F|b_j\rangle = (1 - 2S^*(|b_j\rangle))|b_j\rangle$.

2. Diffusion operation \mathbf{U}_D on $F|\psi_{m-1}\rangle$, where

$$\mathbf{U}_D = 2|\psi_0\rangle\langle\psi_0| - \mathbf{I}_D, \quad (\text{A.8})$$

and \mathbf{I}_D is a $D \times D$ identity matrix.

From now on, the two operations together are referred to as Grover's operation $\mathbf{G} = \mathbf{U}_D F$.

Let θ be the angle that satisfies $\sin \theta = \sqrt{1/D}$. After the m th iteration, the current superposition $|\psi_{m-1}\rangle$ becomes

$$|\psi_m\rangle = \mathbf{G}|\psi_{m-1}\rangle = c_m|b^*\rangle + d_m|\zeta\rangle,$$

where the coefficients c_m and d_m satisfy

$$c_m = \sin((2m+1)\theta), \text{ and } d_m = \cos((2m+1)\theta).$$

When m satisfy $(2m+1)\theta < \pi/2$, the amplitude of oracle state $|b^*\rangle$ is gradually amplified as m increases.

Remark. Grover's algorithm can be easily extended to a search problem with multiple oracle states. For example, if the number of oracle states is D_{sol} , the number of iterations is approximately $\sqrt{D/D_{sol}}\pi/4$. However, if the number of oracle states is not given, one critical issue in applying Grover's algorithm is to determine the number of iterations.

A.3 Review of Quantum counting algorithm

Quantum Fourier transform. Suppose we have a basis state $|k\rangle$ of a T -qubit quantum system. The quantum Fourier transform operation QFT is defined as

$$QFT(|k\rangle) = \frac{1}{\sqrt{2^T}} \sum_{l=0}^{2^T-1} e^{\frac{2lk\pi i}{2^T}} |l\rangle. \quad (\text{A.9})$$

Analogously, the inverse quantum Fourier transform operation QFT^\dagger is defined as

$$QFT^\dagger(|k\rangle) = \frac{1}{\sqrt{2^T}} \sum_{l=0}^{2^T-1} e^{-\frac{2lk\pi i}{2^T}} |l\rangle. \quad (\text{A.10})$$

Quantum counting. In order to present the quantum counting algorithm, let us begin with a simple derivation to reveal the relationship between θ_w and eigenvalues of Grover's operation \mathbf{G}_w . For Grover's operation \mathbf{G}_w in [16], we perform the eigenvalue decomposition of the center matrix on the right-hand side. Grover's operation thus becomes

$$\mathbf{G}_w = \Gamma_w \begin{pmatrix} i/\sqrt{2} & -i/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \begin{pmatrix} e^{2\theta_w i} & 0 \\ 0 & e^{-2\theta_w i} \end{pmatrix} \begin{pmatrix} -i/\sqrt{2} & 1/\sqrt{2} \\ i/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \Gamma_w^\top + \tilde{\Gamma}_w \tilde{\Gamma}_w^\top, \quad (\text{A.11})$$

where i here is the imaginary unit of the complex number. We can see that the eigenvalues of \mathbf{G}_w are $e^{2\theta_w i}$ and $e^{-2\theta_w i}$. Note that the eigenvalues are complex numbers with the phases $\pm 2\theta_w$. Thus, estimating θ_w is equivalent to estimating Grover's operation \mathbf{G}_w 's eigenvalues.

By Eq. (A.11), the eigenvectors (aka eigenstates) of \mathbf{G}_w are $|\gamma_{w,1}\rangle = -\frac{i}{\sqrt{2}}|\psi_w\rangle + \frac{1}{\sqrt{2}}|\zeta_w\rangle$ and $|\gamma_{w,2}\rangle = \frac{i}{\sqrt{2}}|\psi_w\rangle + \frac{1}{\sqrt{2}}|\zeta_w\rangle$, and the corresponding eigenvalues are $e^{2\theta_w i}$ and $e^{-2\theta_w i}$.

That is

$$\begin{cases} \mathbf{G}_w(|\gamma_{w,1}\rangle) = e^{2\theta_w i} |\gamma_{w,1}\rangle, \\ \mathbf{G}_w(|\gamma_{w,2}\rangle) = e^{-2\theta_w i} |\gamma_{w,2}\rangle. \end{cases} \quad (\text{A.12})$$

The quantum counting algorithm uses two quantum registers. One register consists of T qubits initialized in the state $|0\rangle$. We refer to it as *rotation register*, and the initial state of this register is denoted by $|0\rangle^{\otimes T} = \underbrace{|0\rangle \otimes \cdots \otimes |0\rangle}_T$.

The BGS register is initialized at the superposition $|\psi_0\rangle = \frac{1}{\sqrt{D}} \sum_{j=0}^{D-1} |b_j\rangle \equiv \frac{1}{\sqrt{2}} e^{i\theta_w} |\gamma_{w,1}\rangle + \frac{1}{\sqrt{2}} e^{-i\theta_w} |\gamma_{w,2}\rangle$, which is a linear combination of two eigenvectors/eigenstates of Grover's operation. The details of each step in the quantum counting algorithm are summarized below.

Step 1. Prepare both the rotation register and the BGS register in the initial state of the uniform superposition.

We partially apply the Hadamard gate in [6] to all the T qubits in the rotation register.

Hence, the state of the joint system of rotation register and BGS register becomes

$$\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)^{\otimes T} \otimes |\psi_0\rangle.$$

Step 2. For $k = 1, \dots, T$, apply 2^{k-1} controlled Grover operations to the k th joint system consisting of the k th qubit of the rotation register and the whole BGS register.

We first apply the controlled- $\mathbf{G}_w^{2^0}$ operation on the first qubit in the rotation register and all qubits of the BGS register. In particular, the BGS register is treated as the target register, and the qubit of the rotation register is treated as the control qubit. Then, the state of the joint system becomes

$$\begin{aligned} & \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}e^{i2\theta_w}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)^{\otimes T-1} \otimes \frac{1}{\sqrt{2}}e^{i\theta_w}|\gamma_{w,1}\rangle + \\ & \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}e^{-i2\theta_w}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)^{\otimes T-1} \otimes \frac{1}{\sqrt{2}}e^{-i\theta_w}|\gamma_{w,2}\rangle. \end{aligned}$$

We continually apply the controlled- $\mathbf{G}_w^{2^{t-1}}$ operation on $t = 2, \dots, T$ th qubit in the rotation register in addition to the BGS register. The state of the joint system becomes

$$\frac{e^{i\theta_w}}{2^{(T+1)/2}} \bigotimes_{t=1}^T \left(|0\rangle + e^{i2^t\theta_w}|1\rangle\right) \otimes |\gamma_{w,1}\rangle + \frac{e^{-i\theta_w}}{2^{(T+1)/2}} \bigotimes_{t=1}^T \left(|0\rangle + e^{-i2^t\theta_w}|1\rangle\right) \otimes |\gamma_{w,2}\rangle. \quad (\text{A.13})$$

By elementary algebra, Eq. (A.13) can be written as

$$\frac{1}{2^{(T+1)/2}} \sum_{k=0}^{2^T-1} e^{i\theta_w(2k+1)} |k\rangle \otimes |\gamma_{w,1}\rangle + \frac{1}{2^{(T+1)/2}} \sum_{k=0}^{2^T-1} e^{-i\theta_w(2k+1)} |k\rangle \otimes |\gamma_{w,2}\rangle. \quad (\text{A.14})$$

Step 3. Apply the inverse quantum Fourier transform to each qubit in the rotation register.

After we apply inverse quantum Fourier transform QFT^\dagger on the rotation register, the state of the joint system becomes

$$|e_{\theta_w}\rangle \otimes \frac{1}{\sqrt{2}}|\gamma_{w,1}\rangle + |e_{-\theta_w}\rangle \otimes \frac{1}{\sqrt{2}}|\gamma_{w,2}\rangle, \quad (\text{A.15})$$

where $|e_{\theta_w}\rangle = \frac{1}{2^T} \sum_{k,l=0}^{2^T-1} e^{i(\theta_w(2k+1) - \frac{2\pi kl}{2^T})} |l\rangle$ and $|e_{-\theta_w}\rangle = \frac{1}{2^T} \sum_{k,l=0}^{2^T-1} e^{-i(\theta_w(2k+1) + \frac{2\pi kl}{2^T})} |l\rangle$.

Step 4. Measure the joint system of the rotation register and the BGS register. The measurements of the rotation register are used to estimate θ_w .

We now show how to use the measurement result from step 4 to estimate θ_w . By Eq. (A.15), we see that the state of the joint system is a linear combination of two states $|e_{\theta_w}\rangle \otimes |\gamma_{w,1}\rangle$ and $|e_{-\theta_w}\rangle \otimes |\gamma_{w,2}\rangle$ with equal weights. Measuring this joint system can be interpreted as a two-step procedure. In the first step, it randomly samples one state out of $|e_{\theta_w}\rangle \otimes |\gamma_{w,1}\rangle$ and $|e_{-\theta_w}\rangle \otimes |\gamma_{w,2}\rangle$ with the same probability. In the second step, the sampled state is measured. i.e., quantum collapse occurs. Therefore, there are two possible scenarios when we measure the rotation register. Scenario 1, the measurement of the rotation register comes from the state $|e_{\theta_w}\rangle$. Scenario 2, the measurement of the rotation register comes from the state $|e_{-\theta_w}\rangle$. Each scenario occurs with a probability of 0.5. Let $|l_w\rangle$ be the final measurement of the rotation register. We can show that if the measurement comes from the state $|e_{\theta_w}\rangle$, $\frac{l_w}{2^T}$ is very close to $\frac{\theta_w}{\pi}$ with a high probability. If the measurement comes from the state $|e_{-\theta_w}\rangle$, $\frac{l_w}{2^T}$ is very close to $1 - \frac{\theta_w}{\pi}$ with a high probability

We now need to determine whether the measurement is the result of $|e_{\theta_w}\rangle$ or that of $|e_{-\theta_w}\rangle$. If $\frac{l_w}{2^T} \in [0, 0.5)$, $\theta_w \in (0, \frac{\pi}{2})$. Consequently, the result is a measurement of $|e_{\theta_w}\rangle$. Otherwise, the result is a measurement of $|e_{-\theta_w}\rangle$. We thus estimate θ_w as follows,

$$\hat{\theta}_w = \begin{cases} \frac{l_w}{2^T}\pi, & \text{if } 0 \leq \frac{l_w}{2^T} < 0.5 \\ (1 - \frac{l_w}{2^T})\pi, & \text{otherwise.} \end{cases} \quad (\text{A.16})$$

Proof of Lemma 2.1.

This proof is largely adapted and extended from the analysis presented in section 5.2 of Nielsen and Chuang (2010).

We derive the error bound of $|\hat{\theta}_w - \theta_w|$ by considering two scenarios: scenario 1: the

measurement of the rotation register comes from the state $|e_{\theta_w}\rangle$, and scenario 2: the measurement of the rotation register comes from the state $|e_{-\theta_w}\rangle$. We present the analysis for scenario 1 only since the analysis for scenario 2 can be analogously obtained.

In scenario 1, we are measuring the state $|e_{\theta_w}\rangle$. Let $\rho = \lfloor 2^T \frac{\theta_w}{\pi} \rfloor$ and $\rho_m = (\rho + m)(\text{mod } 2^T)$, where “mod” is the modulo operation. Let λ_m be the amplitude of $|\rho_m\rangle$, where $m = -2^{T-1} + 1, \dots, 2^{T-1}$. Thus, we have

$$\lambda_m \equiv \frac{1}{2^T} \sum_{k=0}^{2^T-1} e^{i(\theta_w(2k+1) - \frac{2\pi k(\rho+m)}{2^T})} = \frac{e^{i\theta_w}}{2^T} \left(\frac{1 - W_m^{2^T}}{1 - W_m} \right), \quad (\text{A.17})$$

where $W_m = e^{2\pi i(\frac{\theta_w}{\pi} - \frac{\rho+m}{2^T})}$. By the fact $\frac{2|\theta|}{\pi} \leq |1 - e^{i\theta}| \leq 2$ when $-\pi \leq \theta \leq \pi$, we have

$$|\lambda_m| \leq \frac{1}{2|(2^T \frac{\theta_w}{\pi} - \rho) - m|}. \quad (\text{A.18})$$

Suppose the final measurement of the rotation register is $\frac{l_w}{2^T}$. For a positive integer $\Delta \geq 2$, we have

$$\begin{aligned} \mathbb{P}(|\frac{l_w}{2^T} - \frac{\theta_w}{\pi}| \geq \frac{\Delta}{2^T}) &\leq \sum_{m=-2^{T-1}+1}^{-\Delta} \mathbb{P}(w = \rho_m) + \sum_{m=\Delta}^{2^{T-1}} \mathbb{P}(w = \rho_m) \\ &= \sum_{m=-2^{T-1}+1}^{-\Delta} |\lambda_m|^2 + \sum_{m=\Delta}^{2^{T-1}} |\lambda_m|^2 \\ &\leq \sum_{m=-2^{T-1}+1}^{-\Delta} \frac{1}{4m^2} + \sum_{m=\Delta}^{2^{T-1}} \frac{1}{4(m-1)^2} \\ &= -\frac{1}{4(2^{T-1}-1)^2} + \frac{1}{2} \sum_{m=\Delta}^{2^{T-1}} \frac{1}{m^2} \\ &< \frac{1}{2} \int_{\Delta-1}^{2^{T-1}} \frac{1}{x^2} dx \\ &< \frac{1}{2(\Delta-1)}. \end{aligned}$$

For any $\varepsilon \in (0, 0.25)$, we have

$$\begin{aligned} \mathbb{P}(|\frac{l_w}{2^T} - \frac{\theta_w}{\pi}| > \varepsilon + \frac{1}{2^T}) &\leq \mathbb{P}(|\frac{l_w}{2^T} - \frac{\theta_w}{\pi}| \geq \frac{\lfloor 2^T \varepsilon \rfloor + 2}{2^T}) \\ &< \frac{1}{2[2^T \varepsilon + 1]} \\ &< \frac{1}{2^{T+1} \varepsilon}, \end{aligned}$$

where $\lfloor \cdot \rfloor$ is the floor function. If $\left| \frac{l_w}{2^T} - \frac{\theta_w}{\pi} \right| \leq \frac{\Delta}{2^T}$, we have $\frac{l_w}{2^T} \in (0, \frac{\theta_w}{\pi} + \frac{1}{4})$. Since $\theta_w < \frac{\pi}{4}$, we have $\frac{l_w}{2^T} \in (0, 0.5)$. Therefore, by Eq. (A.16), we have

$$\mathbb{P}\left(\frac{1}{\pi}|\hat{\theta}_w - \theta_w| > \varepsilon + \frac{1}{2^T}\right) = \mathbb{P}\left(\left|\frac{l_w}{2^T} - \frac{\theta_w}{\pi}\right| > \varepsilon + \frac{1}{2^T}\right) < \frac{1}{2^{T+1}\varepsilon}.$$

We can analogously derive the above inequality for scenario 2.

The proof is thus completed. \square

For the convenience of the proofs of the main theorems in the paper, we can get the following corollary straightforward from Lemma 2.1.

Corollary A.1. *Assume all the conditions in Lemma 2.1 are satisfied. For any $\varepsilon_1 \in \left(\frac{\pi}{2^{T-1}}, \frac{\pi}{4}\right)$, we have*

$$\mathbb{P}\left(\left|\hat{\theta}_w - \theta_w\right| > \varepsilon_1\right) < \frac{1}{2^T\varepsilon_1}. \quad (\text{A.19})$$

Remark. Further, for any $\varepsilon_1 \in \left(0, \frac{\pi}{4}\right)$, there exists t_0 such that when $T > t_0$, Eq. (A.19) still holds. Specifically, we can let $t_0 = \log_{\varepsilon_1} \frac{2\pi}{\varepsilon_1}$.

B Calculating BIC via quantum computing

We present the quantum computing method for calculating BIC values in linear regression models. As the first step, we load the sample information into a quantum computer, which is called data encoding. Here, we use the amplitude encoding method to encode the sample, including the covariate $\mathbf{X}^{\mathcal{A}_j}$ and the response, as follows.

$$|\mathbf{X}^{\mathcal{A}_j}\rangle = \sum_{i=1}^n \sum_{k=1}^p x_{i,k}^{\mathcal{A}_j} |c_i\rangle |a_k\rangle, \quad (\text{B.20})$$

$$|\mathbf{y}\rangle = \sum_{i=1}^n \tilde{y}_i |c_i\rangle, \quad (\text{B.21})$$

where $\tilde{x}_{i,k}^{\mathcal{A}_j} = \frac{x_{i,k} I\{k \in \mathcal{A}_j\}}{\|\mathbf{X}^{\mathcal{A}_j}\|_F}$, $\tilde{y}_i = \frac{y_i}{\|\mathbf{y}\|_2}$, $\{|c_i\rangle\}_{i=1}^n$ are n basis states encoding the indices of the cells, $\{|a_i\rangle\}_{k=1}^p$ are p basis states encoding the indices of the ADTs. To construct the data encoding operation $\mathbf{S}_{\mathbf{X}}$ and $\mathbf{E}_{\mathbf{y}}$, see [Weigold et al. \(2020\)](#) for a comprehensive review of data encoding algorithms.

We assume $\mathbf{X}^{\mathcal{A}_j}$ has the singular value decomposition $\mathbf{X}^{\mathcal{A}_j} = \mathbf{U}^{\mathcal{A}_j} \Lambda^{\mathcal{A}_j} \mathbf{V}^{\mathcal{A}_j}$, then we have the BIC value under the linear regression model,

$$\begin{aligned} \text{BIC}(\mathcal{A}_j) &= |\mathcal{A}_j| \log n + n \log n [\text{tr}(\mathbf{y}\mathbf{y}^\top) - \text{tr}(\mathbf{X}^{\mathcal{A}_j} (\mathbf{X}^{\mathcal{A}_j\top} \mathbf{X}^{\mathcal{A}_j})^{-1} \mathbf{X}^{\mathcal{A}_j\top} \mathbf{y}\mathbf{y}^\top)] \\ &= |\mathcal{A}_j| \log n + n \log n [\text{tr}(\mathbf{y}\mathbf{y}^\top) - \text{tr}(\mathbf{X}^{\mathcal{A}_j} (\mathbf{X}^{\mathcal{A}_j\top} \mathbf{X}^{\mathcal{A}_j})^{-1} \mathbf{X}^{\mathcal{A}_j\top} \mathbf{y}\mathbf{y}^\top)] \\ &= |\mathcal{A}_j| \log n + n \log n [\text{tr}(\mathbf{y}\mathbf{y}^\top) - \text{tr}(\mathbf{U}^{\mathcal{A}_j} \mathbf{U}^{\mathcal{A}_j\top} \mathbf{y}\mathbf{y}^\top)]. \end{aligned}$$

Notice that to compare BIC values for different subsets, it is sufficient to calculate $\text{tr}(\mathbf{U}^{\mathcal{A}_j} \mathbf{U}^{\mathcal{A}_j\top} \mathbf{y}\mathbf{y}^\top)$.

In quantum computing, $\text{tr}(\mathbf{U}^{\mathcal{A}_j} \mathbf{U}^{\mathcal{A}_j\top} \mathbf{y}\mathbf{y}^\top)$ is in proportion to the inner products of two quantum states whose density matrices are $\frac{1}{\|\mathbf{U}\|_F^2} \mathbf{U}^{\mathcal{A}_j} \mathbf{U}^{\mathcal{A}_j\top}$ and $\frac{1}{\|\mathbf{y}\|_2^2} \mathbf{y}\mathbf{y}^\top$, respectively.

Consequently, we consider

$$|\mathbf{X}^{\mathcal{A}_j}\rangle = \sum_{r=1}^R \sigma_r^{\mathcal{A}_j} |u_r^{\mathcal{A}_j}\rangle |v_r^{\mathcal{A}_j}\rangle,$$

where $\sigma_r^{\mathcal{A}_j}$ is the normalized r th singular value, $|u_r^{\mathcal{A}_j}\rangle = \frac{1}{\|u_{\cdot,r}^{\mathcal{A}_j}\|_2} \sum_{i=1}^n u_{i,r}^{\mathcal{A}_j} |c_i\rangle$, $|v_r^{\mathcal{A}_j}\rangle = \frac{1}{\|v_{\cdot,r}^{\mathcal{A}_j}\|_2} \sum_{k=1}^p v_{k,r}^{\mathcal{A}_j} |a_k\rangle$, $u_{i,r}^{\mathcal{A}_j}$ is the (i,r) th entry of the matrix $\mathbf{U}^{\mathcal{A}_j}$, and $v_{k,r}^{\mathcal{A}_j}$ is the (k,r) th entry of the matrix $\mathbf{V}^{\mathcal{A}_j}$. By inverting the singular values via the HHL algorithm (Harrow et al., 2009), we convert the state $|\mathbf{X}^{\mathcal{A}_j}\rangle$ to a desired state

$$\sum_{r=1}^R |u_r^{\mathcal{A}_j}\rangle |v_r^{\mathcal{A}_j}\rangle. \quad (\text{B.22})$$

We notice that the partial trace over the register encoding the ADT indices yields the mixed state

$$\sum_{r_1, r_2=1}^R |u_{r_1}^{\mathcal{A}_j}\rangle \langle u_{r_2}^{\mathcal{A}_j}|, \quad (\text{B.23})$$

whose density matrix is $\frac{1}{\|\mathbf{U}\|_F^2} \mathbf{U}^{\mathcal{A}_j} \mathbf{U}^{\mathcal{A}_j \top}$. Furthermore, the density matrix of the state $|\mathbf{y}\rangle$ is $\frac{1}{\|\mathbf{y}\|_2^2} \mathbf{y} \mathbf{y}^\top$. Hence, calculating the inner product of these two states can be conducted via Swap test (Buhrman et al., 2001).

C Proofs of main results

In this section, we present the proofs of our main results in the paper. The following propositions are needed for proving Theorem 3.1 and Theorem 3.3. For Grover's rotations in BGS, given a current benchmark state $|b_w\rangle$, the corresponding θ_w and \mathcal{B}_w are fixed but unknown. The optimal number of rotations is

$$\tau_w = \frac{\pi}{4\theta_w} - \frac{1}{2}. \quad (\text{C.24})$$

In practice, we estimate τ_w through a two-step procedure. Step 1. Since θ_w is unknown, we get its estimate $\hat{\theta}_w$ via the quantum counting algorithm. Step 2. Since the number of rotations is an integer, our estimate is

$$\hat{\tau}_w = \left[\frac{\pi}{4\hat{\theta}_w} - \frac{1}{2} \right]. \quad (\text{C.25})$$

where $\lceil \cdot \rceil$ means rounding to the nearest integer. The following proposition states the error bound of $\hat{\tau}_w$.

Proposition C.1. *Assume all the conditions in Lemma 2.1 are satisfied. For $\varepsilon_2 \in \left(0, \frac{1}{4\theta_w}\right)$,*

there exists a constant t_0 such that when $T > t_0$, we have

$$\mathbb{P} \left(|\hat{\tau}_w - \tau_w| > \frac{1}{2} + \varepsilon_2 \right) < \frac{2\varepsilon_2}{2^T(2\theta_w\varepsilon_2 + 1 - \sqrt{1 - 4\varepsilon_2\theta_w})}. \quad (\text{C.26})$$

Proof of Proposition C.1. Let $\tilde{\tau}_w = \frac{\pi}{4\hat{\theta}_w} - \frac{1}{2}$. We have

$$|\hat{\tau}_w - \tau_w| \leq |\hat{\tau}_w - \tilde{\tau}_w| + |\tilde{\tau}_w - \tau_w|. \quad (\text{C.27})$$

For the first term on the right-hand side of Eq.(C.27), we immediately have

$$|\hat{\tau}_w - \tilde{\tau}_w| = \left| \left[\frac{\pi}{4\hat{\theta}_w} - \frac{1}{2} \right] - \left(\frac{\pi}{4\hat{\theta}_w} - \frac{1}{2} \right) \right| \leq \frac{1}{2}. \quad (\text{C.28})$$

For the second term on the right-hand side of Eq.(C.27), we have

$$\begin{aligned} |\tilde{\tau}_w - \tau_w| &= \frac{\pi}{4} \left| \frac{1}{\hat{\theta}_w} - \frac{1}{\theta_w} \right| \\ &\leq \frac{\pi}{4} \left(\min \left\{ \hat{\theta}_w, \theta_w \right\} \right)^{-2} \left| \hat{\theta}_w - \theta_w \right| \\ &\leq \frac{\pi}{4} \left(\theta_w - \left| \hat{\theta}_w - \theta_w \right| \right)^{-2} \left| \hat{\theta}_w - \theta_w \right|. \end{aligned} \quad (\text{C.29})$$

Notice that in Eq.(C.29), the term $\frac{\pi}{4}(\theta_w - x)^{-2}x$ is monotone increasing in x when $x \in (0, \theta_w)$. Therefore, for any $\varepsilon_1 \in (0, \theta_w)$, there exists a constant t_0 such that when $T > t_0$, by Corollary A.1, we have

$$\begin{aligned} &\mathbb{P} \left(|\tilde{\tau}_w - \tau_w| > \frac{\pi}{4}(\theta_w - \varepsilon_1)^{-2}\varepsilon_1 \right) \\ &\leq \mathbb{P} \left(\frac{\pi}{4} \left(\theta_w - \left| \hat{\theta}_w - \theta_w \right| \right)^{-2} \left| \hat{\theta}_w - \theta_w \right| > \frac{\pi}{4}(\theta_w - \varepsilon_1)^{-2}\varepsilon_1 \right) \\ &= \mathbb{P} \left(\left| \hat{\theta}_w - \theta_w \right| > \varepsilon_1 \right) \\ &< \frac{1}{2^T\varepsilon_1}. \end{aligned} \quad (\text{C.30})$$

We let

$$\varepsilon_2 = \frac{\pi}{4} (\theta_w - \varepsilon_1)^{-2} \varepsilon_1. \quad (\text{C.31})$$

For $\varepsilon_2 \in \left(0, \frac{1}{4\theta_w}\right)$, Eq.(C.31) has two solutions, and the solution in $(0, \theta_w)$ is

$$\varepsilon_1 = \frac{2\theta_w\varepsilon_2 + 1 - \sqrt{(2\theta_w\varepsilon_2 + 1)^2 - 4\varepsilon_2^2\theta_w^2}}{2\varepsilon_2}. \quad (\text{C.32})$$

Plugging Eq.(C.32) into Eq.(C.30), for $\varepsilon_2 \in \left(0, \frac{1}{4\theta_w}\right)$, we have

$$\begin{aligned} \mathbb{P}(|\tilde{\tau}_w - \tau_w| > \varepsilon_2) &< \frac{2\varepsilon_2}{2^T(2\theta_w\varepsilon_2 + 1 - \sqrt{(2\theta_w\varepsilon_2 + 1)^2 - 4\varepsilon_2^2\theta_w^2})} \\ &= \frac{2\varepsilon_2}{2^T(2\theta_w\varepsilon_2 + 1 - \sqrt{1 - 4\varepsilon_2\theta_w})}. \end{aligned} \quad (\text{C.33})$$

Combining Eq.(C.28) and Eq.(C.33), we complete the proof by

$$\begin{aligned} \mathbb{P}\left(|\hat{\tau}_w - \tau_w| > \frac{1}{2} + \varepsilon_2\right) &\leq \mathbb{P}\left(|\hat{\tau}_w - \tilde{\tau}_w| + |\tilde{\tau}_w - \tau_w| > \frac{1}{2} + \varepsilon_2\right) \\ &\leq \mathbb{P}(|\tilde{\tau}_w - \tau_w| > \varepsilon_2) \\ &< \frac{2\varepsilon_2}{2^T(2\theta_w\varepsilon_2 + 1 - \sqrt{1 - 4\varepsilon_2\theta_w})}. \end{aligned} \quad (\text{C.34})$$

□

Corollary C.2. Under the conditions of Proposition C.1, for any $\varepsilon_2 \in \left(0, \frac{1}{4\theta_w}\right)$, there exists a constant t_0 such that when $T > t_0$, we have

$$\mathbb{P}\left(|\hat{\tau}_w - \tau_w| > \frac{1}{2} + \varepsilon_2\right) < \frac{1}{2^T\theta_w}. \quad (\text{C.35})$$

Proof of Corollary C.2. By Proposition C.1, since $1 - \sqrt{1 - 4\varepsilon_2\theta_w} > 0$,

$$\begin{aligned} \mathbb{P}\left(|\hat{\tau}_w - \tau_w| > \frac{1}{2} + \varepsilon_2\right) &\leq \frac{2\varepsilon_2}{2^T(2\theta_w\varepsilon_2 + 1 - \sqrt{1 - 4\varepsilon_2\theta_w})} \\ &< \frac{2\varepsilon_2}{2^{T+1}\theta_w\varepsilon_2} \\ &= \frac{1}{2^T\theta_w}. \end{aligned}$$

□

Next, we present the error analysis of the proposed BGS method. Algorithm 1 can be decomposed into the outer loop and the inner loop. The inner loop consists of steps from (b1) to (b2) with the stopping rule $S_w^g(|b_{w^{new}}\rangle) = 1$, which conducts multiple sets of rotations and measurements until the oracle state $|b_{w^{new}}\rangle$ is found. The outer loop consists of steps from (b1) to (d) with the stopping rule $\sin(\hat{\theta}_w) < \delta$, which iterates until the final benchmark state is found. The following proposition is for the error of the inner loop in BGS.

Proposition C.3 (Error of the inner loop in BGS). *Assume all the conditions in Lemma 2.1 are satisfied. Given a current benchmark state $|b_w\rangle$ with the corresponding θ_w , and \mathcal{B}_w , there exists a constant t_0 such that when $T > t_0$, the output state $|b_w^{new}\rangle$ of the inner loop in BGS has the following error bound*

$$\mathbb{P}(|b_w^{new}\rangle \notin \mathcal{B}_w) < \theta_w + \frac{1}{2^T \theta_w}. \quad (\text{C.36})$$

Proof of Proposition C.3. Combining [18] and the fact that $\sin^2((2\tau_w + 1)\theta_w) = 1$, we have

$$\begin{aligned} \mathbb{P}(|b_w^{new}\rangle \in \mathcal{B}_w) &= \sin^2((2\hat{\tau}_w + 1)\theta_w) \\ &= 1 - (\sin^2((2\tau_w + 1)\theta_w) - \sin^2((2\hat{\tau}_w + 1)\theta_w)) \\ &> 1 - 2\theta_w |\hat{\tau}_w - \tau_w|. \end{aligned} \quad (\text{C.37})$$

For any $\varepsilon_2 \in \left(0, \frac{1}{4\theta_w}\right)$, by Corollary C.2, there exists a constant t_0 such that when $T > t_0$,

we have

$$\begin{aligned}
& \mathbb{P}(|b_{w^{new}}\rangle \notin \mathcal{B}_w) \\
&= 1 - \mathbb{P}(|b_{w^{new}}\rangle \in \mathcal{B}_w) \\
&\leq 1 - \mathbb{P}\left(|b_{w^{new}}\rangle \in \mathcal{B}_w, |\hat{\tau}_w - \tau_w| \leq \frac{1}{2} + \varepsilon_2\right) \\
&= 1 - \mathbb{P}\left(|b_{w^{new}}\rangle \in \mathcal{B}_w \mid |\hat{\tau}_w - \tau_w| \leq \frac{1}{2} + \varepsilon_2\right) \mathbb{P}\left(|\hat{\tau}_w - \tau_w| \leq \frac{1}{2} + \varepsilon_2\right) \\
&< 1 - \left(1 - 2\theta_w\left(\frac{1}{2} + \varepsilon_2\right)\right) \left(1 - \frac{1}{2^T\theta_w}\right). \tag{C.38}
\end{aligned}$$

Taking infimum of Eq.(C.38) with respect to ε_2 , we get

$$\begin{aligned}
\mathbb{P}(|b_{w^{new}}\rangle \notin \mathcal{B}_w) &\leq 1 - (1 - \theta_w) \left(1 - \frac{1}{2^T\theta_w}\right) \\
&= \theta_w + \frac{1}{2^T\theta_w} - \frac{1}{2^T} \\
&< \theta_w + \frac{1}{2^T\theta_w}. \tag{C.39}
\end{aligned}$$

□

Proposition C.4 (Error of the outer loop in BGS). *Assume all the conditions in Theorem 3.1 are satisfied. For the outer loop iteration, there exists a constant t_0 such that when $T > t_0$, we have the following uniform error bound, i.e., it does not depend on $|b_w\rangle$,*

$$\mathbb{P}\left(\sin(\hat{\theta}) < \delta \mid \sin(\theta) \neq 0\right) < \frac{1}{2^T(1/\sqrt{D} - \delta)}. \tag{C.40}$$

Proof of Proposition C.4. By Corollary A.1, there exists a constant t_0 such that when $T > t_0$,

$$\begin{aligned}
\mathbb{P}\left(\sin(\hat{\theta}) < \delta \mid \sin(\theta) \neq 0\right) &\leq \mathbb{P}\left(\sin(\hat{\theta}) < \delta \mid \sin(\theta) = 1/\sqrt{D}\right) \\
&\leq \mathbb{P}\left(|\sin(\hat{\theta}) - \sin(\theta)| > 1/\sqrt{D} - \delta\right) \\
&\leq \mathbb{P}\left(|\hat{\theta} - \theta| > 1/\sqrt{D} - \delta\right) \\
&< \frac{1}{2^T(1/\sqrt{D} - \delta)}.
\end{aligned}$$

□

Proof of Theorem 3.1. To distinguish θ_w 's in different iterations, we let $b_w^{(k)}$, $\theta_w^{(k)}$ and $\hat{\theta}_w^{(k)}$ denote the b_w , θ_w and $\hat{\theta}_w$ in the k -th iteration, respectively. Note that $\sin(\theta_w^{(k)})$ is a stochastic process such that $\sin(\theta_w^{(1)})$ follows a discrete uniform distribution

$$\text{Unif} \left\{ 1/\sqrt{D}, \sqrt{2/D}, \dots, \sqrt{D/D} \right\}, \quad (\text{C.41})$$

and for $k \geq 2$, given $\sin(\theta_w^{(k-1)}) = \sqrt{i/D}$, $\sin(\theta_w^{(k)})$ follows

$$\text{Unif} \left\{ 1/\sqrt{D}, \sqrt{2/D}, \dots, \sqrt{(i-1)/D} \right\}. \quad (\text{C.42})$$

Note that $\sin(\hat{\theta}_w^{(k)})$ is also a stochastic process, which is a measurement of $\sin(\theta_w^{(k)})$, with some error. The error of $\sin(\hat{\theta}_w^{(k)})$ is quantified in Lemma 2.1.

Note that in practice, the two processes $\sin(\theta_w^{(k)})$ and $\sin(\hat{\theta}_w^{(k)})$ end when the algorithm is terminated (at some time $k \leq D$). However, for the convenience of theoretical discussion, we assume that they do not end until $k = D$. Specifically, $\sin(\theta_w^{(k)})$ follows the distribution according to Eq.(C.42), and once $\sin(\theta_w^{(k)}) = 0$, we let the following $\sin(\theta_w^{(k+1)}) = \dots = \sin(\theta_w^{(D)}) = 0$. For $\sin(\hat{\theta}_w^{(k)})$, we let it be the measurement of $\sin(\theta_w^{(k)})$, respectively for $k = 1, \dots, D$.

Suppose the algorithm is terminated at the K -th iteration. The stopping time of BGS is explicitly defined by

$$K = \min \left\{ k \geq 1 : \sin(\hat{\theta}_w^{(k)}) < \delta \right\}, \quad (\text{C.43})$$

which implies $\sin(\hat{\theta}_w^{(k)}) > \delta$ for $i = 1, \dots, K-1$, but $\sin(\hat{\theta}_w^{(K)}) < \delta$. We also define a ‘true’ stopping time K^* that the underlying θ_w reached 0 for the first time, that is,

$$K^* = \min \left\{ k \geq 1 : \sin(\theta_w^{(k)}) = 0 \right\}. \quad (\text{C.44})$$

For the simplicity of the proof, although the algorithm is terminated at K , we theoretically let the processes of $\sin(\theta_w^{(k)})$ and $\sin(\hat{\theta}_w^{(k)})$ go on also for $i > K$.

We aim to bound the error probability $\mathbb{P}(|b_w\rangle \neq |b^*\rangle)$ for the final output state $|b_w\rangle$.

This error probability can also be rewritten as $\mathbb{P}(|b_w^{(K)}\rangle \neq |b^*\rangle)$. Notice that $\sin(\theta_w^{(K)}) = 0$ implies that there is no state with BIC smaller than $|b_w^{(K)}\rangle$, and hence $|b_w^{(K)}\rangle = |b^*\rangle$. Thus we have

$$\begin{aligned}\mathbb{P}(|b_w^{(K)}\rangle \neq |b^*\rangle) &= 1 - \mathbb{P}(|b_w^{(K)}\rangle = |b^*\rangle) \\ &= 1 - \mathbb{P}(\sin(\theta_w^{(K)}) = 0) \\ &= 1 - \mathbb{P}(K \geq K^*) \\ &\leq 1 - \mathbb{P}(K > K^*).\end{aligned}\tag{C.45}$$

By Proposition C.4, there exists a constant t_0 such that when $T > t_0$, we have

$$\begin{aligned}\mathbb{P}(K > K^*) &= \sum_{k=1}^D \mathbb{P}(K > K^*, K^* = k) \\ &= \sum_{k=1}^D \mathbb{P}\left(K > K^* \mid K^* = k\right) \mathbb{P}(K^* = k) \\ &= \sum_{k=1}^D \mathbb{P}\left(\sin(\hat{\theta}_w^{(1)}) \geq \delta, \dots, \sin(\hat{\theta}_w^{(k)}) \geq \delta \mid K^* = k\right) \mathbb{P}(K^* = k) \\ &\geq \sum_{k=1}^D \left(1 - \frac{1}{2^T(1/\sqrt{D} - \delta)}\right)^k \mathbb{P}(K^* = k) \\ &= \mathbb{E}\left[\left(1 - \frac{1}{2^T(1/\sqrt{D} - \delta)}\right)^{K^*}\right].\end{aligned}\tag{C.46}$$

By Jensen's inequality,

$$\begin{aligned}\mathbb{E}\left[\left(1 - \frac{1}{2^T(1/\sqrt{D} - \delta)}\right)^{K^*}\right] &\geq \left(1 - \frac{1}{2^T(1/\sqrt{D} - \delta)}\right)^{\mathbb{E}(K^*)} \\ &= \left(1 - \frac{1}{2^T(1/\sqrt{D} - \delta)}\right)^{\log D}.\end{aligned}\tag{C.47}$$

Thus, we have

$$\mathbb{P}(|b_w^{(K)}\rangle \neq |b^*\rangle) \leq 1 - \left(1 - \frac{1}{2^T(1/\sqrt{D} - \delta)}\right)^{\log D}.\tag{C.48}$$

Note that the function $(1 - \frac{1}{x})^x$ monotonically increases with respect to x for $x > 0$.

Moreover, $\lim_{x \rightarrow \infty} (1 - \frac{1}{x})^x = e^{-1}$. Since $T = \Omega\left(\log \frac{\log D}{(1/\sqrt{D}-\delta)}\right)$, we have $2^T(1/\sqrt{D} - \delta) \rightarrow \infty$,

and further,

$$\frac{\left(1 - \frac{1}{2^T(1/\sqrt{D}-\delta)}\right)^{\log D}}{e^{-\frac{\log D}{2^T(1/\sqrt{D}-\delta)}}} = \left[\frac{\left(1 - \frac{1}{2^T(1/\sqrt{D}-\delta)}\right)^{2^T(1/\sqrt{D}-\delta)}}{e^{-1}} \right]^{\frac{\log D}{2^T(1/\sqrt{D}-\delta)}} \rightarrow 1. \quad (\text{C.49})$$

Thus, we get

$$\left(1 - \frac{1}{2^T(1/\sqrt{D}-\delta)}\right)^{\log D} \sim e^{-\frac{\log D}{2^T(1/\sqrt{D}-\delta)}}. \quad (\text{C.50})$$

By Eq.(C.45) to Eq.(C.50), since $1 - e^{-x} \leq x$ for any $x \in \mathbb{R}$, we have

$$\mathbb{P}\left(|b_w^{(K)}\rangle \neq |b^*\rangle\right) \lesssim 1 - e^{-\frac{\log D}{2^T(1/\sqrt{D}-\delta)}} \leq \frac{\log D}{2^T(1/\sqrt{D}-\delta)}. \quad (\text{C.51})$$

In conclusion, we have

$$\mathbb{P}\left(|b_w^{(K)}\rangle \neq |b^*\rangle\right) \lesssim \frac{\log D}{2^T(1/\sqrt{D}-\delta)}. \quad (\text{C.52})$$

□

Proof of Corollary 3.2. Since $\sqrt{D}\delta \leq 1-c$ for some positive constant $c < 1$, $1/\sqrt{D}-\delta \geq c/\sqrt{D}$, and thus,

$$\frac{\log D}{2^T(1/\sqrt{D}-\delta)} \leq \frac{\sqrt{D}\log D}{c2^T}. \quad (\text{C.53})$$

Plugging Eq.(C.53) to Eq.(C.52), we have

$$\mathbb{P}\left(|b_w^{(K)}\rangle \neq |b^*\rangle\right) \lesssim \frac{\sqrt{D}\log D}{c2^T}. \quad (\text{C.54})$$

□

Proposition C.5. Assume all the conditions in Lemma 2.1 are satisfied. We also assume

T satisfies $\frac{\log D}{2^T(1/\sqrt{D}-\delta)} \rightarrow 0$ as $D \rightarrow \infty$. Suppose $\sqrt{D}\delta$ is lower bounded away from 1, i.e., $\sqrt{D}\delta \geq 1 + c$ for some positive constant c . The final output state $|b_w\rangle$ has the following error

$$\left| \mathbb{P}(|b_w\rangle \neq |b^\star\rangle) - \frac{1}{2} \right| = O\left(\frac{1}{2^T / (\sqrt{D} \log D) - 1}\right). \quad (\text{C.55})$$

Proof of Proposition C.5. We aim to bound

$$\mathbb{P}(|b_w\rangle \neq |b^\star\rangle) = \mathbb{P}(\sin(\theta_w^{(K)}) \neq 0) = 1 - \mathbb{P}(\sin(\theta_w^{(K)}) = 0).$$

We have the following decomposition,

$$\begin{aligned} & \mathbb{P}(\sin(\theta_w^{(K)}) = 0) \\ &= \sum_{k=1}^D \mathbb{P}\left(\sin(\theta_w^{(k)}) = 0 \mid K = k\right) \mathbb{P}(K = k) \\ &= \sum_{k=1}^D \mathbb{P}\left(\sin(\theta_w^{(k)}) = 0 \mid \sin(\hat{\theta}_w^{(k)}) < \delta\right) \mathbb{P}(K = k). \end{aligned} \quad (\text{C.56})$$

Further, we decompose

$$\begin{aligned} & \mathbb{P}\left(\sin(\theta_w^{(k)}) = 0 \mid \sin(\hat{\theta}_w^{(K)}) < \delta\right) \\ &= \sum_{i=1}^D \mathbb{P}\left(\sin(\theta_w^{(k)}) = 0, \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \mid \sin(\hat{\theta}_w^{(k)}) < \delta\right) \\ &= \sum_{i=1}^D \mathbb{P}\left(\sin(\theta_w^{(k)}) = 0 \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D}, \sin(\hat{\theta}_w^{(k)}) < \delta\right) \\ & \quad \mathbb{P}\left(\sin(\theta_w^{(k-1)}) = \sqrt{i/D} \mid \sin(\hat{\theta}_w^{(k)}) < \delta\right). \end{aligned} \quad (\text{C.57})$$

By Bayes rule, we have

$$\begin{aligned}
& \mathbb{P} \left(\sin(\theta_w^{(k)}) = 0 \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D}, \sin(\hat{\theta}_w^{(k)}) < \delta \right) \\
&= \mathbb{P} \left(\sin(\hat{\theta}_w^{(k)}) < \delta \mid \sin(\theta_w^{(k)}) = 0, \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) \\
&\quad \mathbb{P} \left(\sin(\theta_w^{(k)}) = 0 \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) / \mathbb{P} \left(\sin(\hat{\theta}_w^{(k)}) < \delta \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) \\
&= \mathbb{P} \left(\sin(\hat{\theta}_w^{(k)}) < \delta \mid \sin(\theta_w^{(k)}) = 0 \right) \mathbb{P} \left(\sin(\theta_w^{(k)}) = 0 \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) \\
&\quad / \mathbb{P} \left(\sin(\hat{\theta}_w^{(k)}) < \delta \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right). \tag{C.58}
\end{aligned}$$

Note that

$$\mathbb{P} \left(\sin(\theta_w^{(k)}) = 0 \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) = \frac{1}{i}. \tag{C.59}$$

Analogous to the derivation of Theorem 3.1, for $i \geq 1$, as $D \rightarrow \infty$, we have

$$\begin{aligned}
& \mathbb{P} \left(\sin(\hat{\theta}_w^{(k)}) < \delta \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) \\
&\geq \mathbb{P} \left(\sin(\hat{\theta}_w^{(k)}) < \delta, \sin(\theta_w^{(k)}) \leq 1/\sqrt{D} \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) \\
&= \mathbb{P} \left(\sin(\theta_w^{(k)}) \leq 1/\sqrt{D} \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) \\
&\quad \mathbb{P} \left(\sin(\hat{\theta}_w^{(k)}) < \delta \mid \sin(\theta_w^{(k)}) \leq 1/\sqrt{D}, \sin(\theta_w^{(k-1)}) = \sqrt{i/D} \right) \\
&= \frac{2}{i} \left(1 - O \left(\frac{\sqrt{D} \log D}{2^T} \right) \right). \tag{C.60}
\end{aligned}$$

Plugging Eq.(C.59) and Eq.(C.60) into Eq.(C.58), we get

$$\begin{aligned}
& \mathbb{P} \left(\sin(\theta_w^{(k)}) = 0 \mid \sin(\theta_w^{(k-1)}) = \sqrt{i/D}, \sin(\hat{\theta}_w^{(k)}) < \delta \right) \\
&= \mathbb{P} \left(\sin(\hat{\theta}_w^{(k)}) < \delta \mid \sin(\theta_w^{(k)}) = 0 \right) \frac{1}{i} / \left(\frac{2}{i} \left(1 - O \left(\frac{\sqrt{D} \log D}{2^T} \right) \right) \right) \\
&= \frac{1}{i} / \left(\frac{2}{i} \left(1 - O \left(\frac{\sqrt{D} \log D}{2^T} \right) \right) \right) \\
&\leq \frac{1}{2} + O \left(\frac{\sqrt{D} \log D}{2^T} / \left(1 - \frac{\sqrt{D} \log D}{2^T} \right) \right) \\
&\leq \frac{1}{2} + O \left(\frac{1}{2^T / (\sqrt{D} \log D) - 1} \right). \tag{C.61}
\end{aligned}$$

Plugging Eq.(C.61) into Eq.(C.57) and Eq.(C.58), we get

$$\begin{aligned}
& \mathbb{P} (\sin(\theta_w^{(K)}) = 0) \\
&\leq \frac{1}{2} \sum_{k=1}^D \sum_{i=1}^D \mathbb{P} \left(\sin(\theta_w^{(k-1)}) = \sqrt{i/D} \mid \sin(\hat{\theta}_w^{(k)}) < \delta \right) \mathbb{P}(K = k) \\
&= \frac{1}{2} + O \left(\frac{1}{2^T / (\sqrt{D} \log D) - 1} \right). \tag{C.62}
\end{aligned}$$

Thus, we have

$$\left| \mathbb{P}(|b_w\rangle \neq |b^\star\rangle) - \frac{1}{2} \right| = \left| \mathbb{P}(\sin(\theta_w^{(K)}) \neq 0) - \frac{1}{2} \right| = O \left(\frac{1}{2^T / (\sqrt{D} \log D) - 1} \right). \tag{C.63}$$

□

We shall now derive the time and space complexity of BGS. Recall Algorithm 1 is decomposed into the outer loop and the inner loop. Denote the number of inner loop iterations in the j th outer loop iteration by S_j , $j = 1, \dots, K$. The computation time of BGS is $\sum_{j=1}^K S_j$.

We present the following proposition regarding the algorithm's computational cost.

Proposition C.6 (Time complexity). *For Algorithm 1 with $T = \Theta(\log(\sqrt{D} \log D))$ and $\delta = \frac{1}{2\sqrt{D}}$, the following statements hold.*

(i) *Inner loop*:

$$\mathbb{E}(S_j) = O(1), \quad (\text{C.64})$$

for any $j \in \{1, 2, \dots, K\}$.

(ii) *Outer loop*:

$$\mathbb{E}(K) = O(\log D). \quad (\text{C.65})$$

(iii) *Quantum counting algorithm*: Let T_{QP} be the time complexity of applying the quantum counting algorithm once. We have

$$T_{\text{QP}} = O\left(\sqrt{D}(\log D)^2\right). \quad (\text{C.66})$$

(iv) *Grover's algorithm*: Let T_{Grover} be the time complexity of Grover's algorithm. We have

$$T_{\text{Grover}} = O\left(\sqrt{D} \log D\right). \quad (\text{C.67})$$

Proof of Proposition C.6. (i) Recall Proposition C.4 states that

$$\mathbb{P}(|b_{w^{\text{new}}} \rangle \notin \mathcal{B}_w) \leq \theta_w + \frac{1}{2^T \theta_w}.$$

Note that for $1/\sqrt{D} \leq \theta_w \leq \frac{\pi}{4}$, the term $\leq \theta_w + \frac{1}{2^T \theta_w}$ is maximized at either $\theta_w = 1/\sqrt{D}$ or $\theta_w = \frac{\pi}{4}$. For $\theta_w = 1/\sqrt{D}$, when $T > \log 2\sqrt{D}$, we have

$$\mathbb{P}(|b_{w^{\text{new}}} \rangle \notin \mathcal{B}_w) \leq \arcsin\left(1/\sqrt{D}\right) + \frac{1}{2^T \arcsin\left(1/\sqrt{D}\right)} < 1. \quad (\text{C.68})$$

For $\theta_w = 1/\sqrt{D}$, taking $T > \log 2\sqrt{D}$, we have

$$\mathbb{P}(|b_{w^{\text{new}}} \rangle \notin \mathcal{B}_w) \leq \frac{\pi}{4} + \frac{4}{2^T \pi} < 1. \quad (\text{C.69})$$

Thus, taking $T > \log 2\sqrt{D}$, $\mathbb{P}(|b_{w^{\text{new}}} \rangle \notin \mathcal{B}_w)$ is uniformly bounded away from 1. That is, there exists $0 < c < 1$ such that for any $\theta_w < \frac{\pi}{4}$,

$$\mathbb{P}(|b_{w^{\text{new}}} \rangle \notin \mathcal{B}_w) \leq 1 - c. \quad (\text{C.70})$$

Recall that for a geometric distribution with successful probability c , its expectation is $\frac{1}{1-c}$. Since the successful probability of the inner loop is larger than c , its expected time complexity is upper bounded by $\frac{1}{1-c}$. Specifically, for any $j \in \{1, 2, \dots, K\}$,

$$\mathbb{E}(S_j) \leq \frac{1}{1-c} = O(1). \quad (\text{C.71})$$

(ii) The outer loop is exactly the same as a random binary search, and the random binary search's expected time complexity is $O(\log D)$ (Martínez and Roura, 1998). Thus, we have $\mathbb{E}(K) = O(\log D)$.

(iii) Since we take $T = [\log(2\sqrt{D} \log D)]$, by Lemma 2.1, the time complexity of this quantum counting algorithm is

$$O(p2^T) = O\left(\sqrt{D}(\log D)^2\right).$$

(iv) As shown in ref. 16, Grover's algorithm needs $O(\sqrt{D})$ times of Grover's rotations. Note that the time complexity of one Grover's rotation is $O(\log D)$. Thus, the time complexity of the classic Grover's algorithm is $O(\sqrt{D} \log D)$. \square

Now, we are ready to present the proof of Theorem 3.3.

Proof of Theorem 3.3. By summing up all the operations and multiplying the number of (inner and outer) iterations in Algorithm 1, its expected time complexity is upper bounded by

$$\mathbb{E}\left((T_{\text{QP}} + T_{\text{Grover}}) \sum_{j=1}^K S_j\right) \leq \left[O\left(\sqrt{D}(\log D)^2\right) + O\left(\sqrt{D} \log D\right)\right] \mathbb{E}\left(\sum_{j=1}^K S_j\right). \quad (\text{C.72})$$

We now derive the bound of $\mathbb{E}\left(\sum_{j=1}^K S_j\right)$. Let $\Theta = \{\theta_w^{(1)}, \dots, \theta_w^{(K)}\}$ be the set of θ_w 's in the 1st to the K th outer loop iterations in Algorithm 1. Note that both $\theta_w^{(j)}$ and K are

random variables. Given Θ , K is fixed. Let $S = \{S_i, \dots, S_K\}$. S and K are conditionally independent given Θ . We thus have

$$\mathbb{E}_S \left(\sum_{j=1}^K S_j \mid \Theta \right) = \sum_{j=1}^K \mathbb{E}_S \left(S_j \mid \theta_w^{(j)} \right). \quad (\text{C.73})$$

Therefore, by Eq.(C.73) and Proposition C.6, we have

$$\begin{aligned} \mathbb{E} \left(\sum_{j=1}^K S_j \right) &= \mathbb{E} \left(\mathbb{E}_S \left(\sum_{j=1}^K S_j \mid \Theta \right) \right) \\ &= \mathbb{E} \left(\sum_{j=1}^K \mathbb{E}_S \left(S_j \mid \theta_w^{(j)} \right) \right) \\ &\leq \mathbb{E} \left(\sum_{j=1}^K O(1) \right) \\ &\leq \mathbb{E}(K) O(1) \\ &= O(\log D). \end{aligned} \quad (\text{C.74})$$

Combining Eq.(C.72) and Eq.(C.74), we have the time complexity of Algorithm 1

$$\left[O \left(\sqrt{D} (\log D)^2 \right) + O(\sqrt{D} \log D) \right] O(\log D) O(1) = O \left(\sqrt{D} (\log D)^3 \right). \quad (\text{C.75})$$

For the space complexity, since we take $T = [\log(2\sqrt{D} \log D)]$, by Lemma 2.1, the quantum counting algorithm takes $O(\log(\sqrt{D} \log D))$ extra qubits. Recall that the space complexity of the remaining part of Algorithm 1 is the same as that of Grover's algorithm, which is $O(\log D)$. Overall, the space complexity of Algorithm 1 is $O(\log \sqrt{D} + \log \log D) + O(\log D) = O(\log D)$. \square

D Additional results of the simulation study on the comparison of BGS and BSS in logistic regression models

For this simulation study, replicated samples are generated independently according to the logistic regression model,

$$\mathbb{P}\left(z_i = 1 \mid \boldsymbol{\alpha}; \mathbf{x}_i\right) = \frac{1}{1 + \exp(-\mathbf{x}_i^\top \boldsymbol{\alpha})}, \quad i = 1, \dots, n, \quad (\text{D.76})$$

where $z_i \in \{0, 1\}$, $\mathbf{x}_i \in \mathbb{R}^p$, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)^\top \in \mathbb{R}^p$. We set $n = 2,000$ and generate $\{\mathbf{x}_i\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim} N_p(-\frac{3}{p}\mathbf{1}_p, \boldsymbol{\Sigma})$, where $\mathbf{1}_p$ is the p -dimension vector with all entries being 1. We set the (i, j) th entry of $\boldsymbol{\Sigma}$ to be $0.1^{|i-j|}$. The first $\lfloor p/2 \rfloor$ entries in $\boldsymbol{\alpha}$ are set to 1, while the remaining entries are set to 0. Note that we let \mathbf{x}_i have a negative mean of $-\frac{3}{p}\mathbf{1}_p$ to ensure the response z_i having an unbalanced distribution with about 30% of the sample being zero. We generate 100 replicated samples of size n for each $p = \{3, 4, \dots, 13\}$.

To evaluate the subset selection accuracy of the BGS algorithm, we compare it with the best subset selection (BSS) via emulation in a classic computer. For the BSS, the subset with the smallest BIC is selected. We report the frequency of the true subset being selected among 100 replicated samples in Table 2.

Table 2: The percentage of the true subset being selected among 100 replicates by BSS and proposed BGS in weighted logistic regression.

method	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$	$p = 9$	$p = 10$	$p = 11$	$p = 12$	$p = 13$
BSS	97%	96%	88%	92%	90%	95%	90%	85%	93%	87%	84%
BGS	97%	96%	88%	92%	89%	95%	90%	85%	92%	85%	84%

We observe high frequencies (around 90%) in Table 2 for both BGS and BSS across all settings. The frequency of the true subset being selected by BGS is almost identical to that of BSS under all settings.

E Identification of ADTs associated with the targeted genes

Preprocessing of RNA and ATD data in CITE-seq. The workflow for computing RNA and ADT expressions in CITE-seq data is given as follows:

RNA expressions: We utilize the standard pipelines available in Seurat package V5 ([Hao et al., 2023](#)). This pipeline includes essential steps such as normalization (using the “NormalizeData” function) and feature scaling (using the “ScaleData” function). In the normalization step, we use “normalization.method = LogNormalize” and “scale.factor = 1000” in the “NormalizeData” function. All other parameters are kept at their default values.

ADT expressions: We use Seurat package V5 and normalize the ADT expression levels within each cell using the centered-log ratio (CLR) transform. Subsequently, we perform feature scaling and centering using the "ScaleData" function. The CLR transform is achieved by using the “NormalizeData” in Seurat by setting “normalization.method = ‘CLR’ ” and “margin = 2”. The remaining parameters are set to their default values.

ADTs identification using Seurat and SCANPY

1. Seurat Methods: We apply the differential expression methods available in the Seurat package V5 with six different options: Wilcoxon rank sum test (Seurat-wilcox), Student’s t-test (Seurat-t), likelihood-ratio test (Seurat-bimod), ROC analysis (Seurat-roc), logistic regression-based method (Seurat-LR), and hurdle model-based method (Seurat-MAST). In particular, we use the “FindMarkers” function in Seurat package V5. The “test.use” parameter is set to the specific method, such as “wilcox”, “t”, “bimod”, “roc”, “LR”, or “MAST”. All the other parameters are set to their default values.

2. SCANPY Methods: We also apply the differential expression methods available in the SCANPY package V1.9.2 with three different options: Wilcoxon rank sum test (SCANPY-wilcox), Student’s t-test (SCANPY-t), and logistic regression-based method (SCANPY-LR). We use the “tl.rank_genes_groups” function, with the “rankby_abs = True” parameter. The “method” parameter is set to the specific method, such as “wilcoxon”, “logreg”, or “t-test”. All the other parameters are set to their default values.

To utilize the differential expression methods in Seurat and SCANPY packages, we first define two groups of cells based on whether the CD14 RNA level is zero. We then apply each differential expression method to identify and rank the differentially expressed ADTs between the two groups. The output of both Seurat and SCANPY methods is a list of ranked significant variables (ADTs). For a fair comparison, we select the same number of ADTs from the rank list output by Seurat or SCANPY. We vary the number of ADTs to be selected to 2, 4, 6, 8, or 10.

Using the selected ADTs, we fit a linear regression model by regressing CD14 RNA expressions on the expressions of the selected ADTs to the training dataset. We then make predictions and compute the MSE on the testing dataset.

F Panel design for cell type identification

Preprocessing: expression quantification for ATD in CITE-seq. The workflow for computing ADT expressions in CITE-seq is the same as the workflow in section E.

Cell type identification using Seurat and SCANPY. Depending on whether the cell is a cell type of interest, we classify all cells into two groups. We then apply the differential expression methods in the Seurat package and SCANPY package as presented in Section E.

G Additional results of Identification of ADTs associated with the targeted genes

Umap visualization of human cord blood mononuclear cells. To reduce the dimensionality of the ADTs expression data, we employ UMAP and present the visualization of the expression levels of 13 ADTs and RNA CD14 in Fig. 7. Our analysis reveals the presence of strong co-expression patterns among specific ADTs. For example, the CD3 and CD4 ADTs exhibit co-expression within a larger cell cluster, indicating a high likelihood of being CD4 T cells.



Figure 7: Umap visualization of the CBMC CITE-seq dataset. The cells are projected into a 2-dimension domain via applying UMAP on the normalized ADTs' expression levels. In each subfigure, the color darkness of each point represents the expression level of the corresponding ADT/RNA in a cell.

H Additional results of panel design for cell type identification.

Proportions of the 41 major cell types. We present the proportions of the 41 major cell types using pie charts in Fig. 8. Notably, we have observed a highly unbalanced distribution of cell samples. For instance, pDC cells account for only 0.5% of the total cell population.

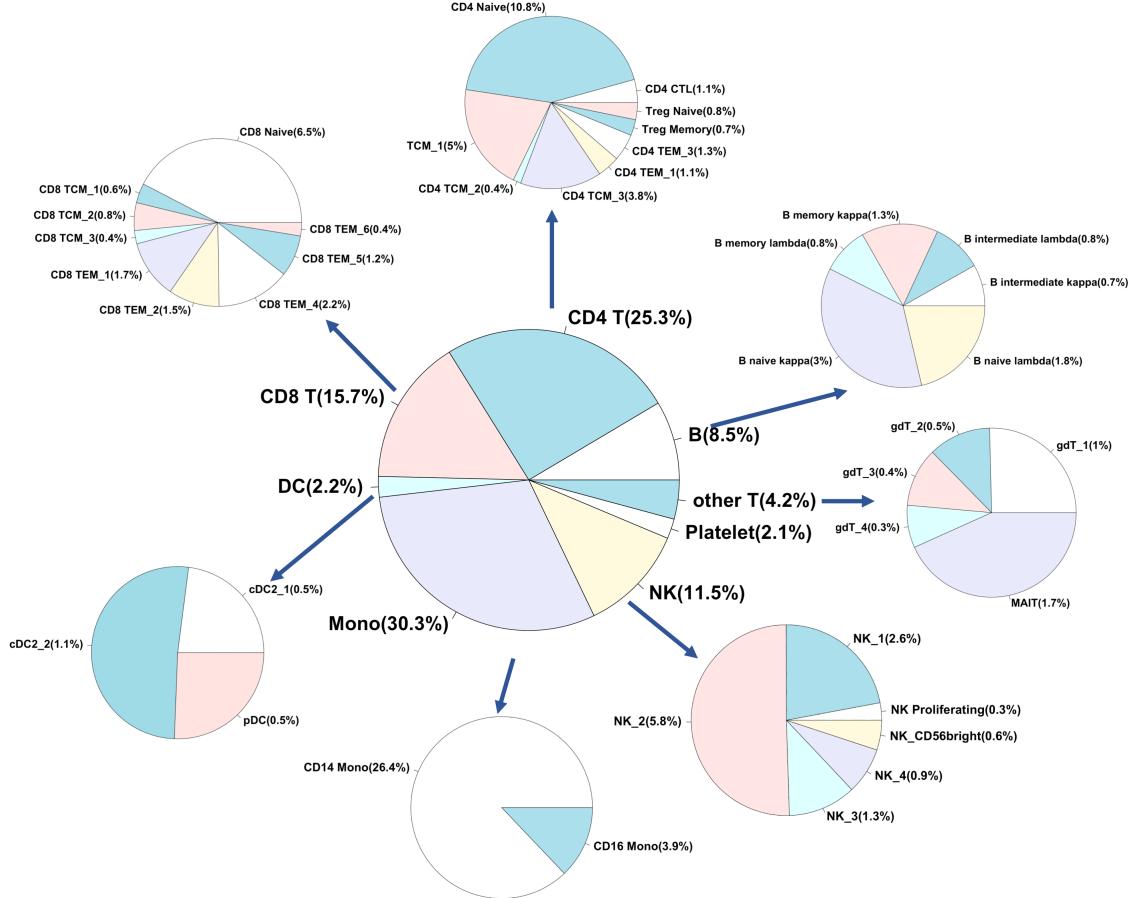


Figure 8: Pie charts of the cell types at two hierarchical levels. At level 1, the coarse cell types include CD4 T, CD8 T, B, NK, DC, Mono, other T, and Platelet cells. Level 2 further divides these cell types into 41 major categories, providing more detailed subdivisions of the level 1 cell types. The pie charts visually present the percentage of each cell type relative to the entire cell population.

Comparison between BGS and BSS for PBMC CITE-seq data. Fig. 9 displays a comparison between the BGS algorithm and the BSS method across panel sizes ranging from 1 to 9. Notably, we observe that the performance of the BGS algorithm closely aligns with that of BSS, indicating their almost identical performance. Additionally, our findings indicate that once the panel size reaches 4, the prediction AUCs for all cell types consistently surpass 0.95. Subsequently, there is no significant increase in AUCs as the panel size continues to increase.

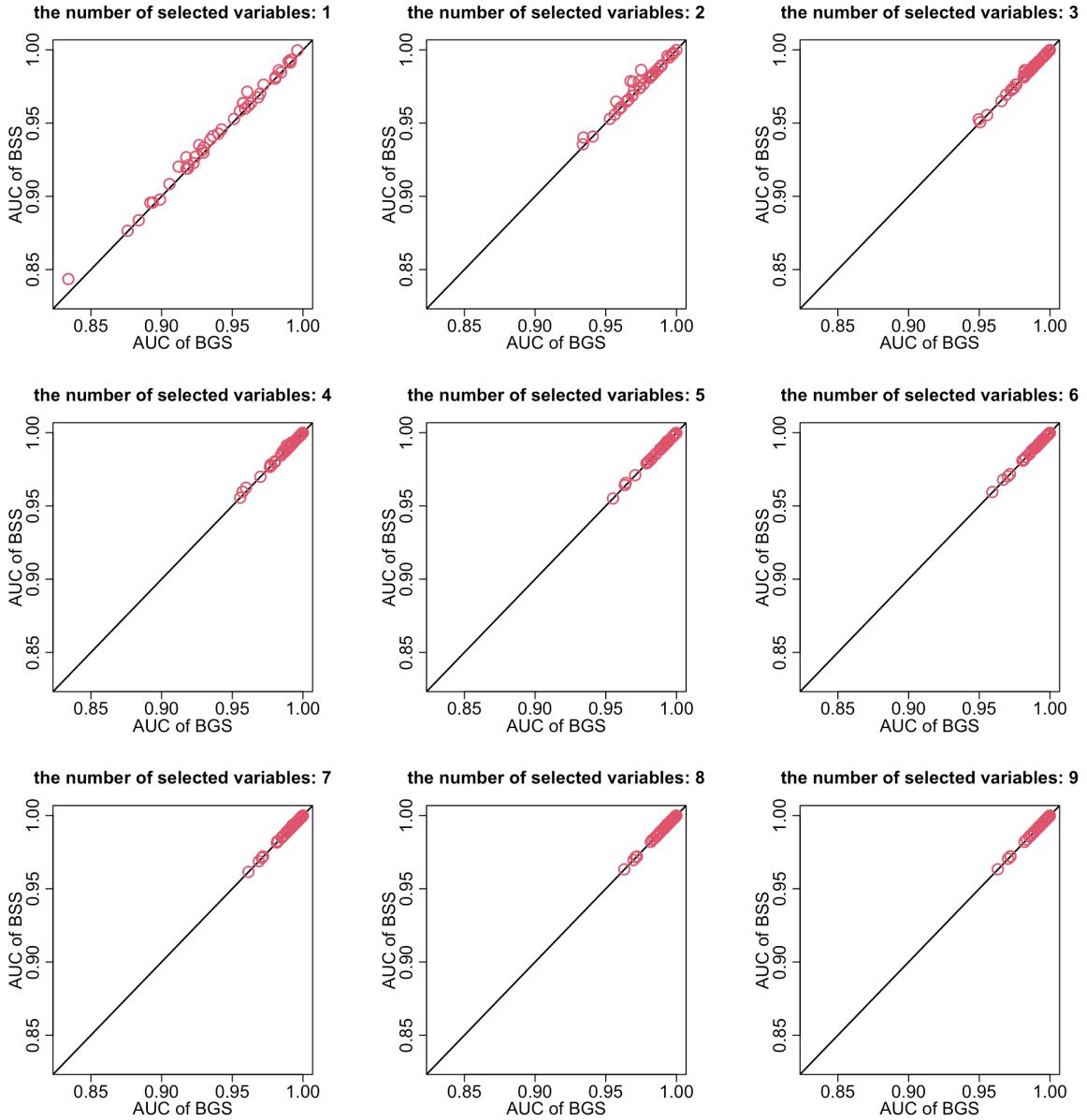


Figure 9: Comparison of the BGS algorithm and the BSS method for panel sizes ranging from 1 to 9. In each subplot, each dot represents a cell type. The X-axis is the average value of the AUCs of BGS on testing datasets for 100 replicated samples. The Y-axis is the average value of the AUCs of BSS on testing datasets for 100 replicated samples.

Performance comparison of BGS, Seurat and SCANPY for cell type prediction. We present the AUCs of BGS, Seurat and SCANPY for each cell type on the testing datasets for various panel sizes in Fig. 10, 11, 12.

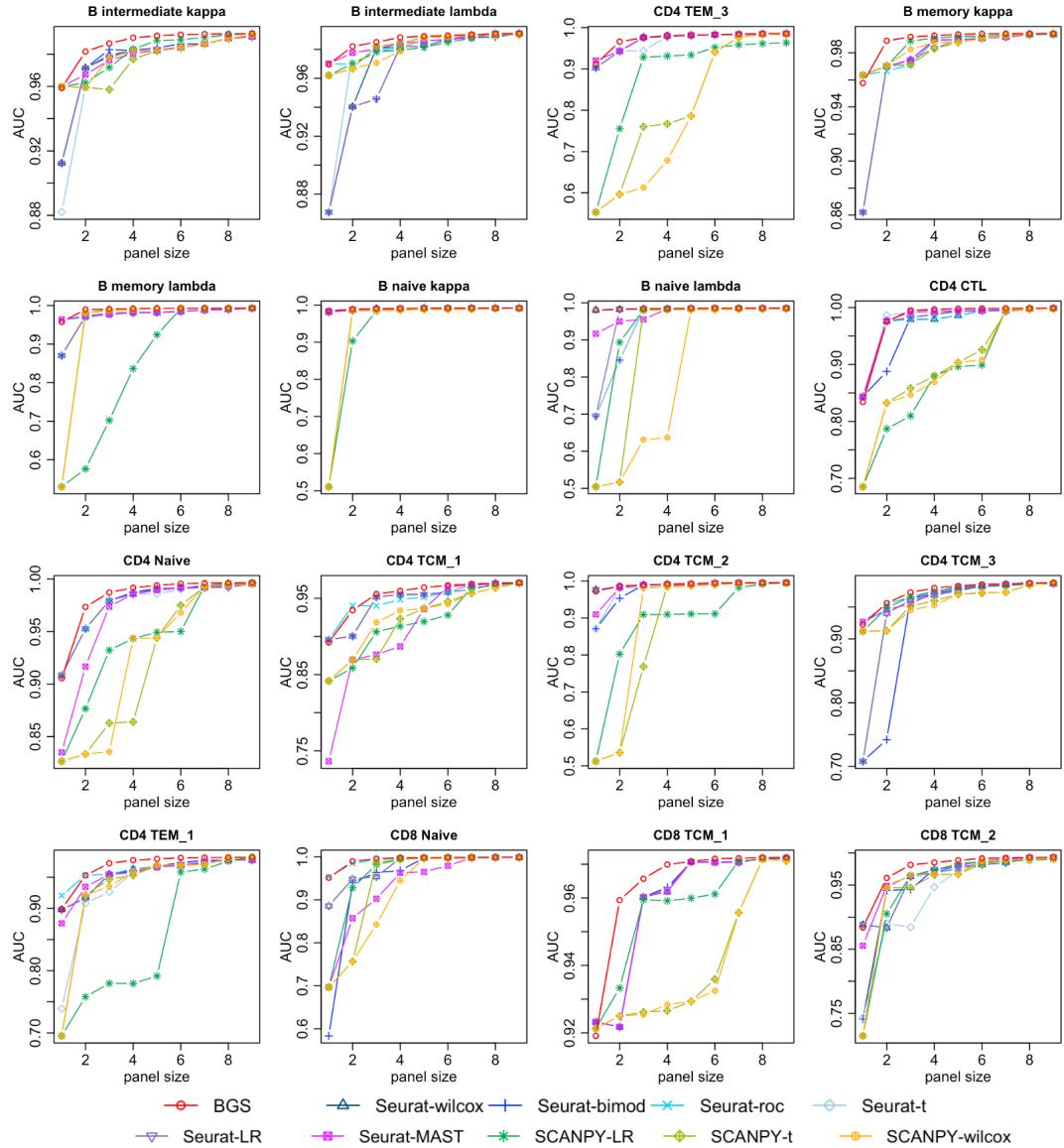


Figure 10: Performance comparison of cell type prediction. The X-axis indicates the panel size, while the Y-axis represents the AUC values of these methods on the testing dataset. Different methods are distinguished by various colors and point shapes.

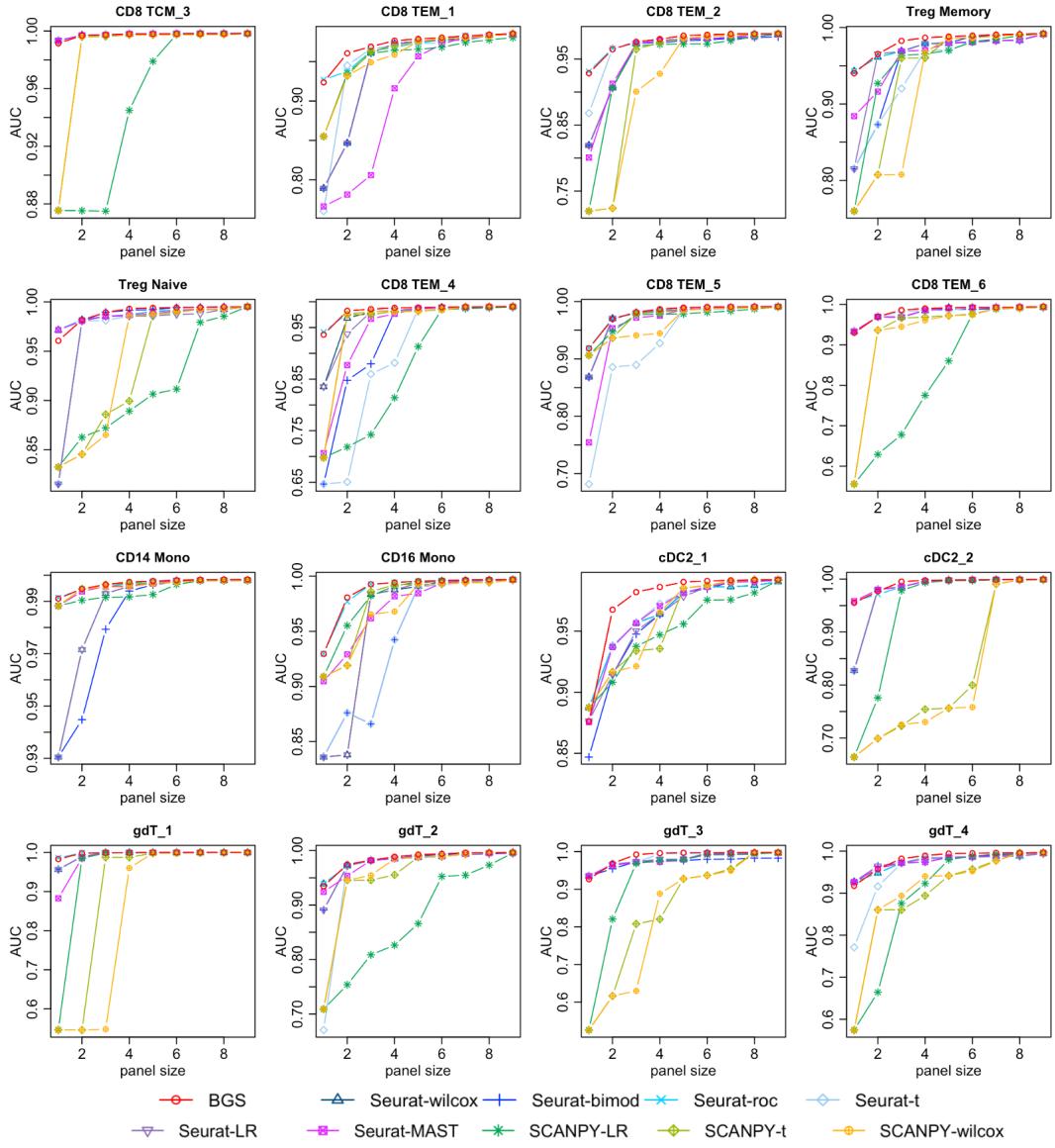


Figure 11: Performance comparison of cell type prediction. The X-axis indicates the panel size, while the Y-axis represents the AUC values of these methods on the testing dataset. Different methods are distinguished by various colors and point shapes.

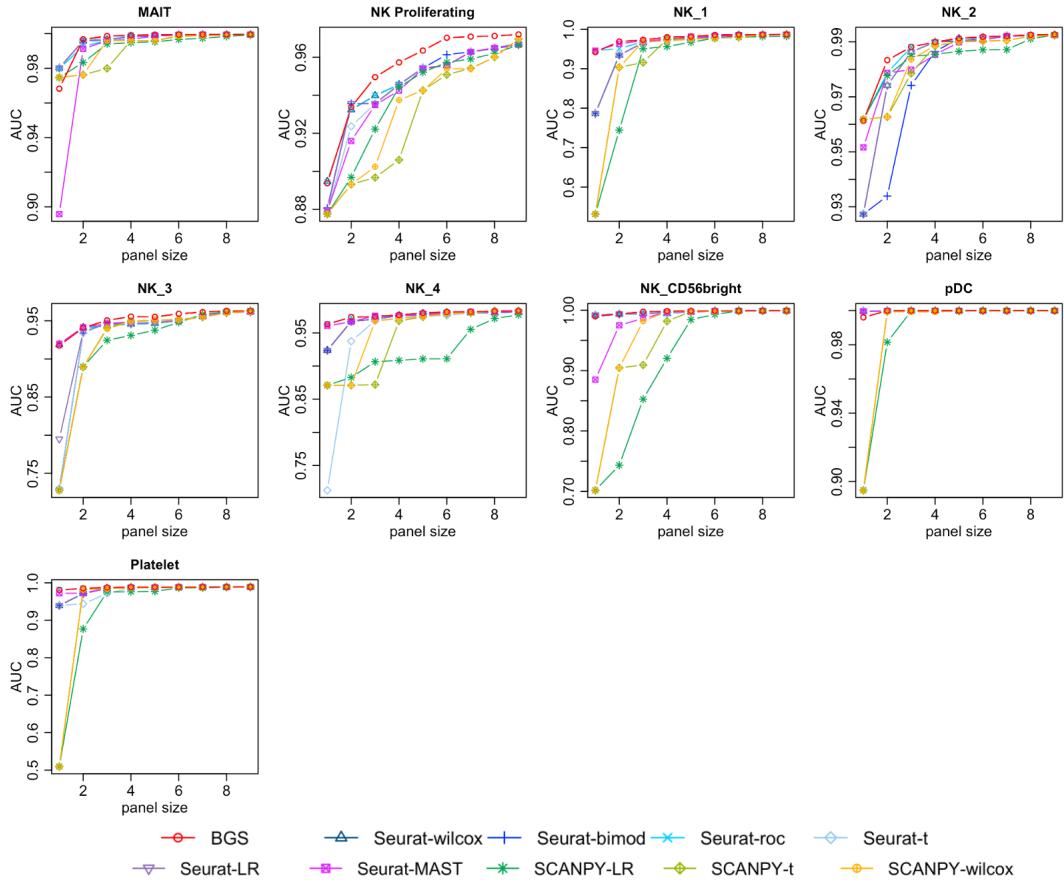


Figure 12: Performance comparison of cell type prediction. The X-axis indicates the panel size, while the Y-axis represents the AUC values of these methods on the testing dataset. Different methods are distinguished by various colors and point shapes.

The best four-marker panels selected by BGS for three cell types. Fig. 13 (a)

The best four-marker panels selected by BGS for Mono cells, other T cells, Platelet, and DC cells. Some markers in the selected panels are consistent with the markers reported in the literature, such as $CD14^+CD16^-$ for CD14 Mono cells, $CD16^+$ for CD16 Mono cells, $CD161^+$ for MAIT cells, $CD304^+CD123^+$ for pDC cells.

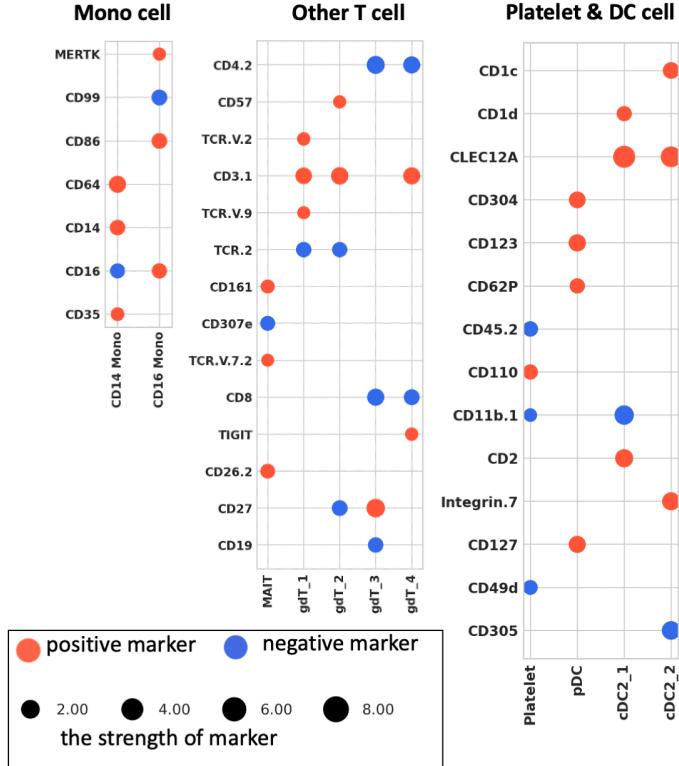


Figure 13: Panels with four markers selected by BGS for Mono cells, other T cells, Platelet and DC cells.

References

- Abbas, A., D. Sutter, C. Zoufal, A. Lucchi, A. Figalli and S. Woerner (2021). The power of quantum neural networks. *Nature Computational Science* 1(6), 403–409.
- Agarwal, D., J. Wang and N. R. Zhang (2020). Data denoising and post-denoising corrections in single cell rna sequencing. *Statistical Science* 35(1), 112–128.

Agresta, L., K. H. Hoebe and E. M. Janssen (2018). The emerging role of cd244 signaling

in immune cells of the tumor microenvironment. *Frontiers in immunology* 9, 2809.

Arute, F., K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends et al. (2019). Quantum

supremacy using a programmable superconducting processor. *Nature* 574(7779), 505–

510.

Beer, K., D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann et al.

(2020). Training deep quantum neural networks. *Nature communications* 11(1), 808.

Bennett, C. H., E. Bernstein, G. Brassard and U. Vazirani (1997). Strengths and weaknesses

of quantum computing. *SIAM Journal on Computing* 26(5), 1510–1523.

Berard, M. and D. F. Tough (2002). Qualitative differences between naive and memory t

cells. *Immunology* 106(2), 127.

Biamonte, J., P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe and S. Lloyd (2017). Quan-

tum machine learning. *Nature* 549(7671), 195–202.

Brassard, G., P. Hoyer, M. Mosca and A. Tapp (2002). Quantum amplitude amplification

and estimation. *Contemporary Mathematics* 305, 53–74.

Brassard, G., P. Høyer and A. Tapp (1998). Quantum counting. In *International Collo-*

quium on Automata, Languages, and Programming, pp. 820–831. Springer.

Buhrman, H., R. Cleve, J. Watrous and R. De Wolf (2001). Quantum fingerprinting.

Physical review letters 87(16), 167902.

Castelvecchi, D. (2023). IBM releases first-ever 1,000-qubit quantum chip. *Na-*

ture 624(7991), 238–238.

Chakraborty, S., S. H. Shaikh, A. Chakrabarti and R. Ghosh (2020). A hybrid quantum

feature selection algorithm using a quantum inspired graph theoretic approach. *Applied*

Intelligence 50(6), 1775–1793.

Farag, S. S. and M. A. Caligiuri (2006). Human natural killer cell development and biology.

Blood reviews 20(3), 123–137.

Ferrer-Font, L., S. J. Small, B. Lewer, K. R. Pilkington, L. K. Johnston, L. M. Park et al. (2021). Panel optimization for high-dimensional immunophenotyping assays using full-spectrum flow cytometry. *Current Protocols* 1(9), e222.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, New York, NY, USA, pp. 212–219. Association for Computing Machinery.

Hao, Y., S. Hao, E. Andersen-Nissen, W. M. Mauck, S. Zheng, A. Butler et al. (2021). Integrated analysis of multimodal single-cell data. *Cell* 184(13), 3573–3587.

Hao, Y., T. Stuart, M. H. Kowalski, S. Choudhary, P. Hoffman, A. Hartman et al. (2023). Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, 1–12.

Harrow, A. W., A. Hassidim and S. Lloyd (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters* 103(15), 150502.

He, Z., L. Li, Z. Huang and H. Situ (2018). Quantum-enhanced feature selection with forward selection and backward elimination. *Quantum Information Processing* 17, 1–11.

Hughston, L. P., R. Jozsa and W. K. Wootters (1993). A complete classification of quantum ensembles having a given density matrix. *Physics Letters A* 183(1), 14–18.

Kaminski, D. A., C. Wei, Y. Qian, A. F. Rosenberg and I. Sanz (2012). Advances in human b cell phenotypic profiling. *Frontiers in immunology* 3, 302.

King, G. and L. Zeng (2001). Logistic regression in rare events data. *Political Analysis* 9(2), 137–163.

Koike, T. and Y. Okudaira (2009, 10). Time complexity and gate complexity. *Physical Review. A* 82.

Li, Y., R.-G. Zhou, R. Xu, J. Luo, W. Hu and P. Fan (2022). Implementing graph-theoretic feature selection by quantum approximate optimization algorithm. *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, W., A. L. Putnam, Z. Xu-Yu, G. L. Szot, M. R. Lee, S. Zhu et al. (2006). Cd127 expression inversely correlates with foxp3 and suppressive function of human cd4+ t reg cells. *The Journal of experimental medicine* 203(7), 1701–1711.

Liu, Y., Y. Chen, H. Lu, W. Zhong, G.-C. Yuan and P. Ma (2024). Orthogonal multi-modality integration and clustering in single-cell data. *BMC bioinformatics* 25(1), 164.

Martínez, C. and S. Roura (1998). Randomized binary search trees. *Journal of the ACM (JACM)* 45(2), 288–323.

Mücke, S., R. Heese, S. Müller, M. Wolter and N. Piatkowski (2023). Feature selection on quantum computers. *Quantum Machine Intelligence* 5(1), 11.

Nguyen, H. H., T. Kim, S. Y. Song, S. Park, H. H. Cho, S.-H. Jung et al. (2016). Naïve cd8+ t cell derived tumor-specific cytotoxic effectors as a potential remedy for overcoming tgf- β immunosuppression in the tumor microenvironment. *Scientific reports* 6(1), 1–10.

Nielsen, M. A. and I. L. Chuang (2010). *Quantum Computation and Quantum Information*. Cambridge University Press.

Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum* 2, 79.

Rawat, K., A. Pal, S. Banerjee, A. Pal, S. C. Mandal and S. Batabyal (2021). Ovine cd14-an immune response gene has a role against gastrointestinal nematode haemonchus contortus—a novel report. *Frontiers in immunology* 12.

- Schrödinger, E. (1935). Discussion of probability relations between separated systems. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Volume 31, pp. 555–563. Cambridge University Press.
- Schroeder, B. A., N. A. LaFranzo, B. J. LaFleur, R. M. Gittelman, M. Vignali, S. Zhang et al. (2021). Cd4+ t cell and m2 macrophage infiltration predict dedifferentiated liposarcoma patient outcomes. *Journal for immunotherapy of cancer* 9(8).
- Schuld, M., I. Sinayskiy and F. Petruccione (2016). Prediction by linear regression on a quantum computer. *Physical Review A* 94(2), 022342.
- Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, 461–464.
- Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review* 41(2), 303–332.
- Stoeckius, M., C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow et al. (2017). Large-scale simultaneous measurement of epitopes and transcriptomes in single cells. *Nature methods* 14(9), 865.
- Turati, G., M. F. Dacrema and P. Cremonesi (2022). Feature selection for classification with qaoa. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 782–785. IEEE.
- Von Dollen, D., F. Neukart, D. Weimer and T. Bäck (2021). Quantum-assisted feature selection for vehicle price prediction modeling. *arXiv preprint arXiv:2104.04049*.
- Wang, Y. (2022). When quantum computation meets data science: Making data science quantum. *Harvard Data Science Review* 4(1).
- Wang, Y. and H. Liu (2022). Quantum computing in a statistical context. *Annual Review of Statistics and Its Application* 9, 479–504.

- Wang, Y. and X. Song (2020). Quantum science and quantum technology. *Statistical Science* 35(1), 51–74.
- Weigold, M., J. Barzen, F. Leymann and M. Salm (2020). Data encoding patterns for quantum computing. In *Proceedings of the 27th Conference on Pattern Languages of Programs*, pp. 1–11.
- White, J. T., E. W. Cross and R. M. Kedl (2017). Antigen-inexperienced memory cd8+ t cells: where they come from and why we need them. *Nature Reviews Immunology* 17(6), 391–400.
- Wolf, F. A., P. Angerer and F. J. Theis (2018). Scanpy: large-scale single-cell gene expression data analysis. *Genome biology* 19, 1–5.
- Wootters, W. K. and W. H. Zurek (1982). A single quantum cannot be cloned. *Nature* 299(5886), 802–803.
- Wu, Y., W.-S. Bao, S. Cao, F. Chen, M.-C. Chen, X. Chen et al. (2021). Strong quantum computational advantage using a superconducting quantum processor. *Physical review letters* 127(18), 180501.
- Zamdborg, L. and P. Ma (2009). Discovery of protein–dna interactions by penalized multivariate regression. *Nucleic acids research* 37(16), 5246–5254.
- Zhong, H.-S., H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo et al. (2020). Quantum computational advantage using photons. *Science* 370(6523), 1460–1463.
- Zhong, W., P. Zeng, P. Ma, J. S. Liu and Y. Zhu (2005, 09). RSIR: regularized sliced inverse regression for motif discovery. *Bioinformatics* 21(22), 4169–4175.