Quantum Information and Computation, Vol. 14, No. 5&6 (2014) 0439–0453

QUANTUM ALGORITHMS FOR SEARCH WITH WILDCARDS AND COMBINATORIAL GROUP TESTING

ANDRIS AMBAINIS
Faculty of Computing, University of Latvia, Raina bulv. 19
Riga, LV-1586, Latvia

ASHLEY MONTANARO
Department of Computer Science, University of Bristol, Woodland Rd.
Bristol, BS8 1UB, UK

Received April 15, 2013
Revised August 14, 2013

We consider two combinatorial problems. The first we call "search with wildcards": given an unknown n-bit string x, and the ability to check whether any subset of the bits of x is equal to a provided query string, the goal is to output x. We give a nearly optimal $O(\sqrt{n}\log n)$ quantum query algorithm for search with wildcards, beating the classical lower bound of $\Omega(n)$ queries. Rather than using amplitude amplification or a quantum walk, our algorithm is ultimately based on the solution to a state discrimination problem. The second problem we consider is combinatorial group testing, which is the task of identifying a subset of at most k special items out of a set of n items, given the ability to make queries of the form "does the set S contain any special items?" for any subset S of the n items. We give a simple quantum algorithm which uses $O(k)$ queries to solve this problem, as compared with the classical lower bound of $\Omega(k\log(n/k))$ queries.

Keywords: Quantum query complexity, oracle interrogation, combinatorial group testing.
Communicated by: R Cleve & R de Wolf

1 Introduction

We present new quantum algorithms for two combinatorial problems. The first problem is *search with wildcards*. In this problem, we are given an n-bit string x and our task is to determine x (so that with probability $1-\epsilon$, all bits of x are correct) using the minimum number of queries in the following *wildcard query* model. In one wildcard query, we can check correctness of any subset of the bits of x. That is, we identify queries with pairs (S,y), where $S\subseteq[n]$ and $y\in\{0,1\}^{|S|}$ and the query returns 1 if $x_S=y$ (here the notation x_S means the subset of the bits of x specified by S).

Wildcard queries are a generalisation of the standard quantum query model; the standard model corresponds to queries in which S contains just one element. Classically, each query in this more general model still provides only one bit of information. Hence, by an information-theoretic argument classical computers still require $\Omega(n)$ queries to solve search with wildcards. Moreover, in the standard quantum query model, identifying x with bounded error would require $\Omega(n)$ queries [1, 2]. Surprisingly, in contrast to these two lower bounds, we have the following theorem.

Theorem 1. *There is a quantum algorithm which solves the search with wildcards problem with worst-case failure probability at most* $1/3$ *using* $O(\sqrt{n}\log n)$ *queries. Further, any bounded-error quantum algorithm which solves this problem must make* $\Omega(\sqrt{n})$ *queries.*

Observe that any (classical or quantum) bounded-error algorithm for search with wildcards which makes q queries can be converted into a Las Vegas algorithm (i.e. one which always succeeds eventually) which makes an expected number of queries which is $O(q)$, as a claimed solution can be checked with one additional query. Rather than using the usual methods of designing quantum algorithms (such as amplitude amplification or quantum walks), our algorithm is based on a novel information-theoretic idea. Our algorithm gradually increases the information about the input string x by repeatedly using the Pretty Good Measurement (PGM) [3, 4] to distinguish a set of quantum states. With one query, we can increase the knowledge about the input x from k bits to $k + \Theta(\sqrt{k})$ bits – which leads to a quantum algorithm using $O(\sqrt{n}\log n)$ queries. We think that this idea (and the natural state distinguishability problem that we solve, in Lemma 3), may be of independent interest and may find more applications.

The second problem is the well known *combinatorial group testing* (CGT). In this problem, we are given oracle access to an n-bit string x such that the Hamming weight of x is at most k. We usually assume that k is much smaller than n. In one query, we can get the OR of an arbitrary subset of the bits of x. The goal is to determine x using the minimal expected number of queries. This models a scenario where we would like to identify a small subset of special items out of a large set of items, given the ability to make queries of the form "does the set S contain any special items?" for any subset S of the items.

The idea of combinatorial group testing[a] dates back to 1943, when it was proposed as a means of identifying and rejecting syphilitic men called up for induction into the US military [5]. Following this seminal work, a vast literature on the subject has developed; see the textbook [6] for a detailed review, or the paper [7] for a discussion of more recent work. Areas to which efficient algorithms for CGT have been applied include molecular biology [8], data streaming algorithms [9], compressed sensing [10], and pattern matching in strings [11].

Classically, it is well-known that the number of queries required to solve CGT is of order $\Theta(k\log(n/k))$ [6]. The lower bound is an information-theoretic argument while the upper bound is based on binary search. In the quantum case, we have the following result.[b]

Theorem 2. *There is a bounded-error quantum algorithm which solves the combinatorial group testing problem using* $O(k)$ *queries. Further, any quantum algorithm which solves CGT with bounded error must make* $\Omega(\sqrt{k})$ *queries.*

Note that our Theorem has no dependence on n (unlike the classical complexity). We prove Theorem 2 in two parts: a $O(k)$-query quantum algorithm in Section 4 below, and a

[a] CGT is sometimes simply known as "group testing"; we prefer the inclusion of "combinatorial" to avoid confusion with the notion of testing a set for being a group.

[b] A previous version of this paper claimed an upper bound of $O(\sqrt{k}\,\mathrm{polylog}(k))$ queries, via a reduction to search with wildcards. However, the reduction was incorrect and the precise quantum query complexity of CGT remains open.

$\Omega(\sqrt{k})$ quantum lower bound in Section 5. Each part of the result is fairly straightforward.

1.1 Related work

One can view the search with wildcards problem as oracle interrogation – i.e. learning the contents of an unknown bit-string x hidden in an oracle – in a non-standard oracle model. There has recently been some interest in this problem, in various different oracle models; we summarise the results which have been obtained as follows.

- First, it was shown by van Dam [12] that in the standard oracle model (where the oracle performs the map $i \mapsto x_i$), there exists a quantum algorithm which learns x with constant success probability using $n/2 + O(\sqrt{n})$ queries, contrasting with the n classical queries required to learn x. Farhi et al. [13] later showed a matching $n/2 + \Omega(\sqrt{n})$ lower bound.

- Iwama et al. have studied the quantum query complexity of counterfeit coin problems [14]. Here we are given a set of n coins, k of which are false (underweight), and the task is to determine the false coins. In this model, a query is specified by $q \in \{0, 1, -1\}^n$ such that $\sum_i q_i = 0$. Then the oracle returns 0 if $q \cdot x = 0$, and 1 otherwise. We imagine that x is a set of coins, and $x_i = 0$ if the i'th coin is fair, and $x_i = 1$ if the i'th coin is false. The oracle simulates a "quantum scale", and $q_i = 1$ (resp. $q_i = -1$) means that we place the i'th coin on the left (resp. right) pan. If the oracle returns 0, the scale is balanced, and if it returns 1, the scale is unbalanced. Iwama et al. showed that there is a quantum algorithm based on amplitude amplification which solves this problem using only $O(k^{1/4})$ queries, beating the classical information-theoretic lower bound of $\Omega(k \log(n/k))$ queries. Note that, similarly to our algorithm for CGT, their result removes any dependence on n from the complexity.

- Finally, recently Cleve et al. have studied oracle interrogation in the model of substring queries [15]. Here the allowed queries are of the form "is y a substring of x?" for $y \in \{0, 1\}^k$, $1 \leq k \leq n$, where a substring of x is a consecutive subsequence of x. Classically, this problem again requires n queries; Cleve et al. proved that quantum algorithms can achieve a linear speedup, giving an algorithm which uses $3n/4 + o(n)$ queries. They also show an $\Omega(n/\log^2 n)$ quantum lower bound.

We also remark that the problem of identifying ("learning") an unknown oracle has been considered in great generality by Ambainis et al. [16, 17] and Atıcı and Servedio [18]. The algorithms given in these works could be applied to the problems studied here, but would not give a bound stronger than $O(n)$ queries in the case of search with wildcards, and $O(k \log n)$ in the case of CGT.

1.2 Preliminaries and notation

We write $[n] := \{1, 2, \ldots, n\}$, and use $|x|$ for the Hamming weight of x and $d(x, y)$ for the Hamming distance between x and y. For $x \in \{0, 1\}^n$, a 1-index (resp. 0-index) of x is an index $i \in [n]$ such that $x_i = 1$ (resp. $x_i = 0$). For readability, we sometimes leave states unnormalised. The two problems that we consider are precisely defined as follows:

- SEARCH WITH WILDCARDS. We are given oracle access to an n-bit string x (with no restriction on Hamming weight) and our task is to determine x using the minimum number of queries. A query is specified by a string $s \in \{0, 1, *\}^n$, and returns 1 if $x_i = s_i$ for all i such that $s_i \neq *$, and returns 0 otherwise. We can equivalently identify queries with pairs (S, y), where $S \subseteq [n]$ and $y \in \{0, 1\}^{|S|}$ and the query $Q_x(S, y)$ returns 1 if $x_S = y$ (here the notation x_S means the subset of the bits of x specified by S). We therefore sometimes refer to queries in this model as subset queries. In the case of quantum algorithms, we give the algorithm access to the unitary oracle which maps $|S\rangle|y\rangle|z\rangle \mapsto |S\rangle|y\rangle|z \oplus Q_x(S, y)\rangle$.
- COMBINATORIAL GROUP TESTING (CGT). We are given oracle access to an n-bit string x such that the Hamming weight of x is at most k; again, our task is to determine x. We usually assume that k is much smaller than n. We are allowed to query arbitrary subsets $S \subseteq [n]$ of the bits of x; a query $Q_x(S)$ returns 1 if there exists $i \in S$ such that $x_i = 1$. In the case of quantum algorithms, we give the algorithm access to the unitary oracle which maps $|S\rangle|z\rangle \mapsto |S\rangle|z \oplus Q_x(S)\rangle$.

We note that search with wildcards is a special case of CGT. Consider an instance of CGT where $k \leq n/2$ and the input is divided into k blocks $B_i = \{2i - 1, 2i\}$ of size 2, $1 \leq i \leq k$, followed by a final block of $n - 2k$ bits. The input is promised to contain exactly one 1 in each of the first k blocks; the position of the 1 within each block B_i encodes a bit z_i. Now consider a subset S of bits queried by an algorithm for CGT, and let $S_i = S \cap B_i$. We may assume that S is a subset of the first $2k$ bits, as the last $n-2k$ bits are promised to be 0. Now observe that by choosing each S_i appropriately, we can make three different kinds of query: $S_i = \{2i - 1\}$ corresponds to "does $z_i = 0$?", $S_i = \{2i\}$ corresponds to "does $z_i = 1$?", and $S_i = \{\}$ corresponds to excluding z_i from the query (the remaining query $S_i = \{2i - 1, 2i\}$ always returns 1 and is hence uninteresting). The overall query $S = \bigcup_i S_i$ is the OR of all of the individual queries. Thus a CGT query corresponds to a subset S of the bits of z and a claimed assignment y to these bits; the response is 1 if any of the bits of y are equal to z. To convert this into an instance of search with wildcards on k bits, simply observe that inverting the response to such a query is equivalent to performing a query (S, y) to $\bar{z}$ where the reply is 1 if $\bar{z}_S = y$. Thus an algorithm for CGT can be used to learn $\bar{z}$ and hence z.

2 Search with wildcards

We now show that we can indeed solve the search with wildcards problem efficiently, proving the upper bound part of Theorem 1 (for the lower bound, see Section 5). Consider an instance of search with wildcards of size n. Let $x \in \{0, 1\}^n$ and $k \in [n]$.

Our proof uses the following state distinguishability result (which we prove in Section 3).

Lemma 3. *Fix $n \geq 1$ and, for any $0 \leq k \leq n$, set*

$$|\psi_x^k\rangle := \frac{1}{\binom{n}{k}^{1/2}} \sum_{S \subseteq [n], |S| = k} |S\rangle|x_S\rangle,$$

where $|x_S\rangle := \bigotimes_{i \in S} |x_i\rangle$. Then, for any $k = n - O(\sqrt{n})$, there is a quantum measurement

(POVM) which, on input $|\psi_x^k\rangle$, outputs $\widetilde{x}$ such that the expected Hamming distance $d(x, \widetilde{x})$ is $O(1)$.

In words, Lemma 3 says that, given a superposition over k-subsets of the bits of x with $k = n - O(\sqrt{n})$, we can output a bit-string that is likely to be very close to x itself. This is in sharp contrast to the analogous situation classically; given any $n - O(\sqrt{n})$ bits of x, determining the remaining $O(\sqrt{n})$ bits succeeds only with exponentially small probability. Roughly speaking, our algorithm for search with wildcards will repeatedly use Lemma 3 to learn $O(\sqrt{n})$ bits of x at a time, fixing the incorrect bits after each measurement.

Consider an instance of search with wildcards of size n. Let $x \in \{0,1\}^n$ and $k \in [n]$. Recall that we denote

$$|\psi_x^k\rangle = \sum_{S:S\subseteq[n],|S|=k} |S\rangle|x_S\rangle,$$

where we write $|x_S\rangle := \otimes_{i\in S}|x_i\rangle$. Let $M_{n,k}$ be a measurement (POVM) for distinguishing the states $|\psi_x^k\rangle$, and assume that $M_{n,k}$ satisfies the following property: for $k \geq n - \sqrt{n}$, and all x, the expected Hamming distance of the outcome $\widetilde{x}$ from x is upper bounded by a constant. By Lemma 3, such a measurement $M_{n,k}$ indeed exists. We can express $M_{n,k}$ as a two-step process, with the first step being a unitary transformation $U_{n,k}$ that maps $|\psi_x^k\rangle$ to a state in $\mathcal{H}_o \otimes \mathcal{H}_g$ (where $\mathcal{H}_o$ is the output register and $\mathcal{H}_g$ is the rest of the state) and the second step being the measurement of $\mathcal{H}_o$ (with the measurement result interpreted as a guess $\widetilde{x}$ for the hidden bit-string x).

We define a sequence of numbers $n_0, \ldots, n_l$, with $n_l = n$ and $n_{i-1} = \lceil n_i - \sqrt{n_i}\rceil$. Our algorithm consists of Stages 0, 1, ..., l.

Stage 0. Generate $|\psi_x^{n_0}\rangle$ by first creating $\sum_{S:S\subseteq[n],|S|=n_0}|S\rangle$ and then querying each $x_i, i \in S$.

Stage s ($s > 0$). Stage s receives $|\psi_x^{n_{s-1}}\rangle$ as the input and outputs $|\psi_n^{n_s}\rangle$. It consists of the following steps:

1. With no queries, transform $|\psi_x^{n_{s-1}}\rangle$ to

$$\sum_{S':S'\subseteq[n],|S'|=n_s} |S'\rangle \sum_{S:S\subseteq S',|S|=n_{s-1}} |S\rangle|x_S\rangle = \sum_{S:S\subseteq[n],|S|=n_s} |S\rangle|\psi_{x_S}^{n_{s-1}}\rangle.$$

   This can be achieved by attaching an ancilla register in the state $|0\rangle$ and performing the map $|S\rangle|0\rangle \mapsto |S\rangle\left(\sum_{S':S\subseteq S',|S'|=n_s}|S'\rangle\right)$, which does not require any queries.

2. Apply $U_{n_s,n_{s-1}}$ on the register holding $|\psi_{x_S}^{n_{s-1}}\rangle$. Use a subset query to verify whether $\widetilde{x_S}$ in the $\mathcal{H}_o$ register is indeed equal to x_S. Measure the outcome of the subset query.

3. If the subset query answers that $\widetilde{x_S} = x_S$, we have a state

$$\sum_{S:S\subseteq[n],|S|=n_s} |S\rangle|x_S\rangle|\varphi_S\rangle$$

   where $|\varphi_S\rangle$ is a state in the $\mathcal{H}_g$ register. Apply the transformation $|S\rangle|\varphi_S\rangle \mapsto |S\rangle|0\rangle$ (which requires no queries) and discard the $\mathcal{H}_g$ register.

4. If the subset query answers that $\widetilde{x_S} \neq x_S$, repeat the following sequence of transformations:
   (a) Use a binary search with $\lceil \log n_s \rceil$ subset queries (performed coherently, without measurements) to find one i for which $(\widetilde{x_S})_i \neq (x_S)_i$. If the algorithm succeeds, change $(\widetilde{x_S})_i$ to the opposite value.
   (b) Use a subset query to verify whether $\widetilde{x_S}$ in the output register is now equal to x_S. Measure the outcome of the subset query.
   (c) If the subset query answers that $\widetilde{x_S} \neq x_S$, return to step 4a.
   (d) If the subset query answers that $\widetilde{x_S} = x_S$, we have a state

$$\sum_{S:S\subseteq[n],|S|=n_s} |S\rangle|x_S\rangle|\varphi_S\rangle$$

   where $|\varphi_S\rangle$ is some "garbage" state consisting of the contents of $\mathcal{H}_g$ after $U_{n_s,n_{s-1}}$ and leftover information from the subset queries in step 4a. Apply the transformation $|S\rangle|\varphi_S\rangle \mapsto |S\rangle|0\rangle$ (which requires no queries) and discard the register holding the $|0\rangle$ state.

The expected number of queries for Stage s ($s > 0$) is 1 for step 2 and $O(D \log n)$ for step 4, where D is the expected number of errors in the answer $\widetilde{x_S}$. Since $D = O(1)$ by Lemma 3, the expected number of queries is $O(\log n)$.

For the number of stages, we can choose $l = O(\sqrt{n})$ so that $n_0 = O(1)$. To see this, consider the result n' of applying the map $z \mapsto \lceil z - \sqrt{z} \rceil$ some number of times, starting with n. Then any z produced during this process satisfies $\lceil z - \sqrt{z} \rceil \leq z - \sqrt{n'} + 1$, and in particular the map needs to be repeated at most $\sqrt{n/2} + 1$ times to produce $n' \leq n/2$ (for large enough n). Repeating this process, the number of iterations required to produce a value which is $O(1)$ is at most $(\sqrt{n/2} + 1) + (\sqrt{n/4} + 1) + \cdots = O(\sqrt{n})$.

Thus the algorithm uses $n_0 = O(1)$ queries in Stage 0 and expected $O(\log n)$ queries in each of the next $O(\sqrt{n})$ stages. Hence, the expected total number of queries is $O(\sqrt{n} \log n)$.

3 The state discrimination problem

Our next task is to prove Lemma 3, i.e. to show that, given the state

$$|\psi_x^k\rangle := \frac{1}{\binom{n}{k}^{1/2}} \sum_{S\subseteq[n],|S|=k} |S\rangle|x_S\rangle,$$

for any $k = n - O(\sqrt{n})$, we can output $\widetilde{x}$ such that the expected Hamming distance between $\widetilde{x}$ and x is constant. We will achieve this using the pretty good measurement [3, 4] (PGM), which is also known as the square root measurement [19] and is defined as follows. Given a set $\{|\phi_i\rangle\}$ of pure states, set $\rho = \sum_i |\phi_i\rangle\langle\phi_i|$. Then the measurement vector corresponding to state $|\phi_i\rangle$ is $|\mu_i\rangle := \rho^{-1/2}|\phi_i\rangle$, the inverse being taken on the support of ρ. This is a valid POVM because

$$\sum_i |\mu_i\rangle\langle\mu_i| = \sum_x \rho^{-1/2}|\phi_i\rangle\langle\phi_i|\rho^{-1/2} = \rho^{-1/2}\left(\sum_i |\phi_i\rangle\langle\phi_i|\right)\rho^{-1/2} = I.$$

The probability that the PGM outputs j on input $|\phi_i\rangle$ is precisely $|\sqrt{G}_{ij}|^2$, where G is the Gram matrix of the states $\{|\phi_i\rangle\}$, $G_{ij} = \langle\phi_i|\phi_j\rangle$. In our case, we have

$$G_{xy} = \langle\psi_x^k|\psi_y^k\rangle = \frac{1}{\binom{n}{k}} \sum_{S\subseteq[n],|S|=k} [x_S = y_S] = \frac{\binom{n-d(x,y)}{k}}{\binom{n}{k}}.$$

As G_{xy} depends only on $x \oplus y$, G is diagonalised by the Fourier transform over $\mathbb{Z}_2^n$ (see for example [20, Theorem 2.3.6]). Eigenvalues $\lambda(s)$ of G, indexed by bit-strings $s \in \{0,1\}^n$, are thus given by the Fourier transform of the function $f(x) = G_{x0} = \frac{\binom{n-|x|}{k}}{\binom{n}{k}}$. Indeed, we have

$$\lambda(s) = \sum_{x\in\{0,1\}^n} (-1)^{s\cdot x} f(x) = \frac{1}{\binom{n}{k}} \sum_{x\in\{0,1\}^n} (-1)^{s\cdot x} \binom{n-|x|}{k} = 2^{n-k} \frac{\binom{n-|s|}{n-k}}{\binom{n}{k}}, \qquad (1)$$

where the final equality is an identity of Delsarte [21, Eq. (48)].

As $\sqrt{G}_{xy}$ also depends only on $x \oplus y$, the expected Hamming distance of the output y from the input x does not depend on x and is equal to

$$D_k := \sum_{y\in\{0,1\}^n} d(x,y)(\sqrt{G}_{xy})^2 = \sum_{y\in\{0,1\}^n} |y|(\sqrt{G}_{0y})^2.$$

We now proceed to upper bound this quantity using Fourier duality. Observe that D_k can be viewed as the inner product between the functions $f(x) = |x|$ and $g(x) = (\sqrt{G}_{0x})^2$. By Plancherel's theorem we have

$$\sum_{x\in\{0,1\}^n} f(x)g(x) = 2^n \sum_{s\in\{0,1\}^n} \hat{f}(s)\hat{g}(s),$$

where for any function f we define $\hat{f}(s) = \frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{s\cdot x} f(x)$. One can easily calculate that

$$\hat{f}(s) = \begin{cases} \frac{n}{2} & \text{if } s = 0^n \\ -\frac{1}{2} & \text{if } |s| = 1 \\ 0 & \text{otherwise.} \end{cases}$$

On the other hand, we can compute the Fourier spectrum of g as follows. As the Fourier transform turns multiplication into convolution, we have

$$\hat{g}(s) = \widehat{\sqrt{g}\sqrt{g}}(s) = \sum_{t\in\{0,1\}^n} \widehat{\sqrt{g}}(t)\widehat{\sqrt{g}}(s+t).$$

We can therefore determine the Fourier spectrum of g directly from that of the function $\sqrt{g}(x) = \sqrt{G}_{0x}$. We have already computed this Fourier transform; up to normalisation, it is just the function giving the eigenvalues of $\sqrt{G}$, or in other words the function $\sqrt{\lambda}(s)$. We thus obtain

$$\begin{aligned} \hat{g}(s) &= \frac{2^{-n-k}}{\binom{n}{k}} \sum_{t\in\{0,1\}^n} \binom{n-|t|}{n-k}^{1/2} \binom{n-d(s,t)}{n-k}^{1/2} \\ &= \frac{2^{-n-k}}{\binom{n}{k}} \sum_{t,u=0}^{n} |\{y : |y| = t, d(s,y) = u\}| \binom{n-t}{n-k}^{1/2} \binom{n-u}{n-k}^{1/2}. \end{aligned}$$

This is a fairly complicated expression, but as $\hat{f}(s) = 0$ when $|s| > 1$, we only need to calculate a few special cases. In particular, we have $\hat{g}(0^n) = 1/2^n$ and

$$\begin{aligned}
\hat{g}(e_i) &= \frac{2^{-n-k}}{\binom{n}{k}} \sum_{t=0}^{n} \binom{n-t}{n-k}^{1/2} \left(\binom{n-1}{t-1} \binom{n-t+1}{n-k}^{1/2} + \binom{n-1}{t} \binom{n-t-1}{n-k}^{1/2} \right) \\
&= 2^{-n-k} \sum_{t=0}^{n} \binom{k}{t} \left(\frac{t}{n} \left(\frac{n-t+1}{k-t+1} \right)^{1/2} + \left(1 - \frac{t}{n}\right) \left(\frac{k-t}{n-t} \right)^{1/2} \right) \\
&=: 2^{-n-k} \sum_{t=0}^{n} \binom{k}{t} T_t
\end{aligned}$$

for bit-strings e_i of Hamming weight 1. Thus $2^n \hat{g}(e_i)$ is equal to 1 when $k = n$ and will be close to 1 when k is close to n. Indeed, set $k = n - c\sqrt{n}$ and consider terms T_t in this sum such that $t = n/2 + a\sqrt{n}$, for $a \in \mathbb{R}$. Then, using the lower bound $\sqrt{x} \geq \frac{3}{2}x - \frac{1}{2}x^2$, which is valid for $x \geq 0$, we have

$$\begin{aligned}
T_t &= \left(\frac{1}{2} + \frac{a}{\sqrt{n}}\right) \left(1 + \frac{c}{\sqrt{n}/2 - (a+c) + 1/\sqrt{n}}\right)^{1/2} + \left(\frac{1}{2} - \frac{a}{\sqrt{n}}\right) \left(1 - \frac{c}{\sqrt{n}/2 - a}\right)^{1/2} \\
&\geq \left(\frac{1}{2} + \frac{a}{\sqrt{n}}\right) \left(1 + \frac{c}{\sqrt{n}/2 - a}\right)^{1/2} + \left(\frac{1}{2} - \frac{a}{\sqrt{n}}\right) \left(1 - \frac{c}{\sqrt{n}/2 - a}\right)^{1/2} \\
&\geq 1 - \frac{1}{2} \left(\frac{c}{\sqrt{n}/2 - a}\right)^2 + \frac{ac}{\sqrt{n}(\sqrt{n}/2 - a)} \\
&= 1 - O(1/n)
\end{aligned}$$

for constant a and c. We thus have $2^n \hat{g}(e_i) \geq 1 - O(1/n)$. Computing the inner product $2^n \sum_{s \in \{0,1\}^n} \hat{f}(s)\hat{g}(s)$, we get

$$D_k = \frac{n}{2} (1 - \hat{g}(e_i)) = O(1)$$

as desired. In Appendix 1, we continue the analysis of the state discrimination problem by giving quite tight upper and lower bounds on the probability of identifying x exactly.

4 Algorithms for combinatorial group testing

We now consider the related problem of combinatorial group testing, beginning by considering the very special case of CGT where $k = 1$. Classically, this problem can be solved with certainty using binary search in $\lceil \log_2 n \rceil$ queries, which is asymptotically tight by an information-theoretic argument.

Lemma 4. *If $k = 1$, CGT can be solved exactly using one quantum query.*

Proof. The result follows from observing that, in order to learn x, it suffices to compute the function $x \cdot s$ for arbitrary $s \in \{0,1\}^n$ (this is the same observation that underpins the quantum oracle interrogation algorithm of van Dam [12]). In the CGT problem, we have access to an oracle which computes $f(s) = \bigvee_i x_i s_i$ for arbitrary $s \in \{0,1\}^n$. But if $|x| \leq 1$, then for any s, $\bigvee_i x_i s_i = x \cdot s$.

Formally, the quantum algorithm proceeds as follows.

1. Create the state $\frac{1}{\sqrt{2^{n+1}}}\sum_{s\in\{0,1\}^n}|s\rangle(|0\rangle-|1\rangle)$.
2. Apply the oracle to create the state

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{s\in\{0,1\}^n}(-1)^{\bigvee_i s_i x_i}|s\rangle(|0\rangle-|1\rangle)=\frac{1}{\sqrt{2^{n+1}}}\sum_{s\in\{0,1\}^n}(-1)^{s\cdot x}|s\rangle(|0\rangle-|1\rangle)$$

3. Apply Hadamard gates to each qubit of the first register and measure to obtain x.

□

Call the above algorithm the $k=1$ algorithm. We can extend this idea to obtain a simple quantum algorithm for CGT which achieves significantly better query complexity than is possible classically, by not depending on n. First assume that $|x|=k$, and let I be the set of 1-indices of x which are currently known (initially, $I=\emptyset$). The algorithm is based on the following subroutine.

1. Construct a subset $S\subseteq[n]\backslash I$ by including each $i\in[n]\backslash I$ with independent probability $1/k$. Write S_j for the j'th element of S.
2. Create the state $\left(\frac{1}{\sqrt{2^{|S|+1}}}\sum_{t\in\{0,1\}^{|S|}}|t\rangle\right)(|0\rangle-|1\rangle)$.
3. Apply the oracle to create the state

$$\frac{1}{\sqrt{2^{|S|+1}}}\sum_{t\in\{0,1\}^{|S|}}(-1)^{\bigvee_{i=1}^{|S|}t_i x_{S_i}}|t\rangle(|0\rangle-|1\rangle);$$

henceforth ignore the second register.

4. Apply Hadamard gates to each qubit of the first register to produce the state

$$\frac{1}{2^{|S|}}\sum_{y\in\{0,1\}^{|S|}}\left(\sum_{t\in\{0,1\}^{|S|}}(-1)^{\bigvee_{i=1}^{|S|}t_i x_{S_i}+\sum_{i=1}^{|S|}t_i y_i}\right)|y\rangle.$$

5. Measure to obtain $y\in\{0,1\}^{|S|}$.
6. For each i such that $y_i=1$, add S_i to I. Reduce k by $|y|$.

Observe that, for all i such that $x_{S_i}=0$, the state produced in Step 4 has zero amplitude on all y such that $y_i=1$. Thus, for each index S_i added to I, $x_{S_i}=1$. On the other hand, the probability that the outcome $y=0^{|S|}$ is obtained is exactly $(1-2^{1-|x_S|})^2$, so the algorithm finds at least one 1-index with probability $1-(1-2^{1-|x_S|})^2$. In particular, if S contains exactly one 1-index i of x, which will occur with probability at least $(1-1/k)^{k-1}\geq 1/e$, we are guaranteed to learn i. Repeating this subroutine, the expected overall number of queries used in order to learn x completely is $O(k)$.

If we only know the upper bound that $|x|\leq k$, we can simply use the above subroutine with guesses $k'=2^i$, $i=0,\dots,\lceil\log_2 k\rceil$. For at least one of these choices (call this k'') such that $k''\leq|x|\leq 2k''$, the probability that the subset S which we pick randomly contains exactly one 1-index is lower bounded by a constant. For each k' we apply the subroutine $O(k')$ times such that, when $k' = k''$, the probability that we do not learn all the 1-indices of x is upper bounded by 1/3. After this procedure, we have used $\sum_{i=0}^{\lceil \log_2 k \rceil} O(2^i) = O(k)$ queries and have learned all of the 1-indices of x, except with probability at most 1/3. We can convert this into a Las Vegas algorithm, i.e. an algorithm that always succeeds eventually, by checking whether we have found all 1-indices of x (by querying the complement of the 1-indices found so far), and repeating if necessary.

5 An almost matching lower bound

We finally prove that our results for the search with wildcards and combinatorial group testing problems are almost optimal. We will use the following very general "strong weighted adversary" bound formulated by Zhang [22] (for the statement given here, see [15, 23]).

Theorem 5. *Let $f : S \to T$ be a function and let Q be a finite set of possible query strings. Let $x \in S$ be an initially unknown input which is accessed via an oracle O_x performing the map $O_x|q\rangle|z\rangle = |q\rangle|z \oplus \zeta(x, q)\rangle$, where $q \in Q$, $z \in \{0, 1\}$, and $\zeta : S \times Q \to \{0, 1\}$ is a function specifying the response to oracle queries. Also let w, w' be weight schemes such that:*

- *Each pair $(x, y) \in S \times S$ is assigned a non-negative weight $w(x, y) = w(y, x)$ such that $w(x, y) = 0$ whenever $f(x) = f(y)$;*
- *Each triple $(x, y, q) \in S \times S \times Q$ is assigned a non-negative weight $w'(x, y, q)$ such that $w'(x, y, q) = 0$ for all x, y, q such that $\zeta(x, q) = \zeta(y, q)$ or $f(x) = f(y)$, and $w'(x, y, q)w'(y, x, q) \geq w(x, y)^2$ for all x, y, q such that $\zeta(x, q) \neq \zeta(y, q)$ and $f(x) \neq f(y)$.*

For all $x \in S$ and $q \in Q$, set $wt(x) = \sum_y w(x, y)$ and $v(x, q) = \sum_y w'(x, y, q)$. Then any quantum query algorithm that computes $f(x)$ with success probability at least 2/3 on every input x must make

$$\Omega\left(\min_{\substack{x,y,q;\, w(x,y)>0,\\ \zeta(x,q)\neq\zeta(y,q)}} \sqrt{\frac{wt(x)wt(y)}{v(x,q)v(y,q)}}\right)$$

queries to the oracle O_x.

Lemma 6. *Any quantum algorithm which solves search with wildcards on n bits with worst-case success probability 2/3 must make $\Omega(\sqrt{n})$ oracle queries.*

Proof. In the search with wildcards problem the input is a string $x \in \{0, 1\}^n$, queries $q = (S, t)$ are specified by $S \subseteq [n]$, $t \in \{0, 1\}^{|S|}$, and $\zeta(x, q) = 1$ if and only if $x_S = t$. We define the following weight scheme: $w(x, y) = 1$ if $d(x, y) = 1$, and $w(x, y) = 0$ otherwise; $w'(x, y, q) = w'(y, x, q) = 1$ if $d(x, y) = 1$ and $\zeta(x, q) \neq \zeta(y, q)$, and $w'(x, y, q) = w'(y, x, q) = 0$ otherwise. For any $x \in \{0, 1\}^n$, $\mathrm{wt}(x) = n$. On the other hand,

$$v(x, q) = |\{y : d(x, y) = 1, \zeta(x, q) \neq \zeta(y, q)\}| = \begin{cases} |S| & [\zeta(x, q) = 1] \\ 1 & [\zeta(x, q) = 0,\, d(x_S, t) = 1] \\ 0 & [\text{otherwise}] \end{cases}.$$

Hence

$$\min_{\substack{x,y,q;\, w(x,y)>0\\ \zeta(x,q)\neq\zeta(y,q)}} \sqrt{\frac{\mathrm{wt}(x)\mathrm{wt}(y)}{v(x,q)v(y,q)}} = \sqrt{n}$$

and the claim follows from Theorem 5. □

Via the reduction from search with wildcards to CGT, Lemma 6 implies that CGT requires $\Omega(\sqrt{k})$ quantum queries, completing the proof of Theorem 2. A simple direct proof of this is to consider the special case where the input string x is of length $k+1$, and is promised to have Hamming weight k. Then the problem reduces to finding the single zero in x, and any query to a subset of size 2 or more always returns 1. This is therefore the same problem as standard unstructured search for a single marked element in a database of size $k+1$, which is known to require $\Omega(\sqrt{k})$ quantum queries [24].

6 Outlook

The major open question left by our work is to fully resolve the quantum query complexity of CGT. A previous version of this paper incorrectly claimed a $O(\sqrt{k}\,\mathrm{polylog}(k))$ algorithm for this problem; it is a very interesting open problem to determine its true complexity.

An alternative way of considering the CGT problem is as a restricted case of the problem of learning juntas via membership queries [25, 26]. A k-junta is a boolean function that depends only on at most k input bits. The general problem of learning juntas is defined as follows. Given oracle access to a function $f : \{0,1\}^n \rightarrow \{0,1\}$, and the promise that f is a k-junta, output a representation of f (e.g. its truth table). It is easy to see that CGT is the special case of this problem where f is restricted to be the OR of at most k of the input bits; our algorithm therefore allows this restricted problem to be solved using $O(k)$ queries. The same algorithm also works if f is promised to be an AND function (i.e. $f(x) = \bigwedge_{i\in S} x_i$, for some S such that $|S| = k$), because in this case querying $f(\bar{x})$ and negating the output simulates a query to a function f' such that $f'(x) = \bigvee_{i\in S} x_i$. It would be interesting to determine whether efficient quantum algorithms could be found for other restricted cases of the junta learning problem.

A related question is *testing* juntas. In this problem we are given a function $f : \{0,1\}^n \rightarrow \{0,1\}$ such that f either is a k-junta, or differs from any k-junta on at least $\epsilon 2^n$ inputs, and must determine which is the case. Classically, this problem can be solved using $O(k/\epsilon + k\log k)$ queries [27], while there is an $\Omega(k)$ lower bound on the number of queries required [28]. In the quantum case, Atıcı and Servedio have given an $O(k/\epsilon)$-query algorithm [26]. It has recently been observed that there are connections between the junta testing problem and CGT [29]. It would be very interesting if our results could be used or generalised to give an $O(\sqrt{k}\,\mathrm{polylog}(k))$ quantum algorithm for testing juntas.

Acknowledgements

AM was supported by an EPSRC Postdoctoral Research Fellowship and would like to thank Raphaël Clifford for helpful comments on a previous version, and David Gosset, Robin Kothari and Rajat Mittal for helpful discussions. AA was suppported by the European Commission under the projects QCS (Grant No. 255961), MQC (Grant No. 320731) and QALGO (Grant No. 600700). We would like to thank Aram Harrow for suggesting a collaboration between the authors and helpful discussions, and two anonymous referees for comments which have improved the paper. Special thanks to Han-Hsuan Lin for spotting a crucial error in a previous version.

Appendix A

Further analysis of the state discrimination problem

In this appendix, we carry out some further analysis of the problem of discriminating the states $|\psi_x^k\rangle$ discussed in Section 3. We have the bound from [30] that

$$(\sqrt{G}_{xx})^2 \geq \frac{1}{\sum_{y\in\{0,1\}^n} |\langle\psi_x^k|\psi_y^k\rangle|^2}, \tag{A.1}$$

which allows us to prove the following lower bound on the probability that the PGM outputs x exactly.

Lemma 7. *Set $k = n - a\sqrt{n}$ for some $0 \leq a \leq 1$. Then $(\sqrt{G}_{xx})^2 \geq 1 - 2a^2 - O(1/\sqrt{n})$.*

Proof. By (A.1) we have

$$(\sqrt{G}_{xx})^2 \geq \frac{\binom{n}{k}^2}{\sum_{d=0}^n \binom{n}{d}\binom{d}{k}^2} = \frac{\binom{n}{k}}{\sum_{d=0}^n \binom{d}{k}\binom{n-k}{d-k}}.$$

We now upper bound the reciprocal of this quantity, setting $g = n-k$, $i = n-d$ to obtain

$$\begin{aligned}
\frac{1}{\binom{n}{g}}\sum_{i=0}^{g}\binom{n-g+i}{i}\binom{g}{i} &= \frac{1}{\binom{n}{g}}\sum_{i=0}^{g}\binom{n-g}{i}\binom{g}{i}\frac{(n-g+i)\dots(n-g+1)}{(n-g)\dots(n-g-i+1)}\\
&= \frac{1}{\binom{n}{g}}\sum_{i=0}^{g}\binom{n-g}{i}\binom{g}{i}\left(1+\frac{i}{n-g}\right)\dots\left(1+\frac{i}{n-g-i+1}\right)\\
&\leq \frac{1}{\binom{n}{g}}\sum_{i=0}^{g}\binom{n-g}{i}\binom{g}{i}\left(1+\frac{g}{n-2g+1}\right)^{g+1}\\
&\leq e^{g(g+1)/(n-2g+1)}\\
&\leq 1 + 2a^2 + O(1/\sqrt{n}).
\end{aligned}$$

□

We also record here an exact expression for the probability of getting outcome y on input x. Let $K_k^n(x)$ be the k'th Krawtchouk polynomial [21], defined by

$$K_k^n(x) = \sum_{i=0}^{k}(-1)^i\binom{x}{i}\binom{n-x}{k-i}.$$

Lemma 8.

$$(\sqrt{G}_{xy})^2 = 2^{-(n+k)}\binom{n}{d(x,y)}^{-2}\left(\sum_{z=0}^{n}K_{d(x,y)}^n(z)\binom{n}{z}^{1/2}\binom{k}{z}^{1/2}\right)^2. \tag{A.2}$$

Proof. Essentially immediate from the discussion in Section 3; the entries of $\sqrt{G}$ can be calculated using

$$\begin{aligned}\sqrt{G}_{xy} &= \frac{1}{2^n}\sum_{s\in\{0,1\}^n}(-1)^{(x\oplus y)\cdot s}\lambda(s)^{1/2} = \frac{1}{2^{(n+k)/2}\binom{n}{k}^{1/2}}\sum_{s\in\{0,1\}^n}(-1)^{(x\oplus y)\cdot s}\binom{n-|s|}{n-k}^{1/2}\\ &= \frac{1}{2^{(n+k)/2}\binom{n}{k}^{1/2}}\sum_{z=0}^{n}\binom{n-z}{n-k}^{1/2}\sum_{s\in\{0,1\}^n,|s|=z}(-1)^{(x\oplus y)\cdot s}\\ &= \frac{1}{2^{(n+k)/2}\binom{n}{k}^{1/2}}\sum_{z=0}^{n}\binom{n-z}{n-k}^{1/2}K_z^n(d(x,y)),\end{aligned}$$

where $\lambda(s)$ are the eigenvalues of G (see eqn. (1)). Lemma 8 then follows using well-known identities for binomial coefficients and Krawtchouk polynomials [21]. □

We finally turn to putting upper bounds on how well x can be identified given a state of the form $|\psi_x^k\rangle$. We first observe that there is no loss of generality in putting upper bounds on the success probability of the PGM, as it is in fact the optimal measurement for identifying x (in terms of minimising the average probability of error). This follows from a result of Eldar and Forney [19] which shows that the PGM minimises the probability of error of state discrimination for states which are geometrically uniform, i.e. generated by applying an abelian group to an initial state $|\phi\rangle$. This holds for our states, as they can be thought of as being generated by applying the unitary U_y defined by $U_y|S\rangle|x\rangle = |S\rangle|x+y_S\rangle$ to the initial state $\sum_{S\subseteq[n],|S|=k}|S\rangle|0\rangle$. The set $\{U_y\}$, $y\in\{0,1\}^n$, clearly forms an abelian group. As a more concise proof, optimality of the PGM follows directly from the diagonal entries of $\sqrt{G}$ being equal [3].

Lemma 9. *Set $k = n - a\sqrt{n}$ for some $a\geq 0$. Then*

$$(\sqrt{G}_{xx})^2 \leq 4e^{-a^2/32}.$$

Proof. We have

$$\sqrt{G}_{xx} = 2^{-(n+k)/2}\sum_{z=0}^{n}\binom{n}{z}^{1/2}\binom{k}{z}^{1/2}.$$

Now split the sum into two parts to obtain

$$\sqrt{G}_{xx} = 2^{-(n+k)/2}\sum_{z\leq k/2+a\sqrt{k}/4}\binom{k}{z}^{1/2}\binom{n}{z}^{1/2} + 2^{-(n+k)/2}\sum_{z>k/2+a\sqrt{k}/4}\binom{k}{z}^{1/2}\binom{n}{z}^{1/2}$$

$$\leq \left(\frac{1}{2^k}\sum_{z\leq\frac{k}{2}+\frac{a\sqrt{k}}{4}}\binom{k}{z}\right)^{1/2}\left(\frac{1}{2^n}\sum_{z>\frac{k}{2}+\frac{a\sqrt{k}}{4}}\binom{n}{z}\right)^{1/2} + \left(\frac{1}{2^k}\sum_{\frac{k}{2}+\frac{a\sqrt{k}}{4}}\binom{k}{z}\right)^{1/2}\left(\frac{1}{2^n}\sum_{z>\frac{k}{2}+\frac{a\sqrt{k}}{4}}\binom{n}{z}\right)^{1/2}$$

by Cauchy-Schwarz. We now use the Chernoff bound that

$$\frac{1}{2^n}\sum_{z\geq n/2+b\sqrt{n}}\binom{n}{z}\leq e^{-b^2/2}$$

for any $b > 0$, which implies

$$\frac{1}{2^n} \sum_{z \le k/2 + a\sqrt{k}/4} \binom{n}{z} \le e^{-a^2/32}, \quad \frac{1}{2^k} \sum_{z > k/2 + a\sqrt{k}/4} \binom{k}{z} \le e^{-a^2/32},$$

noting that $k/2 + a\sqrt{k}/4 \le n/2 - a\sqrt{n}/4$ by assumption. The claimed upper bound follows. $\square$

References

1. E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation in determining parity. *Phys. Rev. Lett.*, 81:5442–5444, 1998. `quant-ph/9802045`.
2. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. `quant-ph/9802049`.
3. V. P. Belavkin. Optimal multiple quantum statistical hypothesis testing. *Stochastics*, 1:315–345, 1975.
4. P. Hausladen and W. Wootters. A "pretty good" measurement for distinguishing quantum states. *J. Mod. Opt.*, 41(12):2385–2390, 1994.
5. R. Dorfman. The detection of defective members of large populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.
6. D. Du and F. Hwang. *Combinatorial Group Testing and Its Applications.* World Scientific, 2000.
7. E. Porat and A. Rothschild. Explicit non-adaptive combinatorial group testing schemes. In *Proc. 35th International Conference on Automata, Languages and Programming (ICALP'08)*, pages 748–759, 2008. `arXiv:0712.3876`.
8. M. Farach, S. Kannan, E. Knill, and S. Muthukrishnan. Group testing problems with sequences in experimental molecular biology. In *Proc. Compression and Complexity of Sequences 1997*, pages 357–367, 1997.
9. G. Cormode and S. Muthukhrishan. What's hot and what's not: tracking most frequent items dynamically. *ACM Trans. Database Syst.*, 30(1):249–278, 2005.
10. G. Cormode and S. Muthukhrishan. Combinatorial algorithms for compressed sensing. In *Proc. 13th Colloquium on Structural Information and Communication Complexity (SIROCCO'06)*, pages 280–294, 2006.
11. R. Clifford, K. Efremenko, E. Porat, and A. Rothschild. Pattern matching with don't cares and few errors. *J. Comput. Syst. Sci.*, 76(2):115–124, 2010.
12. W. van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proc. 39th Annual Symp. Foundations of Computer Science*, pages 362–367. IEEE, 1998. `quant-ph/9805006`.
13. E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. How many functions can be distinguished with k quantum queries?, 1999. `quant-ph/9901012`.
14. K. Iwama, H. Nishimura, R. Raymond, and J. Teruyama. Quantum counterfeit coin problems. In *Proc. 21st International Symposium on Algorithms and Computation (ISAAC 2010)*, pages 85–96, 2010. `arXiv:1009.0416`.
15. R. Cleve, K. Iwama, F. Le Gall, H. Nishimura, S. Tani, J. Teruyama, and S. Yamashita. Reconstructing strings from substrings with quantum queries. In *Proc. 13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT'12)*, pages 388–397, 2012. `arXiv:1204.4691`.
16. A. Ambainis, K. Iwama, A. Kawachi, H. Masuda, R. Putra, and S. Yamashita. Quantum identification of Boolean oracles. In *Proc. STACS 2004*, pages 93–104. Springer, 2004. `quant-ph/0403056`.
17. A. Ambainis, K. Iwama, A. Kawachi, R. Raymond, and S. Yamashita. Improved algorithms for quantum identification of boolean oracles. *Theor. Comput. Sci.*, 378:41–53, 2007. `quant-ph/0411204`.
18. A. Atıcı and R. Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5):355–386, 2005. `quant-ph/0411140`.

19. Y. C. Eldar and G. D. Forney, Jr. On quantum detection and the square-root measurement. *IEEE Trans. Inform. Theory*, 47(3):858–872, 2001. `quant-ph/0005132`.
20. T. Ceccherini-Silberstein, F. Scarabotti, and F. Tolli. *Harmonic Analysis on Finite Groups*. Cambridge University Press, 2008.
21. I. Krasikov and S. Litsyn. Survey of binary Krawtchouk polynomials. In *Codes and Association Schemes*, volume 56 of *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, pages 199–212. American Mathematical Society, 1999.
22. S. Zhang. On the power of Ambainis lower bounds. *Theoretical Computer Science*, 339(2–3):241–256, 2005. `quant-ph/0311060`.
23. R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2:1–18, 2006. `quant-ph/0409116`.
24. C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. `quant-ph/9701001`.
25. E. Mossel, R. O'Donnell, and R. Servedio. Learning functions of k relevant variables. *J. Comput. Syst. Sci.*, 69(3):421–434, 2004.
26. A. Atıcı and R. A. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6:323–348, 2007. `arXiv:0707.3479`.
27. E. Blais. Testing juntas nearly optimally. In *Proc. 41^{st} Annual ACM Symp. Theory of Computing*, pages 151–158, 2009.
28. H. Chockler and D. Gutfreund. A lower bound for testing juntas. *Inf. Proc. Lett.*, 90(6):301–305, 2004.
29. D. García Soriano. *Query-Efficient Computation in Property Testing and Learning Theory*. PhD thesis, University of Amsterdam, 2012.
30. A. Montanaro. On the distinguishability of random quantum states. *Comm. Math. Phys.*, 273(3):619–636, 2007. `quant-ph/0607011`.