PLOS | ONE

RESEARCH ARTICLE

Automatic identification of optimal marker genes for phenotypic and taxonomic groups of microorganisms

Elad Segev[1☯]*, Zohar Pasternak[2☯], Tom Ben Sasson[3☯], Edouard Jurkevitch[2], Mira Gonen[4]

1 Department of Mathematics, Holon Institute of Technology, Holon, Israel, **2** Department of Plant Pathology and Microbiology, Faculty of Agriculture, Food and Environment, The Hebrew University of Jerusalem, Rehovot, Israel, **3** Department of Mathematics and Computer Science, The Open University of Israel, Raanana, Israel, **4** Department of Computer Science, Ariel University, Ariel, Israel

☯ These authors contributed equally to this work.
* elad1segev@gmail.com

Abstract

Finding optimal markers for microorganisms important in the medical, agricultural, environmental or ecological fields is of great importance. Thousands of complete microbial genomes now available allow us, for the first time, to exhaustively identify marker proteins for groups of microbial organisms. In this work, we model the biological task as the well-known mathematical "hitting set" problem, solving it based on both greedy and randomized approximation algorithms. We identify unique markers for 17 phenotypic and taxonomic microbial groups, including proteins related to the nitrite reductase enzyme as markers for the non-anammox nitrifying bacteria group, and two transcription regulation proteins, *nusG* and *yhiF*, as markers for the Archaea and *Escherichia/Shigella* taxonomic groups, respectively. Additionally, we identify marker proteins for three subtypes of pathogenic *E. coli*, which previously had no known optimal markers. Practically, depending on the completeness of the database this algorithm can be used for identification of marker genes for any microbial group, these marker genes may be prime candidates for the understanding of the genetic basis of the group's phenotype or to help discover novel functions which are uniquely shared among a group of microbes. We show that our method is both theoretically and practically efficient, while establishing an upper bound on its time complexity and approximation ratio; thus, it promises to remain efficient and permit the identification of marker proteins that are specific to phenotypic or taxonomic groups, even as more and more bacterial genomes are being sequenced.

Check for updates

OPEN ACCESS

Citation: Segev E, Pasternak Z, Ben Sasson T, Jurkevitch E, Gonen M (2018) Automatic identification of optimal marker genes for phenotypic and taxonomic groups of microorganisms. PLoS ONE 13(5): e0195537. https://doi.org/10.1371/journal.pone.0195537

Editor: Christos A. Ouzounis, CPERI, GREECE

Received: November 1, 2017

Accepted: March 23, 2018

Published: May 2, 2018

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The authors received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Introduction

The first complete bacterial genome sequence was published in 1995 [1]. Since then, sequencing technology has developed rapidly, causing a dramatic reduction in the cost of sequencing, which made bacterial genome sequencing affordable to a great number of labs [2]. The

Ensembl database of whole genomes of bacteria has grown from about 9,000 to more than 40,000 in a few short years [3,4], and this number continues to increase exponentially each year [5]. This large number of bacterial genomes enabled us, for the first time, to identify marker genes for specific groups of microbial organisms based on the full complement of genes in each genome [6]. Currently, molecular typing of specific microbial groups is mostly done using either multi-locus sequence typing (MLST) or core genome MLST (cgMLST). In MLST, 5–16 (usually 7–8) housekeeping genes are selected as molecular markers and their sequence is compared between isolates [7]. However, the limited repertoire of highly conserved genes, as well as their sequence conservation, may sometime limit the discriminative power of this method, as evident in the typing of *Enterococcus faecium* [8]. The solution is usually to increase the number of genes, so in cgMLST, between 1500–3000 marker genes are used, which increases the discriminative power but forces any new isolate to be fully sequenced before it can be typed, thus requiring complex genomic analysis. A different way to improve MLST is by discarding the usage of housekeeping genes in favor of small groups of genes that are unique to specific taxonomic or phenotypic groups. This allows quick and affordable typing, using PCR instead of whole-genome sequencing, while retaining high discriminative power.

When using non-housekeeping marker genes for the MLST scheme, choosing the right ones is critical: non-representative or non-unique genes can lead to erroneous typing [8], or to an inefficient process due to requiring too many marker genes per group in order to verify the genomes' membership. Genetic markers are often selected ad hoc, using too few reference genomes and/or manual inspection of the results [9]. Therefore, an algorithm for finding optimal markers for specific groups of organisms is of great value to ecological and medical research [10,11]. Several software tools were developed for this purpose, but these are mostly limited to a single pathogenic organism [12], are computationally intensive [13], or can only identify marker genes for defined taxonomic groups [14,15]. Existing methods are not capable of creating novel typing schemes for any group of genomes (from one to thousands) by user choice, in a user-friendly manner while operating quickly and efficiently on any personal computer. We use an innovative approach for this problem, mapping it using the well-known Hitting Set (HS) mathematical problem. To increase the discriminative power of the approach, we use polypeptides instead of genes. Given a set of bacteria where each bacterium is represented by the set of the proteins it contains, a subset of bacteria to type will contain a minimum subset of specific proteins not found in the other bacteria. This minimum subset of proteins is thus present in every bacterium of the subset, marking the phenotype or taxon, while at least one protein is missing in the other bacteria, which do not have the same phenotype or do not belong to the taxon. The problem of typing a set of bacteria can thus be solved mathematically.

To solve this problem, we start from the set of all of the proteins that are present in all the bacteria in our database. These proteins will be marked as P_1, P_2 etc. in our case; a bacterium is defined as the set of proteins that it consists of.

For example, as seen in Fig 1A, bacterium no. 1 (denoted B_1) is defined as the set of P_1, P_2 and P_3, meaning that this bacterium contains only these three proteins, and bacterium B_2 contains only proteins P_1, P_2 and P_4. Given that B_1 and B_2 form a group of interest which we wish to identify, P_1 or P_2 can serve as marker proteins for this group of interest. Thus, the minimal set of proteins which identify this group can be either P_1 or P_2. If a third bacterium exists, which is not included in the group of interest and consists of P_2 and P_5, then P_2 loses its identifying property for the group of interest and the minimal set becomes P_1 only (Fig 1B). Not every group of interest can be exclusively identified by a set of proteins: in some cases, such as Fig 1C, where B_4 is not a part of the group of interest and contains P_1, P_2 and P_5, there is no set of proteins that can identify the $B_1 \cup B_2$ group. As the number of proteins and bacteria

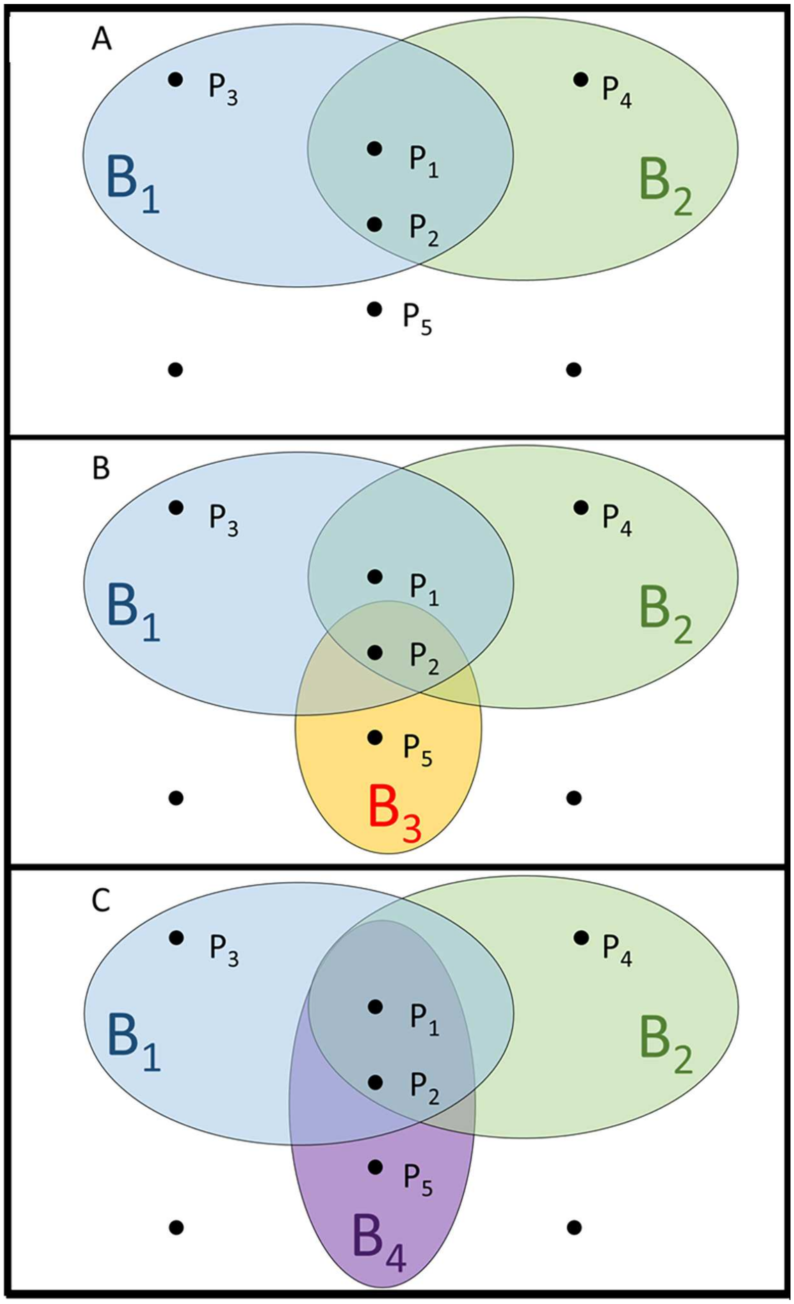


Fig 1. Graphical representation of the proteins (denoted P_1, P_2, P_3, P_4, P_5) which can serve as markers for the bacterial (denoted B_1, B_2, B_3, B_4) group of interest consisting of B_1 and B_2: (A) shows that P_1, P_2 can serve as a minimal set of markers for the group of interest; (B) P_1 only can serve as a marker for the group of interest; and (C) there are no markers for the group of interest.

https://doi.org/10.1371/journal.pone.0195537.g001

increases, an exact solution for finding a minimal set of proteins out of millions of known proteins in order to identify any group of interest out of thousands of organisms becomes impossible to solve in a reasonable timeframe. We will show that this problem cannot be solved efficiently since it is NP-hard [16,17], namely, no efficient algorithm for solving this problem is known. The definition of the hitting set problem is the following: given a ground set *S* and a collection *C* of subsets of *S*, find a hitting set with a minimum cardinality, i.e., a subset $S' \subseteq S$

such that S' contains at least one element from each subset in C. We will elaborate on the exact connection between the hitting set problem and our problem of identifying a set of proteins in the Materials and Methods subsection named Problem Definitions and Notations. In S1 Algorithm, we show that even if we are willing to relax our problem to that of finding a hitting set of a limited size, an exact approach is impractical. Since this problem is of great importance, a lot of effort has been made to find efficient approximation algorithms to it. That is, efficient algorithms that return a solution to the hitting set problem, which is at most r times the size of a minimum set, where r is the approximation factor. Here, we apply an approximation algorithm which finds relatively small sets of proteins that identify the group of interest. Obviously, these sets are not necessarily minimal. For example, in Fig 1A, an approximate solution might be the set containing both P_1 and P_2, whereas an exact (i.e. minimal) solution will be either P_1 or P_2.

To find an approximation algorithm to the hitting set problem, we explore approximation algorithms to an equivalent problem, the "set cover" problem [16], which in addition to several known approximation algorithms has many heuristics for solving it efficiently [18–27]. The definition of the set cover problem is the following: given a ground set S and a collection C of subsets of S, find a subset $C' \subseteq C$ such that every element in S is contained in some subset in C'. When considering the most suitable algorithm for our work, we looked for algorithms that will be both theoretically and practically efficient. It was proven that the set cover problem is approximable within $O(log|\mathrm{S}|)$, where $|S|$ is the size of the ground set S, using a greedy algorithm [28–31]. Therefore, a greedy algorithm can achieve the same corresponding approximation ratio as the hitting set problem. Moreover, a randomized algorithm based on linear programming achieves the same approximation ratio with a probability of at least 1-1/$|S|$ [32]. It was shown that the set cover problem cannot be approximated in polynomial time to within a factor of $(1 - o(1)) \cdot ln|S|$ (unless $P = NP$) [33,34]. This means that the greedy approximation algorithm and the randomized algorithm theoretically achieve the best possible ratio. Thus, we implemented the greedy algorithm and a randomized algorithm based on randomized rounding of linear program constraints [35–38]. Both algorithms were used to identify non-anammox nitrifying bacteria and predatory bacteria as phenotypic groups, Archaea and *Escherichia/Shigella* as taxonomic groups, and 13 different pathogenic sub-groups of *E. coli* as combined phenotypic/taxonomic groups.

Materials and methods

Biological Data. Genome and protein data were obtained as outlined in [6] from the 2016 version of the orthologous protein cluster table created and maintained by the microbial genome database (MBGD) [39] and freely available at http://mbgd.genome.ad.jp/htbin/view_arch.cgi. This table is updated yearly and is arranged so that each row is an orthologous cluster (i.e. the same protein) and each column is a genome. Ortholog identification and grouping is performed by the DomClust [40] and DomRefine [39] procedures, with MergeTree [41] adding new genomes to the table. Ortholog classification is based on all-against-all clustering with local alignment of the protein domain sequences. The raw MBGD data were automatically cleaned, as outlined in [6]: unnecessary and redundant data were deleted, and all protein occurrences were transformed from protein names into binary data so that each datapoint in the table contains either a one or a zero only (the protein is present or absent in the genome, respectively). This reduces the file size by two orders of magnitude and enables efficient algorithm usage. This approach enables one to make use of any orthologous group classification, for example splitting or merging groups according to the biological system. In order to fully challenge our methods and algorithms, we implemented the algorithm using 17 different microbial groups which represent a wide variety of classification criteria: non-anammox nitrifying bacteria and predatory bacteria as phenotypic groups, Archaea and *Escherichia/Shigella* as taxonomic Groups, and 13 different subtypes of pathogenic *E. coli* as taxonomic/phenotypic groups. First, of the 4742 complete (i.e. non-draft) genomes of bacteria and Archaea available in the MBGD database, 12 belonged to non-anammox nitrifying bacteria from two groups: ammonia oxidizers (*Nitrosomonas europaea* ATCC 19718, *Nitrosomonas eutropha* C71, *Nitrosomonas* sp. AL212, *Nitrosomonas* sp. Is79A3, *Nitrosospira multiformis* ATCC 25196, *Nitrosococcus halophilus* Nc4, *Nitrosococcus oceani* ATCC 19707, *Nitrosococcus watsoni* C-113, *Nitrobacter hamburgensis* X14, *Nitrosomonas communis* Nm2) and nitrite oxidizers (*Nitrobacter winogradskyi* Nb-255, *Nitrospira defluvii*). Genomes of bacterial nitrifiers using anaerobic ammonium oxidation (anammox) were not available. Second, MBGD contained 16 genomes belonging to known predatory bacteria: *Bdellovibrio bacteriovorus* HD100, *Bdellovibrio bacteriovorus* 109J, *Bdellovibrio bacteriovorus* tiberius, *Bdellovibrio bacteriovorus* W, *Bdellovibrio exovorus*, *Bacteriovorax marinus* SJ, *Cytophaga hutchinsonii* ATCC 33406, *Flavobacterium johnsoniae* UW10, *Herpetosiphon aurantiacus* ATCC 23779, *Micavibrio aeruginosavorus* ARL-13, *Micavibrio aeruginosavorus* EPB, *Myxococcus xanthus* DK 1622, *Sorangium cellulosum* So ce 56, *Sorangium cellulosum* So0157-2, *Saprospira grandis* Lewin, *Stigmatella aurantiaca* DW4/3-1. Third, we obtained all 226 archaeal genomes available at MBGD, comprising all archaeal classes and families (Table A in S1 File); fourth, the 147 genomes belonging to the *Escherichia/Shigella* bacterial genus (Table B in S1 File); and fifth, the genomes comprising 13 *E. coli* pathotypes (Table B in S1 File). The taxonomic and phenotypic identifications were based on the latest published data[42]. E. coli strain pathogroup assignment was according to the EnteroBase database freely available at https://enterobase.warwick.ac.uk

Problem Definition and Notation. Given a set $\mathcal{B}$ of bacteria, a subset B of $\mathcal{B}$, and a set of orthologous proteins P, we want to find a minimum subset $\hat{P}$ of P that would identify the bacteria in B. Namely, all orthologous proteins in $\hat{P}$ are in every bacterium in B, and for every bacterium in $\bar{B}:=\mathcal{B}\setminus B$ there is at least one ortholog protein in $\hat{P}$ that is missing. We consider each bacterium as a subset of the orthologous proteins it contains. Thus, the formal definition of the problem is as follows: given a ground set of m elements $P = \{p_1...,p_m\}$, a collection $\mathcal{B}$ of n subsets $\mathcal{B} = \{B_1, \ldots, B_n\}$, such that $B_j \subseteq P$ and $|B_j| \geq 1$ for all $1 \leq j \leq n$, and a subcollection $B \subseteq \mathcal{B}$ of size k, $B = \{B_1...,B_k\}$, we want to find a minimum size subset of P, $\hat{P}$, such that for each $p \in \hat{P}$ it holds that $p \in \bigcap_{j=1}^{k} B_j$, and for each $B_j \in \mathcal{B}\setminus B$ it holds that there exists some $p \in \hat{P}$ such that $p \notin B_j$. *Algorithm Development*. We use a greedy approximation algorithm and a randomized approximation that is based on Linear Programming for the hitting set problem. An advantage for the random algorithm is that different runs of the algorithm may produce different results in an efficient manner.

Algorithm 1, based on hitting set

```
1. $\hat{P} \leftarrow B_1$.
2. for all $p \in \hat{P}$ do
    (a) if $p \notin B_2 \cap B_3 \cap ... \cap B_k$ then $\hat{P} \leftarrow \hat{P} \setminus \{p\}$
3. for $k + 1 \leq j \leq n$ let $\tilde{B}_j = \{p \in \hat{P} | p \notin B_j\}$.
4. let $\tilde{B} = \{\tilde{B}_{k+1}, \ldots, \tilde{B}_n\}$.
5. if $\tilde{B}$ contains an empty set–return "no hitting set".
```

```
6. run an algorithm of the hitting set problem on the input $\hat{P}, \tilde{B}$. Namely, sub algorithm 1 or 2 on
   input $\hat{P}, \tilde{B}$, respectively.
```

We can conclude that there is no possible hitting set for an instance of a problem in step 5 even before running an explicit sub-algorithm for the hitting set problem. Let $\hat{P}$ and $\tilde{B}$ be the result sets at the end of stage 4 in algorithm 1. If at least one of the sets in $\tilde{B}$ is empty, then obviously there is no hitting set. Notice that if all sets in $\tilde{B}$ are not empty, there is always a hitting set since we can take $\hat{P}$. By the construction of sets in $\tilde{B}$, $\hat{P}$ must hit each set $B_i \in \tilde{B}$ if it is not empty. As noted in the previous subsection, the hitting set problem is equivalent to the set cover problem. Moreover, we can use any algorithm to the set cover problem to solve the hitting set problem. Consider an instance S, C to the set cover problem. We define the following instance to the hitting set problem. The ground set is defined to be $\hat{S} = C$, namely, each element is a subset of the instance to the set cover problem. For each element in $e \in S$, we define a subset of $\hat{S}$ which is the set of subsets of S that contain e. Therefore, a minimum cardinality cover of S, C is a minimum cardinality hitting set of $\hat{S}$, $\hat{C}$. To solve Item 6 of Algorithm 1 we first use the following greedy algorithm for the hitting set problem:

Sub algorithm 1, greedy algorithm for hitting set (S, C)

```
Input: universe $S = \{s_1,...,s_m\}$, $C = \{C_1,...,C_n\}$, s.t. $C_i \subseteq S$ for all $1 \leq i \leq n$.
Output: $\hat{S} \subseteq S$.
1. $\hat{S} \leftarrow \emptyset$.
2. $\hat{C} \leftarrow C$.
3. while $\hat{C} \neq \emptyset$ do
    a. Select $s \in S$ such that $s$ hits the largest number of subsets in $\hat{C}$ (i.e. select $s$ s.t. $|\{C_i \in \hat{C} | C_i \cap \{s\} \neq \emptyset\}|$ is of maximum cardinality)
    b. Remove the hit subsets from $\hat{C}$, namely $\hat{C} \leftarrow \hat{C} \setminus \{C_i \in \hat{C} | C_i \cap \{s\} \neq \emptyset\}$.
    c. $\hat{S} \leftarrow \hat{S} \cup \{s\}$.
4. Return $\hat{S}$.
```

Lemma 1 given an instance (S,C), sub-algorithm 1 finds a hitting set of size of at most $OPT \cdot O(\log|C|) = OPT \cdot O(\log n)$, where OPT is the size of an optimal HS, with time complexity $O(\min\{m,n\} \cdot n \cdot m)$.

Proof The correctness and approximation ratio of the lemma follows from the correctness and approximation ratio of the greedy algorithm for the set cover problem [29], and the tight connection between the set cover problem and the hitting set problem [43]. To compute the time complexity of the algorithm, let $\hat{C}_j$ be the set that needs to be hit at step j, and let $\hat{S}_j$ be the set of elements that have not been selected yet at step j. The most expensive operation in the loop of Item 3 is Item 3a. In the worst case of Item 3a for every $s \in S$, the algorithm goes over all the subsets $C_i \in \hat{C}_j$ to check whether $s \in C_i$. Using a reasonable data structure, we can assume that the access time for checking whether $s \in C_i$ is $O(1)$. Therefore, the running time of Item 3a at step j is $|\hat{S}_j| \cdot (\sum_{C_i \in \hat{C}j} O(1)) = |\hat{S}_j| \cdot |\hat{C}_j|$. Let ℓ be the maximum number of steps the loop in Item 3 is performed. Obviously $\ell \leq min\{m, n\}$. Thus, time complexity of Sub-

Algorithm 1 is.

$$O\left(\sum_{j=1}^{\ell}|\hat{S}_j|\cdot|\hat{C}_j|\right)=O\left(\sum_{j=1}^{\min\{m,n\}}(m-j)\cdot(n-j)\right)=O(\min\{m,n\}\cdot n\cdot m)$$

Notice that the theoretical upper bound on the time complexity can be reduced for specific instances of the problem. For example, if every element in S appears in many subsets in C then the number of steps ℓ, of performing the loop in Item 3 of Sub-Algorithm 1 is much smaller than n.

Lemma 1 implies the following theorem:

Theorem 1 Algorithm 1 finds a set of proteins $\hat{P}$ such that $|\hat{P}|=O(log(n-k)\cdot|P_{OPT}|)$, where P_{OPT} is an optimal set of proteins that identifies B, with time complexity of $O(m^2\cdot n)$.

Proof: We first note that the initial set $\hat{P}$ is an intersection of all the subsets of proteins in the tested set $\{B_1,. . .,B_k\}$. Therefore, any set returned by the algorithm to identify the tested set must be a subset of the initial $\hat{P}$ Moreover, the returned set must not identify the control set $\{B_{k+1},. . .,B_n\}$, Since Sub-Algorithm 1 returns a subset of $\hat{P}$ that is a hitting set of $\tilde{B}$, for each bacteria B_j of the control set $\{B_{k+1},. . .,B_n\}$ the returned $\hat{P}$ must include at least one protein that does not exist in B_j. This implies the correctness of the algorithm.

By lemma 1, with $S = B_1 \cap. . .\cap B_k$ and $C=\tilde{B}$, it holds that $|\hat{P}|=OPT\cdot O(\log|\tilde{B}|)=OPT\cdot O(\log(n-k))$, where OPT is an optimal solution to the HS problem on the given S, C. Notice that by the definition of our problem, every optimal solution for identifying B needs to find a HS of $\tilde{B}$ as well. Thus, it holds that $OPT = |P_{OPT}|$, so $|\hat{P}|=|P_{OPT}|\cdot O(log(n-k))$, as claimed.

To prove the bound on the time complexity, it must be noted that the running time of Item 2 of Algorithm 1 is $O(|B_1|\cdot\sum_{j=1}^{k}|B_j|)$. The running time of Item 3 of Algorithm 1 is

$$O\left(|\hat{P}|\cdot\sum_{j=k+1}^{n}|B_j|\right)=O\left(|B_1\cap\ldots\cap B_k|\cdot\sum_{j=k+1}^{n}|B_j|\right)=O\left(|B_1|\cdot\sum_{j=k+1}^{n}|B_j|\right).$$

By Lemma 1 the time complexity of finding the hitting set in item 5 on $(\hat{P},\tilde{B})$ is

$$\begin{aligned}&O(\min\{|B_1\cap\ldots\cap B_k|,(n-k)\}\cdot|B_1\cap\ldots\cap B_k|\cdot(n-k))\\&\quad=O(\min\{|B_1|,(n-k)\}\cdot|B_1)\cdot(n-k)).\end{aligned}$$

Therefore, the total time complexity of the algorithm is

$$\begin{aligned}&O\left(|B_1|\cdot\sum_{j=1}^{n}|B_j|+\min\{|B_1|,n-k\}\cdot|B_1|\cdot(n-k)\right)\\&\quad=O(m^2\cdot n+\min\{m,n)\cdot m\cdot(n-k)=O(m^2\cdot n).\end{aligned}$$

This completes the theorem.

Notice that again, in practice, the actual time complexity is smaller than our upper bound, which makes our algorithm very fast in reality, as demonstrated in the next section. We now show another version of Algorithm 1 in which we replace Sub-Algorithm 1 by a Linear Programming randomized algorithm for the hitting set problem. For $1\le i\le m$ set $x_i = 1$ if $s_i\in\hat{S}$ and $x_i = 0$ otherwise. Thus, finding a hitting set can be formulated as an integer linear program. We relax the integer linear program to a fractional one, and then use randomized rounding to get an integer solution.

Sub Algorithm 2, linear programming randomized algorithm for hitting set (S, C)

Input: universe $S = \{s_1, \ldots, s_m\}$, $C = \{C_1, \ldots, C_n\}$, *s.t.* $C_i \subseteq S$ *for all* $1 \leq i \leq n$.
Output: $\hat{S} \subseteq S$.

1. *Solve the following fractional linear programming problem:*

$$Minimize \sum_{i=1}^{m} x_i$$

$$subject\ to \sum_{i:s_i \in C_j} x_i \geq 1,\ \forall C_j \in C$$

$$0 \leq x_i \leq 1, 1 \leq i \leq m$$

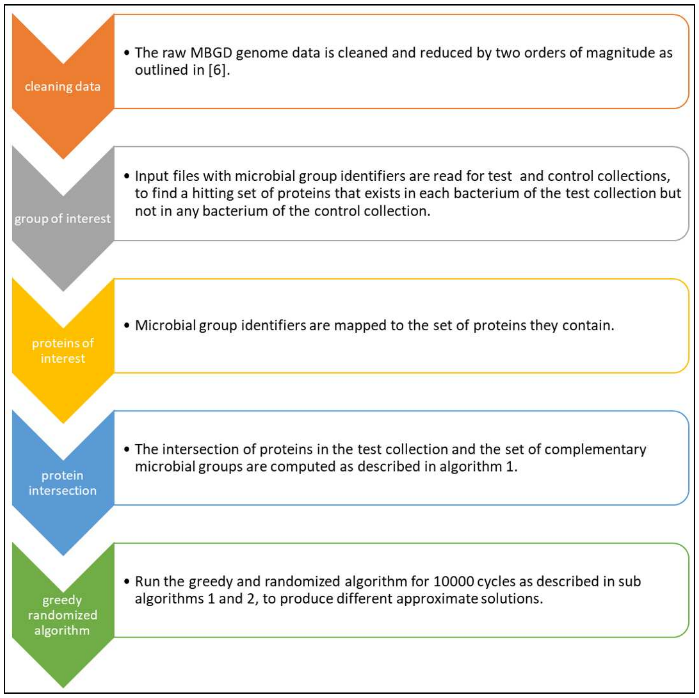
2. *For all* $1 \leq i \leq m$ *let* $\hat{x}_i$ *be the value assigned to* x_i*, in an optimal fractional solution of the previous linear program.*
3. *Let* X_i*,* $1 \leq i \leq m$ *be independent random variables such that*

$$Pr[X_i = 0] = (1 - \hat{x}_i)^{c \cdot \log\ n}, for\ constant\ c,\ and$$

$$Pr[X_i = 1] = 1 - (1 - \hat{x}_i)^{c \cdot logn}.$$

4. $\hat{S} \leftarrow \{s_i \in S | X_i = 1\}$.
5. *Return* $\hat{S}$.

According to [35–38] Sub-Algorithm 2 returns a hitting set of size $c \cdot \log n$ times the size of an optimal solution, with a probability of at least $1 - \frac{1}{2^c}$. Therefore Algorithm 1, when using Sub-Algorithm 2, achieves, with high probability, the same theoretical approximation ratio of Algorithm 1 when using Sub-Algorithm 1. As the constant c presented in item 3 of the algorithm grows, the probability that the result is indeed a valid HS grows, but the expected size of the result also increases. Based on our experience, choosing the constant $c = 1$ gives us a relatively small error probability, while not increasing the received HS size. In each case the random algorithm returned a HS, we verified that it is indeed a HS. The time complexity of Sub-Algorithm 2 is derived mainly from the time complexity of solving the Linear Program of the problem in step 1. We have used Dantzig's Simplex method [44] to solve linear programs. This method has an exponential time complexity in the worst case but is highly efficient in practice. It is possible to rerun this sub algorithm efficiently as we note that once we have solved the linear program, we are left with assigning a binary value for each variable and then deciding whether the result is an actual hitting set, so the time complexity of a single rerun for an instance of this algorithm is $n \cdot m$. For our purposes, we have found that 10,000 reruns of this algorithm per instance produces reasonably low runtime and at the same time produces a large variety of interesting outcomes. In some occasions, the result of the randomized algorithm may be optimal and deterministic. If the solution for the linear program is binary, that is, each variable receives either 0 or 1, then we have found a hitting set without the need for linear relaxation. That is the optimum value for the integer program of the given instance for exact hitting set program. In that case, it is redundant to rerun the algorithm as it will generate the same optimal result. This randomized approach presents several different hitting sets.



- The raw MBGD genome data is cleaned and reduced by two orders of magnitude as outlined in [6].
- Input files with microbial group identifiers are read for test and control collections, to find a hitting set of proteins that exists in each bacterium of the test collection but not in any bacterium of the control collection.
- Microbial group identifiers are mapped to the set of proteins they contain.
- The intersection of proteins in the test collection and the set of complementary microbial groups are computed as described in algorithm 1.
- Run the greedy and randomized algorithm for 10000 cycles as described in sub algorithms 1 and 2, to produce different approximate solutions.

Fig 2. Software structure and output.

https://doi.org/10.1371/journal.pone.0195537.g002

An overview representation of the software structure is presented in Fig 2. This section details the description of the algorithm. The source code can be downloaded directly from https://www.dropbox.com/sh/s6u8fh69ygzkuuk/AABLpFPyWjY3kLGID6H2cG-Ja?dl=0

The algorithm may produce three types of solutions: (i) Optimal minimal solutions. It is sometimes possible to guarantee for a result set to be minimal (rather than an approximation) as a result of a possible optimal solution for the linear program for the HS formulation with sub-algorithm 2. (ii) Collection of approximated solutions. One or more hitting sets that are no larger than $OPT \cdot O(\log n)$ where OPT is the minimal solution and n is the number of sets to hit (microbial groups in our case). (iii) No solution. In the process of algorithm 1, it is possible to say whether there is no set that exclusively hits all of the tested groups of an instance of the problem.

Results and discussion

Our algorithms were employed to find the hitting set of 17 different microbial groups representing a wide variety of classification criteria (Table 1): 1) The non-anammox nitrifying bacteria, a group distinct from all other bacteria by its metabolic phenotype; 2) The second phenotypic group was the predatory bacteria, a group distinct by its trophic phenotype; 3) The third tested group was a taxonomic one–Archaea, a group distinct from Bacteria by domain-level taxonomy; 4) The fourth tested group was a taxonomic one: the *Escherichia/Shigella* groups, distinct by genus-level taxonomy, and; 5) The other 13 tested groups were of pathogenic *E. coli*. each distinguished from all other bacteria by both taxonomy and phenotype. In all cases where hitting sets were found, one of the hitting sets found by the random algorithm was identical to the hitting sets found by the greedy algorithm. The non-anammox nitrifiers used in this study are a group of 12 bacteria with a known genome, which perform the nitrification of ammonia or ammonium to nitrite. Five minimal HS were discovered, each

Table 1. Hitting sets (marker proteins) of 17 microorganism groups. HS, hitting set. Min., minimal. Greedy and random refer to the algorithm type. Phen., phenotypic. Tax., taxonomic. AIEC, adherent-invasive *E. coli*. EPEC, enteropathogenic *E. coli*. UPEC, uropathogenic *E. coli*. STEC, Shiga toxin-producing *E. coli*. NMEC, neonatal meningitis-associated *E. coli*. ExPEC, extra-intestinal pathogenic *E. coli*. ETEC, enterotoxigenic *E. coli*. EIEC, enteroinvasive *E. coli*. EHEC, enterohemorrhagic *E. coli*. EAEC, enteroaggregative *E. coli*. APEC, avian pathogenic *E. coli*. EAHEC, enteroaggregative hemorrhagic *E. coli*.

Group name	Group distinction	Run time (sec)	No. of HS genes (greedy)	Min. no. of HS genes (random) with representative genes	No. of HS with min. no. of HS genes (random)	Proven optimal HS
Nitrifying bacteria	Phen.	7.32	5	5 (NirK, NirC)	5	0
Predatory bacteria	Phen.	7.11	-	-	0	-
Archaea	Tax. (domain)	7.28	1	1 (NusG)	1	1
Escherichia/ Shigella	Tax. (genus)	7.38	2	2 (YhiF)	10	0
All pathogenic *E. coli*	Phen. and tax.	7.35	-	-	0	-
AIEC	Phen. and tax.	7.35	3	3 (TnpR)	9	0
EPEC	Phen. and tax.	7.73	5	5 (YedK, ImpC)	7	0
UPEC	Phen. and tax.	7.39	-	-	0	-
STEC	Phen. and tax.	7.36	-	-	0	-
NMEC	Phen. and tax.	7.31	-	-	0	-
ExPEC	Phen. and tax.	7.37	-	-	0	-
ETEC	Phen. and tax.	7.37	-	-	0	-
EIEC	Phen. and tax.	7.43	-	-	0	-
EHEC	Phen. and tax.	7.30	-	-	0	-
EAEC	Phen. and tax.	7.34	-	-	0	-
APEC	Phen. and tax.	7.37	-	-	0	-
EAHEC	Phen. and tax.	7.38	4	3 (FliC)	1	1

https://doi.org/10.1371/journal.pone.0195537.t001

containing five proteins (Table C in S1 File). Among these proteins were nitrite reductase (NirK)[45], an enzyme that catalyzes the reduction of nitrite to nitric oxide; siroheme synthase (CysG), an enzyme involved in the synthesis of siroheme, a heme-like prosthetic group used by the nitrite reductase [46]; Formate/nitrite transporter (NirC), a transmembrane channel that transports nitrite in and out of the cell [44]; and a nitric oxide reductase activation protein (NorD)[45], involved in reducing nitric oxide to nitrous oxide. Although it may appear that these proteins are only relevant to the first group of non-anammox nitrifiers (i.e. nitrite oxidizers), it was actually discovered that the second group (ammonia oxidizers) also depend on the nitrite reductase enzyme for efficient growth by its oxidation of ammonia to nitrite via hydroxylamine [47]. Thus, many of the HS proteins that our algorithm discovered are highly relevant to the biological phenotype that is unique to all non-anammox nitrifying bacteria, serving as a kind of "positive control" for our algorithm. The second phenotypic group, predatory bacteria, was not found to have any hitting sets proteins when compared to other (i.e. non-predatory) bacteria. This confirms previous studies (e.g. Pasternak et al. [48]), which also concluded that bacterial predation is not facilitated by unique proteins, and serves as a kind of "negative control" for our algorithm. In many examples bacteria performing similar metabolic functions (e.g. nitrification) all use similar enzymes to carry out the process. However, predation is functionally a more diverse process, and there are no molecular signatures specific to bacterial predation. Indeed, several features were shown to be highly enriched in predators, including adhesins, proteases and particular metabolic proteins, used for binding to, processing and consuming prey, respectively; in addition, most predators use the mevalonate pathway of isoprenoid biosynthesis, whereas almost all other bacteria use the DOXP pathway [48]. However, all of these proteins are also found in the genomes of non-predatory bacteria, and the differences might not lie in their presence or absence but rather in transcriptional and/or post-transcriptional regulation.

The next group, Archaea, was compared to all the bacterial genomes in the database and an optimal single protein (NusG) HS solution was found to sufficiently distinguish the groups. NusG is a transcription elongation factor that binds to RNA polymerases and assists in RNA synthesis from a DNA template[49]. RNA polymerases are under strong evolutionary pressure to maintain their structure; therefore, the structure and sequence of NusG should also be conserved. In fact, this protein is considered to be the only transcription elongation factor whose sequence is universally conserved in all three domains of life: Bacteria, Archaea and Eukarya [50]. Surprisingly, our analysis found the amino-acid sequence to be very distinct between Archaea and the other two domains, and a consequent phylogenetic analysis of 500 NusG protein sequences (Fig 3) clearly confirmed this difference. Since this solution is an optimal solution the algorithms are deterministic—meaning there might be other genes specifically conserved only in Archaea in the current database which the algorithm ignores.

The fourth tested group is a taxonomic one; the bacterial genus *Escherichia/Shigella* was characterized by ten hitting sets, all with two proteins, where one of them is the YhiF protein which is a transcription regulator (Table D in S1 File). YhiF takes part in modulating levels of expression of LEE (locus of enterocyte effacement) proteins—a group of proteins responsible for attaching and effacing to the intestine of the host. Through the mechanisms of control of LEE expression, YhiF appears to play a central role in *Escherichia/Shigella* colonizing the host intestinal epithelium [51]. However, as it is present in all *Escherichia/Shigella* strains, including commensal ones, it is a taxonomic marker rather than a pathogenicity marker.

Finally, we searched for hitting sets which define 13 different subgroups—or pathotypes—of pathogenic *E. coli* bacteria. These pathotypes are defined according to their clinical symptoms (e.g. enterohemorrhagic) and therefore most do not have perfect marker proteins, i.e. ones that exist in all the strains of a specific pathotype and only in those strains [49]. Indeed, our analysis found that 10 of these 13 pathogroups showed no hitting sets, confirming that

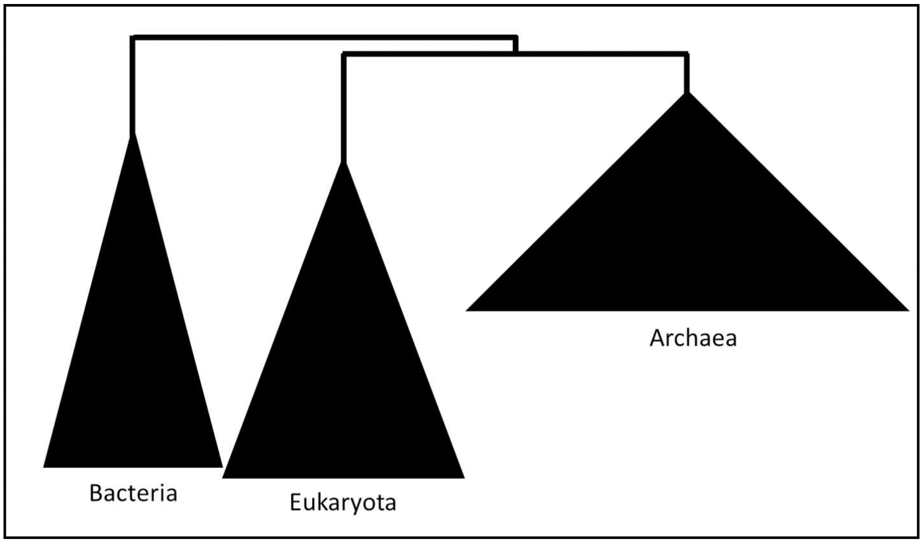


Fig 3. Maximum-likelihood phylogenetic tree of the NusG protein. All archaeal NusG sequences were taken from the GenBank database, along with their most similar bacterial and eukaryotic homologs for a total of 500 protein sequences. The bootstrap consensus tree inferred from 100 replicates was taken to represent the evolutionary history of the taxa analyzed. Branches were merged at the domain level.

https://doi.org/10.1371/journal.pone.0195537.g003

genomes within each group do not necessarily have taxonomic or phenotypic affiliation with each other and therefore non-housekeeping marker proteins are impossible to find in most groups. Even so, three important pathotypes could be efficiently defined: adherent-invasive *E. coli* (AIEC) with minimal hitting sets containing three proteins (Table E in S1 File), enteropathogenic *E. coli* (EPEC) with five proteins (Table F in S1 File) and enteroaggregative hemorrhagic *E. coli* (EAHEC) with three proteins (Table G in S1 File). EPEC is an important cause of diarrhea and premature death in children, especially in developing countries [51]; currently, the major diagnostic marker for EPEC is the *eae* gene (coding for intimin, an outer membrane adhesive protein), yet genomes from other pathotypes also possess this gene [52], making it a non-optimal marker. AIEC is not associated with diarrhea but is thought to contribute to the development of Crohn's disease, a chronic inflammatory bowel syndrome [53]. It currently has no known diagnostic markers [52]. EAHEC is associated with food poisoning in the developed world [54]. Many of the hitting sets proteins that we found are either uncharacterized (i.e. 'hypothetical') or are from a viral (bacteriophage) source with DNA cut-and-paste functionality, e.g. transposases and integrases. One of our EAHEC HS members, FliC, has been used before as a marker for this group [54]. *E. coli* subtyping schemes are invaluable in identifying outbreaks and treating infection patients, but the current subtyping technology is imprecise and potentially misleading because *E. coli* genomes constantly change and evolve.

Our algorithm enables the accurate identification of marker genes for any microbial group, depending on the completeness of the database. Once marker genes are established and confirmed, new and unknown genomes can quickly be assigned to their group via MLST, without the need for whole-genome sequencing. specifically, our study found potential marker genes which in the future may enable reliable diagnosis of the EAHEC, EPEC and AIEC strains of pathogenic *E. coli*, thus improving the treatment of *E. coli*-related diseases. In addition to microbial identification, such analysis may help uncover novel genes pertinent to the grouping, such as virulence-associated or habitat-specific genes. As these genes are group-specific, they are prime candidates for further research which aims to understand the genetic basis of the group's phenotype as well as possible targets for antibiotic treatment. Finally, our algorithm may also help discover novel functions which are uniquely shared among a group of microbes. For example, several uncharacterized ("hypothetical") genes were found in this study to be pathogroup-specific; further investigation of these genes and proteins may reveal their possible function in connection to their specific group, leading to improved understanding and specific antibacterial treatments.

Supporting information

S1 File. Microbial groups and their genomes, and hitting sets (HS) found in this study. Table A in S1 Fie) Archaea genomes; Table B in S1 File) *Escherichia*/*Shigella* genomes; Table C in S1 File) nitrifiers HS; Table D in S1 File) *Escherichia*/*Shigella* HS; Table E in S1 File) AIEC (adherent-invasive *E. coli*) HS; Table F in S1 File) EPEC (enteropathogenic *E. coli*) HS. Table G in S1 File) EAHEC (enteroaggregative hemorrhagic *E. coli*) HS.
(XLSX)

S1 Algorithm. A mathematical explanation for the necessity of approximation solution.
(DOCX)

Author Contributions

Conceptualization: Elad Segev, Zohar Pasternak, Tom Ben Sasson, Edouard Jurkevitch, Mira Gonen.

Formal analysis: Elad Segev, Zohar Pasternak, Tom Ben Sasson.

Investigation: Zohar Pasternak.

Methodology: Elad Segev, Zohar Pasternak, Tom Ben Sasson, Mira Gonen.

Software: Elad Segev, Tom Ben Sasson, Mira Gonen.

Supervision: Elad Segev, Edouard Jurkevitch, Mira Gonen.

Validation: Elad Segev, Zohar Pasternak, Tom Ben Sasson.

Writing – original draft: Elad Segev, Zohar Pasternak, Tom Ben Sasson, Edouard Jurkevitch, Mira Gonen.

Writing – review & editing: Elad Segev, Zohar Pasternak, Tom Ben Sasson, Edouard Jurkevitch, Mira Gonen.

References

1. Fleischmann RD, Adams MD, White O, Clayton RA, Kirkness EF, Kerlavage AR, et al. Whole-Genome Random Sequencing and Assembly of Haemophilus Influenzae Rd. Science. 1995; 269: 496–512. PMID: 7542800
2. Hall N. Advanced sequencing technologies and their wider impact in microbiology. J Exp Biol. 2007; 210: 1518–1525. https://doi.org/10.1242/jeb.001370 PMID: 17449817
3. Kersey PJ, Allen JE, Christensen M, Davis P, Falin LJ, Grabmueller C, et al. Ensembl Genomes 2013: scaling up access to genome-wide data. Nucleic Acids Res. 2014; 42: D546–D552. https://doi.org/10.1093/nar/gkt979 PMID: 24163254
4. Kersey PJ, Allen JE, Armean I, Boddu S, Bolt BJ, Carvalho-Silva D, et al. Ensembl Genomes 2016: more genomes, more complexity. Nucleic Acids Res. 2016; 44: D574–D580. https://doi.org/10.1093/nar/gkv1209 PMID: 26578574
5. Land M, Hauser L, Jun S- R, Nookaew I, Leuze MR, Ahn T-H, et al. Insights from 20 years of bacterial genome sequencing. Funct Integr Genomics. 2015; 15: 141–161. https://doi.org/10.1007/s10142-015-0433-4 PMID: 25722247
6. Pasternak Z, Sasson TB, Cohen Y, Segev E, Jurkevitch E. A New Comparative-Genomics Approach for Defining Phenotype-Specific Indicators Reveals Specific Genetic Markers in Predatory Bacteria. PLOS ONE. 2015; 10: e0142933. https://doi.org/10.1371/journal.pone.0142933 PMID: 26569499
7. Pérez-Losada M, Cabezas P, Castro-Nallar E, Crandall KA. Pathogen typing in the genomics era: MLST and the future of molecular epidemiology. Infect Genet Evol J Mol Epidemiol Evol Genet Infect Dis. 2013; 16: 38–53. https://doi.org/10.1016/j.meegid.2013.01.009 PMID: 23357583
8. Leopold SR, Sawyer SA, Whittam TS, Tarr PI. Obscured phylogeny and possible recombinational dormancy in Escherichia coli. BMC Evol Biol. 2011; 11: 183. https://doi.org/10.1186/1471-2148-11-183 PMID: 21708031
9. Dutilh BE, Snel B, Ettema TJG, Huynen MA. Signature Genes as a Phylogenomic Tool. Mol Biol Evol. 2008; 25: 1659–1667. https://doi.org/10.1093/molbev/msn115 PMID: 18492663
10. Wu D, Jospin G, Eisen JA. Systematic Identification of Gene Families for Use as "Markers" for Phylogenetic and Phylogeny-Driven Ecological Studies of Bacteria and Archaea and Their Major Subgroups. PLOS ONE. 2013; 8: e77033. https://doi.org/10.1371/journal.pone.0077033 PMID: 24146954
11. Kristensen DM, Waller AS, Yamada T, Bork P, Mushegian AR, Koonin EV. Orthologous Gene Clusters and Taxon Signature Genes for Viruses of Prokaryotes. J Bacteriol. 2013; 195: 941–950. https://doi.org/10.1128/JB.01801-12 PMID: 23222723
12. Yoshida CE, Kruczkiewicz P, Laing CR, Lingohr EJ, Gannon VPJ, Nash JHE, et al. The Salmonella In Silico Typing Resource (SISTR): An Open Web-Accessible Tool for Rapidly Typing and Subtyping Draft Salmonella Genome Assemblies. PLOS ONE. 2016; 11: e0147101. https://doi.org/10.1371/journal.pone.0147101 PMID: 26800248
13. Brinkac LM, Beck E, Inman J, Venepally P, Fouts DE, Sutton G. LOCUST: a custom sequence locus typer for classifying microbial isolates. Bioinforma Oxf Engl. 2017; 33: 1725–1726. https://doi.org/10.1093/bioinformatics/btx045 PMID: 28130240
14. Huang B, Zhao D, Fang N-X, Hall A, Eglezos S, Blair B. An optimized binary typing panel improves the typing capability for Campylobacter jejuni. Diagn Microbiol Infect Dis. 2013; 77: 312–315. https://doi.org/10.1016/j.diagmicrobio.2013.09.005 PMID: 24139878

15. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015; 25: 1043–1055. https://doi.org/10.1101/gr.186072.114 PMID: 25977477

16. Karp RM. Reducibility among Combinatorial Problems. Complexity of Computer Computations. Springer, Boston, MA; 1972. pp. 85–103. https://doi.org/10.1007/978-1-4684-2001-2_9

17. Leeuwen J van. Algorithms and complexity. Amsterdam [u.a.: Elsevier [u.a.; 1998.

18. Parallel Randomized Heuristics For The Set Covering Problem—Semantic Scholar [Internet]. [cited 17 Sep 2017]. Available: /paper/Parallel-Randomized-Heuristics-For-The-Set-Coverin-STELLA-CATA-LANO/daf62fd516301432a6b649a4a464e7ccec795b27

19. Chu PC, Beasley JE. A Genetic Algorithm for the Multidimensional Knapsack Problem. J Heuristics. 1998; 4: 63–86. https://doi.org/10.1023/A:1009642405419

20. An effective and simple heuristic for the set covering problem—Semantic Scholar [Internet]. [cited 17 Sep 2017]. Available: /paper/An-effective-and-simple-heuristic-for-the-set-cove-Lan-DePuy/2eea0face6ad1e1f47d5954c023e99a2a68fa33c

21. Caprara A, Fischetti M, Toth P. A Heuristic Method for the Set Covering Problem. Oper Res. 1999; 47: 730–743. https://doi.org/10.1287/opre.47.5.730

22. Cormode G, Karloff H, Wirth A. Set Cover Algorithms for Very Large Datasets. Proceedings of the 19th ACM International Conference on Information and Knowledge Management. New York, NY, USA: ACM; 2010. pp. 479–488. 10.1145/1871437.1871501

23. Kordalewski D. New Greedy Heuristics For Set Cover and Set Packing. ArXiv13053584 Cs. 2013; Available: http://arxiv.org/abs/1305.3584

24. Umetani S, Yagiura M. RELAXATION HEURISTICS FOR THE SET COVERING PROBLEM. J Oper Res Soc Jpn. 2007; 50: 350–375. https://doi.org/10.15807/jorsj.50.350

25. Spasovski S, Madevska-Bogdanova A. Optimization of the Polynomial Greedy Solution for the Set Covering Problem. Proceedings of the Tenth Conference on Informatics and Information Technology. Skopje, Macedonia: Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, Macedonia; 2013. pp. 175–177. Available: http://ciit.finki.ukim.mk

26. Goldschmidt O, Hochbaum DS, Yu G. A Modified Greedy Heuristic for the Set Covering Problem with Improved Worst Case Bound. Inf Process Lett. 1993; 48: 305–310. https://doi.org/10.1016/0020-0190(93)90173-7

27. Hassin R, Levin A. A Better-Than-Greedy Approximation Algorithm for the Minimum Set Cover Problem. SIAM J Comput. 2005; 35: 189–200. https://doi.org/10.1137/S0097539704444750

28. Chvatal V. A Greedy Heuristic for the Set-Covering Problem. Math Oper Res. 1979; 4: 233–235. https://doi.org/10.1287/moor.4.3.233

29. Lovász L. On the ratio of optimal integral and fractional covers. Discrete Math. 1975; 13: 383–390. https://doi.org/10.1016/0012-365X(75)90058-8

30. Johnson DS. Approximation algorithms for combinatorial problems. J Comput Syst Sci. 1974; 9: 256–278. https://doi.org/10.1016/S0022-0000(74)80044-9

31. Stein SK. Two combinatorial covering theorems. J Comb Theory Ser A. 1974; 16: 391–397. https://doi.org/10.1016/0097-3165(74)90062-4

32. Approximation Algorithms | Vijay V. Vazirani | Springer [Internet]. Available: http://www.springer.com/gp/book/9783540653677

33. Dinur I, Steurer D. Analytical Approach to Parallel Repetition. Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing. New York, NY, USA: ACM; 2014. pp. 624–633. 10.1145/2591796.2591884

34. Feige U. A Threshold of Ln N for Approximating Set Cover. J ACM. 1998; 45: 634–652. https://doi.org/10.1145/285055.285059

35. Khachiyan LG. Polynomial algorithms in linear programming. USSR Comput Math Math Phys. 1980; 20: 53–72. https://doi.org/10.1016/0041-5553(80)90061-0

36. Peleg D, Schechtman G, Wool A. Randomized approximation of bounded multicovering problems. Algorithmica. 1997; 18: 44–66. https://doi.org/10.1007/BF02523687

37. Raghavan P. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. 27th Annual Symposium on Foundations of Computer Science (sfcs 1986). 1986. pp. 10–18. 10.1109/SFCS.1986.45

38. Raghavan P, Tompson CD. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. Combinatorica. 1987; 7: 365–374. https://doi.org/10.1007/BF02579324

39. Uchiyama I, Mihara M, Nishide H, Chiba H. MBGD update 2015: microbial genome database for flexible ortholog analysis utilizing a diverse set of genomic data. Nucleic Acids Res. 2015; 43: D270–276. https://doi.org/10.1093/nar/gku1152 PMID: 25398900

40. Uchiyama I. Hierarchical clustering algorithm for comprehensive orthologous-domain classification in multiple genomes. Nucleic Acids Res. 2006; 34: 647–658. https://doi.org/10.1093/nar/gkj448 PMID: 16436801

41. Uchiyama I, Mihara M, Nishide H, Chiba H. MBGD update 2013: the microbial genome database for exploring the diversity of microbial world. Nucleic Acids Res. 2013; 41: D631–635. https://doi.org/10.1093/nar/gks1006 PMID: 23118485

42. Rosenberg E, DeLong EF, Stackebrandt E, Lory S, Thompson F, editors. The Prokaryotes: Prokaryotic Biology and Symbiotic Associations [Internet]. 4th ed. Berlin Heidelberg: Springer-Verlag; 2013. Available: //www.springer.com/la/book/9783642301933

43. Ausiello G, D'Atri A, Protasi M. Structure preserving reductions among convex optimization problems. J Comput Syst Sci. 1980; 21: 136–153. https://doi.org/10.1016/0022-0000(80)90046-X

44. HOW GOOD IS THE SIMPLEX ALGORITHM. Defense Technical Information Center; 1970.

45. Zumft WG. Cell biology and molecular basis of denitrification. Microbiol Mol Biol Rev. 1997; 61: 533–616. PMID: 9409151

46. Murphy MJ, Siegel LM, Tove SR, Kamin H. Siroheme: A New Prosthetic Group Participating in Six-Electron Reduction Reactions Catalyzed by Both Sulfite and Nitrite Reductases. Proc Natl Acad Sci U S A. 1974; 71: 612–616. PMID: 4595566

47. Cantera JJL, Stein LY. Role of nitrite reductase in the ammonia-oxidizing pathway of Nitrosomonas europaea. Arch Microbiol. 2007; 188: 349–354. https://doi.org/10.1007/s00203-007-0255-4 PMID: 17541778

48. Pasternak Z, Pietrokovski S, Rotem O, Gophna U, Lurie-Weinberger MN, Jurkevitch E. By their genes ye shall know them: genomic signatures of predatory bacteria. ISME J. 2013; 7: 756–769. https://doi.org/10.1038/ismej.2012.149 PMID: 23190728

49. Torres M, Balada J-M, Zellars M, Squires C, Squires CL. In vivo effect of NusB and NusG on rRNA transcription antitermination. J Bacteriol. 2004; 186: 1304–1310. https://doi.org/10.1128/JB.186.5.1304-1310.2004 PMID: 14973028

50. Yakhnin AV, Babitzke P. NusG/Spt5: are there common functions of this ubiquitous transcription elongation factor? Curr Opin Microbiol. 2014; 18: 68–71. https://doi.org/10.1016/j.mib.2014.02.005 PMID: 24632072

51. Robins-Browne RM. Traditional Enteropathogenic Escherichia coli of Infantile Diarrhea. Rev Infect Dis. 1987; 9: 28–53. https://doi.org/10.1093/clinids/9.1.28 PMID: 3547577

52. Robins-Browne RM, Holt KE, Ingle DJ, Hocking DM, Yang J, Tauschek M. Are Escherichia coli Pathotypes Still Relevant in the Era of Whole-Genome Sequencing? Front Cell Infect Microbiol. 2016; 6. https://doi.org/10.3389/fcimb.2016.00141 PMID: 27917373

53. Alhagamhmad MH, Day AS, Lemberg DA, Leach ST. An overview of the bacterial contribution to Crohn disease pathogenesis. J Med Microbiol. 2016; 65: 1049–1059. https://doi.org/10.1099/jmm.0.000331 PMID: 27501828

54. Brzuszkiewicz E, Thürmer A, Schuldes J, Leimbach A, Liesegang H, Meyer F-D, et al. Genome sequence analyses of two isolates from the recent Escherichia coli outbreak in Germany reveal the emergence of a new pathotype: Entero-Aggregative-Haemorrhagic Escherichia coli (EAHEC). Arch Microbiol. 2011; 193: 883–891. https://doi.org/10.1007/s00203-011-0725-6 PMID: 21713444