Clustering Metagenome Short Reads using Weighted Proteins

Gianluigi Folino[1], Fabio Gori[2], Mike S.M. Jetten[2], Elena Marchiori[2]*

[1] ICAR-CNR, Rende, Italy
[2] Radboud University, Nijmegen, The Netherlands

Abstract. This paper proposes a new knowledge-based method for clustering metagenome short reads. The method incorporates biological knowledge in the clustering process, by means of a list of proteins associated to each read. These proteins are chosen from a reference proteome database according to their similarity with the given read, as evaluated by BLAST. We introduce a scoring function for weighting the resulting proteins and use them for clustering reads. The resulting clustering algorithm performs automatic selection of the number of clusters, and generates possibly overlapping clusters of reads. Experiments on real-life benchmark datasets show the effectiveness of the method for reducing the size of a metagenome dataset while maintaining a high accuracy of organism content.

1 Introduction

The rapidly emerging field of metagenomics seeks to examine the genomic content of communities of organisms to understand their roles and interactions in an ecosystem. Given the wide-ranging roles microbes play in many ecosystems, metagenomics studies of microbial communities will reveal insights into protein families and their evolution. Because most microbes will not grow in the laboratory using current cultivation techniques, scientists have turned to cultivation-independent techniques to study microbial diversity.

At first shotgun Sanger sequencing was used to survey the metagenomic content, but nowadays massive parallel sequencing technology like 454 or Illumina, allow random sampling of DNA sequences to examine the genomic material present in a microbial community [5]. Using metagenomics, it is now possible to sequence and assemble genomes that are constructed from a mixture of organisms.

While it is common to refer to the genome sequence as if it were a single, complete and contiguous DNA string, it is in fact an assembly of billions of small, partially overlapping DNA fragments.

For a given sample, one would like to determine the phylogenetic provenance of the obtained fragments, the relative abundance of its different members, their metabolic capabilities, and the functional properties of the community as

* Contact author: elenam@cs.ru.nl

a whole. To this end, computational analyses are becoming increasingly indispensable tools [11,13].

Sophisticated computer algorithms (assemblers and scaffolders) merge these DNA fragments into contigs, and place these contigs into sequence scaffolds using various methods and tools (cf., e.g., [2,6,4]). Clustering methods are used for rapid analysis of sequence diversity and internal structure of the sample [8], for discovering protein families present in the sample [3], and as a pre-processing set for performing comparative genome assembly [12], where a reference closely related organism is employed to guide the assembly process.

In this paper we focus on the problem of clustering metagenome short reads [3,8].

Our approach is inspired by a recent work by Dalevi et al [3], where a method for clustering reads is proposed based on a set of proteins, called *proxygenes*, obtained by BLASTx of the reads against the protein sequences of a reference database. However, the results of the method proposed in [3] depend on the read selected at the beginning of the procedure.

We propose a new robust method for clustering metagenome short reads based on weighted proteins. The method generates a set of clusters, where each cluster is represented by one *proxygene*. This method has the following desirable features:

- it incorporates biological knowledge in the clustering process,
- it performs automatic selection of the number of clusters,
- it generates possibly overlapping clusters of reads.

Specifically, the proposed method consists of three main steps.

First, it uses a specialized version of BLAST (Basic Local Alignment Search Tool), called BLASTx, for associating a list of hits to each read. Each hit consists of one protein, two score values, called bit and identities, which measure the quality of the read-protein matching, and one confidence value, called E-value, which amounts to a confidence measure of the matching.

Next, a maximum of K proteins for each read, among those having E-value smaller than a given threshold α are selected. The selected proteins are weighted by means of a novel measure based on the bit- and identity- scores, which assigns small weights to proteins of high average quality.

Finally, the reads are clustered by translating the clustering problem into an instance of the weighted set covering problem (`WSC`). The `WSC` is a popular constrained optimization problem used in many real-life applications. Given a set of weighted columns and a set of rows, where each row is covered by at least one column, the `WSC` problem amounts to find a set of columns covering all the rows and having minimum total weight. In our context, columns are proteins and rows are reads. A protein covers a read if it belongs to the set of the selected hits of that read. We employ a publicly available fast heuristic algorithm for the weighted set covering problem [10]. The resulting clustering method generates a set of clusters, where each cluster is represented by one protein called *proxygene*.

In order to assess the effectiveness and benefits of the proposed clustering method, we consider simulated metagenomic datasets recently introduced in [3].

We measure the quality of the resulting clusters by means of the organism content of the clusters [8], their size, number and overlapping.

Specifically, we analyze the behavior of the clustering algorithm when varying its parameters K and α. Results show that the number of clusters decreases when bigger values of K are chosen while their overlapping increases. The organism content of the clusters does not change substantially for higher values of K and small α (0.01), indicating the effectiveness of the proposed approach in reducing the size of a metagenome dataset while maintaining a high accuracy of organism content.

The proposed method can therefore be used for reducing the size of the dataset while maintaining accuracy of functional and taxonomic content of a metagenome, and for discovering knowledge related to the protein content and the taxonomic organization of the organisms contained in the sample.

In general, the results substantiate the effectiveness of the proposed clustering method for mining metagenomic datasets.

2 Clustering Metagenome Short Reads

Clustering analysis for metagenomics amounts to group similar partial sequences, such as raw sequence reads, or candidate ORF (Open Reading Frame) sequences generated by an assembly program into clusters in order to discover information about the internal structure of the considered dataset, or the relative abundance of protein families. Different methods for clustering analysis of metagenomic datasets have been proposed, which can be divided into two main approaches. Sequence- and evidence-based methods. *Sequence-based* methods compare directly sequences using a similarity measure either based on sequence overlapping [8] or on extracted features such as oligonucleotide frequency [2]. *Evidence-based* methods employ knowledge extracted from external sources in the clustering process, like proteins identified by a BLASTx search (proxygenes) [3].

Here we use the latter approach for clustering short reads. Specifically, we propose a clustering method consisting of the following main steps:

1. Run BLASTx on the reads;
2. Assign weights to proteins resulting from BLASTx;
3. Cluster the reads using the weighted proteins obtained from the previous step as candidate cluster prototypes.

The result is a set of possibly overlapping clusters of reads, where each cluster is represented by a protein. The number of clusters is automatically determined by the algorithm. The proposed method has just two parameters: the maximum number K of hits selected for each read, and the E-value threshold α. Below the steps of the method are described in detail.

3 Run BLASTx on the Reads

The knowledge used by the proposed clustering algorithm is extracted by a reference proteome database by matching reads to that database by means of

BLASTx, a powerful search program. BLASTx belongs to the BLAST (Basic Local Alignment Search Tool) family, a set of similarity search programs designed to explore all of the available sequence databases regardless of whether the query is protein or DNA [7,9]. BLASTx is the BLAST program designed to evaluate the similarities between DNA sequences and proteins; it compares nucleotide sequence queries dynamically translated in all six reading frames to peptide sequence databases. The scores assigned in a BLAST search have a statistical interpretation, making real matches easier to distinguish from random background hits. In the following we summarize the main features of BLAST.

BLAST uses a heuristic algorithm that seeks local as opposed to global alignments and is therefore able to detect relationships among sequences that share only isolated regions of similarity [1]. When a query is submitted, BLAST works by first making a look-up table of all the *words* (short subsequences, three letters in our case) and *neighboring words*, i.e., similar words in the query sequence. The sequence database is then scanned for these strings; the locations in the databases of all these words are called *word hits*. Only those regions with word hits will be used as alignment seeds. When one of these matches is identified, it is used to initiate gap-free and gapped extensions of the word. After the algorithm has looked up all possible words from the query sequence and extended them maximally, it assembles the statistically significant alignment for each query-sequence pair, called *High-scoring Segment Pair* (HSP).

The matching reliability is evaluated trough *Bit Score*, S_B, and *E-value*, E. The *bit score* of one HSP is computed as the sum of the scoring matrix values for that segment pair. The *E-value* is the number of times one might *expect* to see such a query-sequence match (or a better one) merely by chance.

Another score very important for BLASTx is *Identities score*, defined as the proportion of the amino-acids in the database sequence that are matched by the amino-acids translation of the current query frame.

We refer to [7,9] for a formal description of these measures.

In our method, the E-value is used to constrain the number of output hits, while the bit and identities scores are used to weight proteins as follows.

4 Assign Weights to Proteins

From each hit that BLASTx outputs for a given read r, we extract a 4-dimensional vector $h = (p, S_B, Id, E)$ where p is the matched protein, S_B the bit score, Id the identities score, and E the E-value of that match. With abuse of notation we refer to such a vector as *hit* of r.

For a read r let $Hit_{\alpha,r}$, the sequence, sorted in increasing order of E-values, of its hits having E-value smaller than a given threshold α. Denote by $r_1, \ldots, r_m$ the set of reads r with non-empty $Hit_{\alpha,r}$. Let K_{α,r_i} be the sequence of the first K elements of Hit_{α,r_i} (the entire sequence if K exceeds the length). We write K_i instead of K_{α,r_i} when no ambiguity arises.

Let $P = \{p_1, \dots, p_n\}$ be the set of proteins occurring in $\cup_{i=1}^{m} K_i$. For each protein $p \in P$ define the set

$$H_p := \{h \in \cup_{i=1}^{m} K_i \mid h(1) = p\}$$

where $h(1)$ denotes the first component of the hit vector h. Thus H_p consists of the selected hits containing p.

Define the weight of p as follows.

$$w_p := 1 + \ulcorner \frac{1}{|H_p|} \sum_{h \in H_p} (100 \frac{\texttt{max\_score} - S_B(h)}{\texttt{max\_score} - \texttt{min\_score}} + 100 - \mathit{Id}(h)) \urcorner,$$

where $\ulcorner v \urcorner$ denotes the smallest integer bigger or equal than v, $S_B(h)$ and $\mathit{Id}(h)$ the bit- and identity-score of h, respectively, and $|H_p|$ the cardinality of H_p. By construction weights are positive integers between 1 and 201.

The bit score has been used e.g. in [3] to define a measure of protein relevance.

Our approach for scoring proteins differs from e.g. the one used in [3] in two main ways. First, we score proteins using also the identities score. Second, we score each protein globally, by considering all the hits involving that protein, while in [3] proteins are score locally. In the latter approach, first reads and proteins are clustered together, and only at the end of the clustering process, each protein of a cluster is scored by means of the cumulative bit score of its alignment to the reads within the same cluster.

5 Clustering Reads using Weighted Proteins

The clustering algorithm selects a set of cluster representatives from P, whose union covers all the considered reads, and with minimum total weight. The clusters are generated by a fast heuristic algorithm for the weighted set covering problem [10] applied to the m selected reads and the set P of proteins weighted as described above. The number of clusters is automatically computed by the procedure.

Formally, consider the matrix $a \in \{0,1\}^{m \times n}$ such that

$$a_{ij} = \begin{cases} 1 & \text{if column } j \text{ covers row } i \\ 0 & \text{otherwise.} \end{cases}$$

and the vector w of protein weights w_j, $j = 1, \dots, n$ positive integers. So a row i is covered by a column j if a_{ij} is equal to 1. In the context of our application $a_{ij} = 1$ if protein p_j occurs in the set K_i of selected hits of read r_i.

The *weighted set covering problem* (WSC, in short) can be formulated as a constrained optimization problem as follows.

$$\min_{x \in \{0,1\}^n} \sum_{j=1}^{n} x_j w_j, \qquad \text{such that } \sum_{j=1}^{n} a_{ij} x_j \geq 1, \qquad \text{for} \quad i = 1, \dots, m. \tag{WSC}$$

The variable x_j indicates whether column j belongs to the solution ($x_j = 1$) or not ($x_j = 0$).

The m constraint inequalities are used to express the requirement that each row i be covered by at least one column (that is, for each read r_i, at least one protein in K_i is chosen). The weight w_j specifies the cost of column j.

The weighted set covering problem is one of the oldest and best studied NP-hard problems. It has been successfully employed to tackle real-life problems in diverse domains, including biology (cf., e.g., [14]). Here we use a fast heuristic algorithm for `WSC`[1] originally developed for tackling airline crew scheduling problems [10].

A solution corresponds to a subset of P consisting of those proteins p_j such that $x_j = 1$. Each of the selected proteins is a *proxygene*. It represents a cluster containing those reads r having that protein in K_r.

Example 1. We illustrate this process by means of a toy example (cf. Figure 1). Suppose given a set of five reads $\{r_1, \ldots, r_5\}$ and suppose that the proteins occurring in the selected hits of these reads are:

- $\{p_1, p_3, p_5\}$ for read r_1, $\{p_1, p_3, p_6\}$ for read r_2,
- $\{p_2, p_5\}$ for read r_3,
- $\{p_2\}$ for read r_4, and
- $\{p_2, p_3, p_6\}$ for read r_5.

Assume for the sake of simplicity that all proteins have equal weight. Then Figure 1 (left part) shows the corresponding 5-row, 6-column matrix a_{ij}. The WSC-clustering algorithm applied to this problem instance outputs the set $\{p_2, p_3\}$ of columns, having total weight equal to 2 (see Figure 1 right part). The selected columns correspond to the two clusters $\{r_3, r_4, r_5\}$ and $\{r_1, r_2, r_5\}$, respectively.

6 Experiments

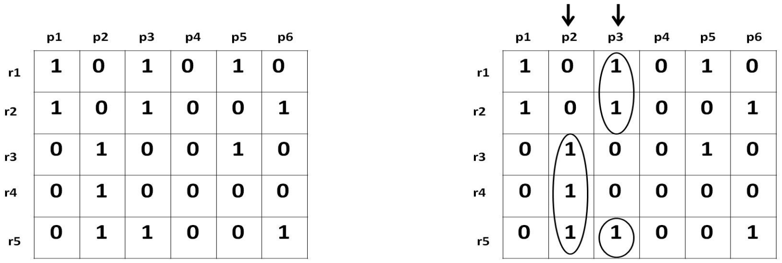
We consider three complex metagenome datasets introduced in [3], called in the following M1, M2 and M3. These datasets consist of reads from 9, 5 and 8 genome projects, sequenced at the Joint Genome Institute (JGI) using the 454 GS20 pyrosequencing platform that produces $\sim$ 100 bp reads. From each genome project, reads are sampled randomly at coverage level $0.1X$. The coverage is defined as the average number of times a nucleotide is sampled. This resulted in a total of 35230, 28870 and 35861 reads, respectively.

Table 1 shows the names of the organisms and the number of reads generated for the M1 dataset. The reader is referred to [3] for a detailed description of all the datasets.

In our experiments we use the `NR`[2] (non-redundant) protein sequence database as reference database for BLASTx. The parameters of the external software we

[1] Publicly available at `http://www.cs.ru.nl/~elenam`

[2] Publicly available at ftp://ftp.ncbi.nlm.nih.gov/blast/db.



	p1	p2	p3	p4	p5	p6
r1	1	0	1	0	1	0
r2	1	0	1	0	0	1
r3	0	1	0	0	1	0
r4	0	1	0	0	0	0
r5	0	1	1	0	0	1

	p1	p2	p3	p4	p5	p6
r1	1	0	1	0	1	0
r2	1	0	1	0	0	1
r3	0	1	0	0	1	0
r4	0	1	0	0	0	0
r5	0	1	1	0	0	1

Fig. 1. Left: input covering matrix; position (i,j) contains a 1 if protein p_j occurs in the set of selected hits of read r_i, otherwise it contains a 0. Right: the proteins selected by the `WSC`-clustering algorithm are indicated by arrows.

Id.	Organism	genome size (bp)	reads sampled
a	`Clostridium phytofermentans ISDg`	4 533 512	4638
b	`Prochlorococcus marinus NATL2A`	1 842 899	1866
c	`Lactobacillus reuteri 100-23`	2 174 299	2371
d	`Caldicellulosiruptor saccharolyticus DSM 8903`	2 970 275	2950
e	`Clostridium sp. OhILAs`	2 997 608	2934
f	`Herpetosiphon aurantiacus ATCC 23779`	6 605 151	6937
g	`Bacillus weihenstephanensis KBAB4`	5 602 503	4158
h	`Halothermothrix orenii H 168`	2 578 146	2698
i	`Clostridium cellulolyticum H10`	3 958 683	3978

Table 1. Characteristics of the organisms used in the experiments: the identifier and name of the organism, the size of its genome and the total number of reads sampled (M1 dataset).

used are set as follows. For BLASTx the default parameters were used. In all experiments WSCP was run with pre-processing ($-p$), number of iterations equal to 1000 ($-x1000$), one tenth of the best actual cover used as starting partial solution ($-a0.1$), and 150 columns to be selected for building the initial partial cover at the first iteration ($-b150$). For lack of space, we refer to [10] for a detailed description of the WSCP program.

6.1 Evaluation

First we set α to a reasonable value, equal to 0.01, which amounts to remove reads with E-value greater than α, resulting in the selection of 21236 reads for M1, 21064 for M2 and 24043 for M3. We analyze the clusterings obtained by varying the value of K by means of the following characteristics.

- The number of clusters obtained, their size and overlapping.
- The reduction factor, defined as the number of selected reads divided by the number of clusters.

- The homogeneity of the clusters as measure by the so-called *cluster purity*, defined as the maximum fraction of its elements belonging to the same organism, that is

  $$purity(C) := \frac{1}{|C|} \max_{i=1,\ldots,n_{org}} (|C|_{organism=i}),$$

  where $|C|_{organism=i}$ denotes the number of elements of cluster C belonging to organism i, and n_{org} the number of organisms.

A similar analysis is performed by fixing K to a reasonable value, equal to 50, and varying the threshold α.

K	1	2	10	50	1000
number of proteins selected	13594	19967	66005	174110	360578
number of clusters	13594	13197	12599	12091	11763
number of singleton clusters	9003	8334	7420	6666	6145
maximum size of clusters	17	21	23	28	32
total size of overlapping	0	273	877	1640	2979

Table 2. Summary of the results of experiments for α=0.01 and varying K (M1 dataset).

Figure 2 shows in more detail the trends of the cluster homogeneity and of the reduction factor.

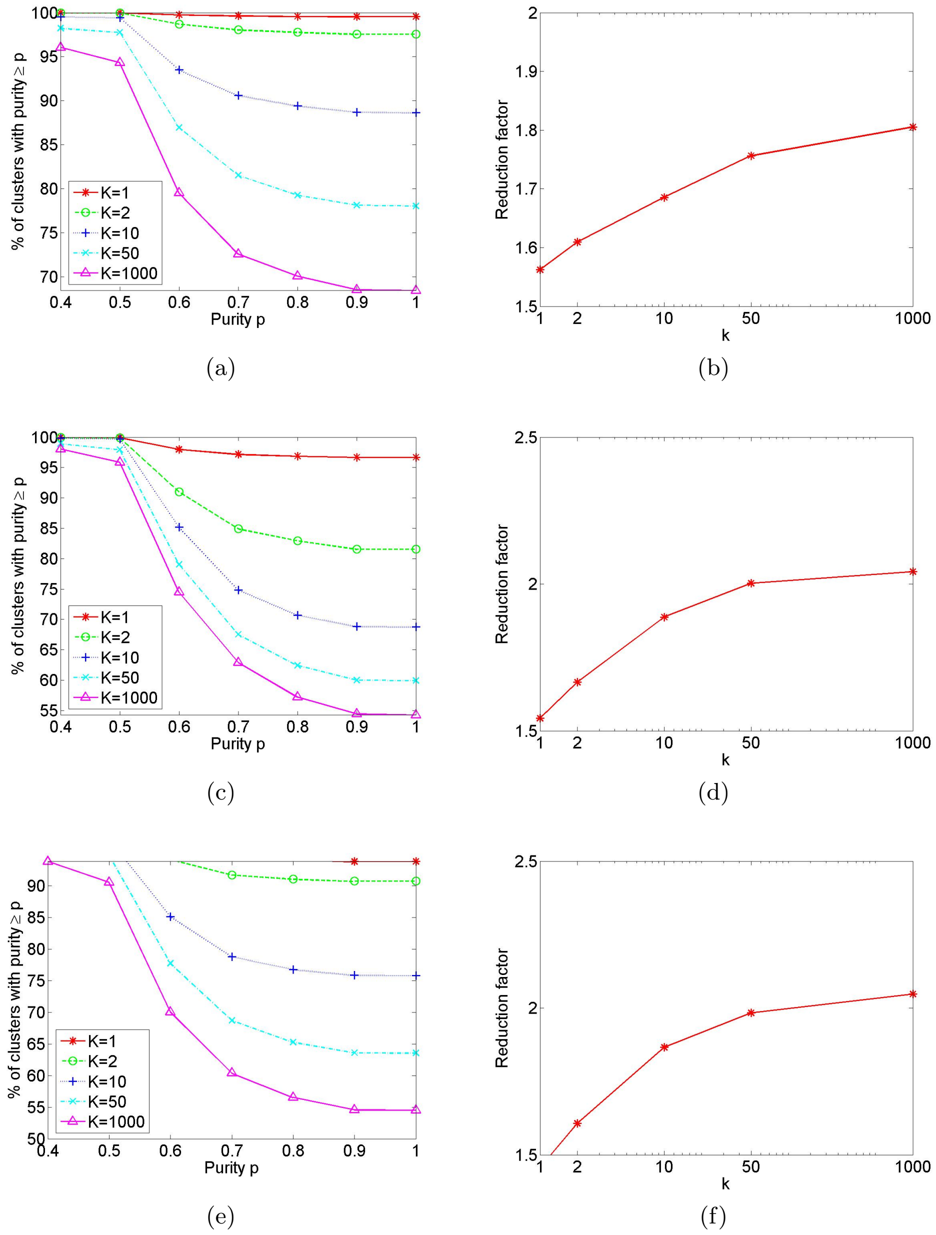
6.2 Results: fixed value of α=0.01 and varying K values

Table 2 summarizes the results with this parameter setting for dataset M1. The number of selected proteins increases when K increases, while the number of clusters decreases, indicating effectiveness of the method to select few proxy-genes. Furthermore, the number of singleton clusters also decreases for higher values of K, indicating a stronger bias towards the grouping of reads. A similar trend can be observed for datasets M2 and M3 (results omitted for lack of space).

Figures 2 (a, c and f) show the percentage of non-singleton clusters having purity greater or equal than a given value p, for selected values of p in $[0.4, 1]$. For all the datasets, the curve at $K = 1$ dominates all other ones, justified by the fact that the corresponding clustering contains many clusters of small size, which are likely to have higher purity. For instance, for the M1 dataset, about 75% and 35% of the clusters have size equal to 2, for $K = 1$ and 1000, respectively.

Figures 2 (b,d and f) show the reduction factor for different values of K. As expected, a larger value of K results into a higher reduction factor.

Finally, Figure 3 shows how the number of reads occurring in more than k clusters varies for different choices of K (M1 dataset). For a small value of K (equal to 5) a read occurs in at most 3 clusters, while for a very high value of



(a) (b)

(c) (d)

(e) (f)

Fig. 2. a,c,e) plot of percentage of non-singleton clusters with purity ≥ p for different values of K (M1, M2 and M3). b,d,f) plot of the reduction factor for different values of K (M1, M2 and M3).

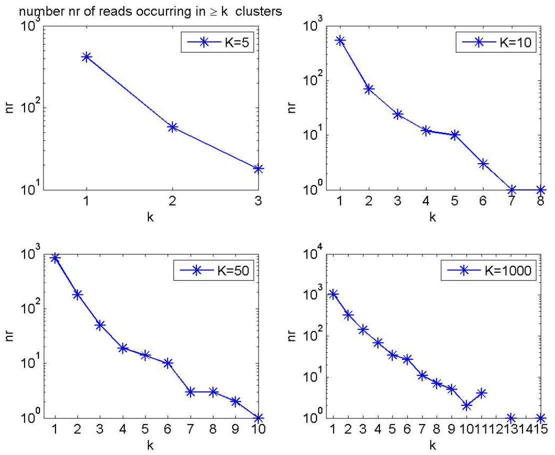


Fig. 3. Plot of the number nr of reads occurring in k clusters (M1 dataset).

K (equal to 1000) a read occurs in at most 15 clusters. Indeed, as reported in Table 2, the total overlapping shows a substantial increase for high values of K, where by total overlapping we mean the sum of cardinality of the clusters minus the number (21236) of selected reads. These results can be justified by the fact that K is an upper bound on the maximum number of clusters one read may belong to. Similar results, not showed for lack of space, were obtained using M2 and M3 datasets.

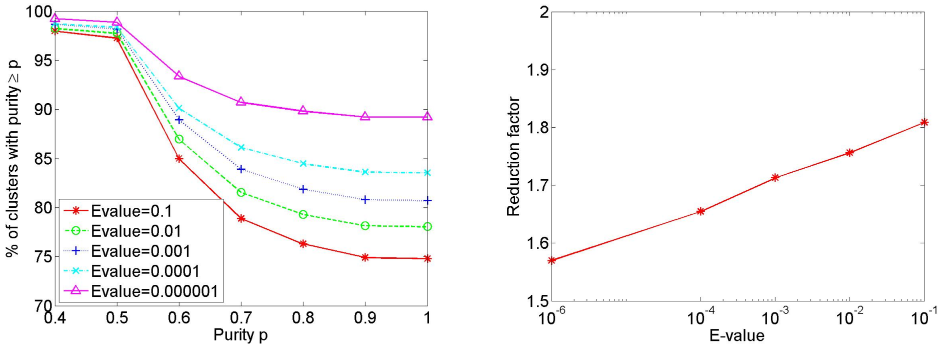
6.3 Results: fixed value of $K = 50$ and varying α values

α	0.1	0.01	0.001	0.0001	1e-006
number of reads selected	22219	21236	20300	19085	16736
number of proteins selected	208443	174110	146524	116682	72149
number of clusters	12283	12091	11850	11534	10660
number of singleton clusters	6464	6666	6772	6889	6801
maximum size of clusters	30	28	27	23	19
total size of overlapping	2026	1640	1326	1066	528

Table 3. Summary of the results of experiments for $K = 50$ and varying α (M1 dataset).

Table 3 summarizes the results with this parameter setting for dataset M1. Higher values of α result into the selection of an higher number of reads and of proteins. Moreover, clusters of bigger size and overlapping are obtained.

The plots of Figure 4 show that on the M1 dataset, small α values lead to clusterings where 90% of the clusters are very accurate, in terms of organism content, and a reduction factor of about 1.6. For higher values of α clusters purity decreases reaching a minimum of about 75%, while reduction factor increases reaching a maximum of about 1.8. Similar results are obtained for datasets M2 and M3. Thus, the user can decide a tradeoff between purity and reduction, depending on the specific research question to be addressed.



(a) (b)

Fig. 4. a) plot of percentage of non-singleton clusters with purity $\geq p$ for different values of α. b) plot of the reduction factor for different values of α (M1 dataset).

7 Conclusion and Future Work

This paper introduced a new evidence-based method for clustering metagenome short reads and analyzed its performance on benchmark metagenome datasets. Results indicated effectiveness of the proposed method as a tool for mining metagenome data.

We focussed on the experimental analysis of the two parameters of the proposed clustering method, K and α. As for the computational cost, the `WSC`-clustering algorithm is very efficient, due to the fast heuristic employed to search for an optimal set cover. However, the extraction of the hits from the initial dataset of reads is computationally expensive. Nevertheless, the latter process can be parallelized by partitioning the reads and running BLASTx independently on each group of the partition.

In the future, we intent to investigate in more depth the biological meaning of the resulting clusters, in particular their functional and taxonomic content, in order to discover knowledge related to the protein content and the taxonomic organization of the organisms contained in metagenomes.

Furthermore, it is interesting to investigate if the clusterings obtained by varying the value of such parameters could be used for analyzing the dynamics of organism grouping, as modeled by the protein-based clustering, in particular whether such model of organism-grouping dynamics is related to the taxonomic evolution of the corresponding metagenome sample.

Acknowledgements

We would like to thank Mavrommatis Konstantinos for providing the datasets in [3] as well as useful information about such data.

References

1. S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J Molecular Biology*, 215(3):403–10, 1990.
2. C.K. Chan, A.L. Hsu, S. Tang, and S.K. Halgamuge. Using growing self-organising maps to improve the binning process in environmental whole-genome shotgun sequencing. *Journal of Biomedicine and Biotechnology*, 2008.
3. D. Dalevi, N. N. Ivanova, K. Mavromatis, S. D. Hooper, E. Szeto, P. Hugenholtz, N. C. Kyrpides, and V. M. Markowitz. Annotation of metagenome short reads using proxygenes. *Bioinformatics*, 24(16), 2008.
4. K. Mavromatis et al. Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. *Nature Methods*, 4(6):495–500, 2007.
5. S. Yooseph et al. The sorcerer ii global ocean sampling expedition: Expanding the universe of protein families. *PLoS Biol*, 5(3):432–466, 2007.
6. D. Hernandez, P. Francois, L. Farinelli, M. Osteras, and J. Schrenzel. De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer. *Genome Research*, 18(5):802–809, 2008.
7. I. Korf, M. Yandell, and J. Bedell. *BLAST*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2003.
8. Weizhong Li, John C. Wooley, and Adam Godzik. Probing metagenomics by rapid cluster analysis of very large datasets. *PLoS ONE*, 3(10), 2008.
9. T. Madden. *The BLAST Sequence Analysis Tool*, chapter 16. Bethesda (MD): National Library of Medicine (US), 2002.
10. E. Marchiori and A. Steenbeek. An evolutionary algorithm for large scale set covering problems with application to airline crew scheduling. In *Real World Applications of Evolutionary Computing*, volume LNCS 1083, pages 367–381, 2000.
11. A.C. McHardy and I. Rigoutsos. Whats in the mix: phylogenetic classification of metagenome sequence samples. *Current Opinion in Microbiology*, 10:499503, 2007.
12. M. Pop, A. Phillippy, A.L. Delcher, and S.L. Salzberg. Comparative genome assembly. *Briefings in Bioinformatics*, 5(3):237–248, 2004.
13. J. Raes, K. U. Foerstner, and P. Bork. Get the most out of your metagenome: computational analysis of environmental sequence data. *Current Opinion in Microbiology*, 10:490–498, 2007.
14. W. Zhao, M.L. Fanning, and T. Lane. Efficient RNAi-based gene family knockdown via set cover optimization. *Artificial Intelligence in Medicine*, 35(1-2):61–73, 2005.