View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Radboud Repository

Radboud Repository

Radboud University Nijmegen

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a preprint version which may differ from the publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/75597

Please be advised that this information was generated on 2017-12-06 and may be subject to change.

Evidence-based Clustering of Reads and Taxonomic Analysis of Metagenomic Data

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. 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. In this paper we focus on clustering methods and their application to taxonomic analysis of metagenomic data. Clustering analysis for metagenomics amounts to group similar partial sequences, such as raw sequence reads, 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. Here we focus on evidence-based methods for clustering that employ knowledge extracted from proteins identified by a BLASTx search (proxygenes). We consider two clustering algorithms introduced in previous work and a new one. We discuss advantages and drawbacks of the algorithms, and use them to perform taxonomic analysis of metagenomic data. To this aim, three real-life benchmark datasets used in previous work on metagenomic data analysis are used. Comparison of the results indicate satisfactory coherence of the taxonomies output by the three algorithms, with respect to phylogenetic content at the class level and taxonomic distribution at phylum level. In general, the experimental comparative analysis substantiates the effectiveness of evidence-based clustering methods for taxonomic analysis of metagenomic data.

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

* Contact author: elenam@cs.ru.nl

community [4]. Using metagenomics, it is now possible to sequence and assemble genomes that are constructed from a mixture of organisms.

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 a whole. To this end, computational analysis is becoming increasingly indispensable [11,13]. In particular, 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 clustering methods and their application to taxonomic analysis of metagenomic data. 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]. In this paper we focus on the latter approach for clustering short reads.

We consider two clustering algorithms introduced in previous work [3,5] and a refinement of the latter one based on ensemble techniques. These algorithms cluster reads using weighted proteins as evidence. Such proteins are obtained by a specialized version of BLAST (Basic Local Alignment Search Tool), called BLASTx, which associates a list of *hits* to one 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 between the read and the protein.

Specifically, in [3] an algorithm, here called `LWproxy` (Local Weight proxy), is introduced, that clusters reads and those proteins in their sets of hits simultaneously, in such a way that one cluster of proteins is associated to one cluster of reads. Then it assigns one *local* weight to each protein of a cluster, using the cumulative BLASTx bit score of those reads in corresponding cluster having that protein as one of their hits. The protein with best weight (highest cumulative bit score) is selected as *proxygene* of the cluster of reads.

In [5], an alternative method for clustering metagenome short reads based on weighted proteins is proposed, here called `GWproxy` (Global Weight proxy). The method first assigns *global* weights to each protein using the BLASTx identity and bit score of those reads having that protein as one of their hits. Next, the method groups reads into clusters using an instance of the weighted set covering problem, with reads as rows and proteins as columns. It seeks the smallest set of columns covering all rows and having best total weight. A solution corresponds to a clustering of reads and one protein (proxygene) associated to each cluster.

While in [3] the proxygene of a cluster is selected within a set of proteins associated to that cluster, in `GWproxy` clustering and proxygene selection are performed at the same time.

In this paper we introduce a refinement of `GWproxy` based on the following ensemble technique, called `EGWproxy` (Ensemble Global Weight proxy). The algorithm associates a list of proteins to each cluster resulting from application of `GWproxy`, such that each protein occurs as hit of each of the reads of that cluster. Such a list is used for refining the biological analysis of the cluster, for instance by assigning a taxonomic identifier (*taxID*) by means of weighted majority vote among the taxID's of the proteins in the associated list.

We discuss advantages and drawbacks of the above clustering algorithms, and use them to perform taxonomic analysis of metagenomic data. To this aim, three real-life benchmark datasets used in previous work on metagenomic data analysis are used. These datasets were introduced in [3] and used to perform a thorough analysis of evidence-based direct- and indirect (that is, using proxygenes) annotation methods for short metagenomic reads. The results of such analysis substantiated advantages and effectiveness of indirect methods over direct ones.

The results of the three considered evidence-based clustering algorithms indicate satisfactory coherence of the taxonomies output by the algorithms, with the `GWproxy` based algorithms yielding taxonomic content closer to that of the metagenome data. In general, the experimental comparative analysis substantiates the effectiveness of evidence-based methods for taxonomic analysis of metagenomic data.

2 Clustering Metagenome Short Reads using Proxygenes

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 consider the latter approach for clustering short reads.

The knowledge used by the clustering algorithms here considered 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.

2.1 The BLAST alignment method

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 of read r and protein p is evaluated trough *Bit Score*, denoted by $S_B(r, p)$, and *E-value*, denoted by 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 important BLASTx score of matching between r and p is *Identities score*, denoted by $Id(r, p)$, 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] for a formal description of these measures.

We turn now to describe the three methods here used for taxonomic analysis of metagenomic data. Here and in the sequel we assume the BLASTx has been applied to a metagenomic data set with a given Evalue cutoff value. We denote by $R = \{r_1, \ldots, r_m\}$ the resulting set of reads having at least one BLASTx hit for the given cutoff, and by $P = \{p_1, \ldots, p_n\}$ the set of proteins occurring in the hit of at least one read of R.

2.2 `LWproxy`

`LWproxy` generates a collection of pairs (C_i, P_i), where C_i is a set of reads and P_i a set of proteins. The algorithm can be summarized as follows.

1. Set $i = 0$.
2. $X = R$.
3. If X is empty then terminate, otherwise set $i = i + 1$.

4. Select randomly[1] one read r_i from X as seed of cluster $C_i = \{r_i\}$.
5. Set P_i to the set of hits of r_i.
6. Remove r_i from X.
7. Add to C_i all reads having one element of P_i as a best hit, and remove them from X.
8. Add to P_i all hits of those reads added to C_i in the previous step.
9. If no reads are added then go to step 3, otherwise go to step 7.

When the clustering process is terminated, the method assigns one proxygene to each C_i by selecting from P_i the protein having highest cumulative bit-score.

Example 1. Suppose given a set of five reads $\{r_1, \ldots, r_5\}$ and suppose that the proteins occurring in their hits:

- $\{p_1, p_3, p_5\}$ for read r_1, with best hit p_3.
- $\{p_1, p_3, p_5\}$ for read r_2, with best hit p_3.
- $\{p_2, p_4\}$ for read r_3, with best hit p_2.
- $\{p_2\}$ for read r_4, with best hit p_2.
- $\{p_2, p_3, p_5\}$ for read r_5, with best hit p_2.

Suppose `LWproxy` selects r_5 as seed of the first cluster C_1. Then it adds all the other reads to C_1, since their best hit is in the list of hits of r_5. P_1 becomes equal to the entire set of proteins. Suppose for simplicity that all proteins have equal bit-score. Then `LWproxy` selects p_2 as proxygene, since it has highest cumulative bit-score.

2.3 `GWproxy`

While `LWproxy` constructs clusters incrementally, `GWproxy` searches for clusters in a given search space, consisting of clusters characterized by the proteins as follows. We say that a protein *covers* a read if the protein occurs as one of the hits of that read. Then each protein characterizes one cluster, consisting of the reads it covers. Moreover, we can assign to each protein a *global* weight, representing the cost of selecting that protein as cluster representative. The weight of protein p is defined as

$$w(p) = 1 + \left\lceil \frac{1}{N_p} \sum_{r \mid p \textit{ hit of } r} \left(100 \frac{\mathtt{max} - S_B(r,p)}{\mathtt{max} - \mathtt{min}} + 100 - Id(r,p)\right) \right\rceil,$$

where $\lceil v \rceil$ denotes the smallest integer bigger or equal than v, and N_p is the number of reads having p as one of their hits. The maximum and minimum value of S_B over the considered pairs of reads and proteins, `max` and `min`, respectively, are used to scale $S_B(r,p)$. The weight is such that better proteins have smaller w value (smaller cost).

[1] We consider here random seed selection. However, in [3] the criterion for selecting a seed is not specified.

Clustering then amount at finding a minimum set of proteins in R that, together, cover all the reads in R and have minimum total cost. Formally, consider the vector of protein weights $w \in \mathbb{N}^n$ and the matrix $A \in \{0,1\}^{m \times n}$ whose elements a_{ij} are such that

$$a_{ij} = \begin{cases} 1, & \text{if } p_j \text{ covers } r_i, \\ 0, & \text{otherwise.} \end{cases}$$

We want to solve the following constrained optimization problem (*weighted set covering problem* (WSC, in short)).

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

The variable x_j indicates whether p_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 read r_i be covered by at least one protein. The weight w_j specifies the cost of protein p_j.

Here a fast heuristic algorithm[2] for WSC [10] is applied to find a solution. 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 the cluster consisting of those reads covered by that protein.

Example 2. We illustrate the application of GWproxy on the toy problem of Example 1. 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}. Application of GWproxy outputs proteins p_2, p_3 (see Figure 1 right part). The selected proteins correspond to the two clusters of reads $\{r_3, r_4, r_5\}$ and $\{r_1, r_2, r_5\}$, with p_2 and p_3 as associated proxygenes, respectively.

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

	p1	↓ p2	↓ p3	p4	p5
r1	1	0	1	1	0
r2	1	0	1	0	1
r3	0	1	0	1	0
r4	0	1	0	0	0
r5	0	1	1	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: proxygenes selected by the GWproxy are indicated by arrows.

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

2.4 EGWproxy

The aim of this algorithm is to refine the clustering produced by GWproxy as follows. Each cluster that GWproxy outputs is represented by one protein. However, because of the short length of reads, and because in general the size of clusters is not very big (see analysis in [5]), it may well happen that more than one protein covers all the reads of a cluster. All such proteins can be considered as equivalent representative of that cluster. However, GWproxy selects only one of such proteins, among those having best score.

It is biologically meaningful to consider the information contained in all such proteins when performing protein family and taxonomic analysis of that cluster. To this aim, for each cluster, EGWproxy outputs the maximum set of proteins such that, each protein in that set covers all reads of that cluster. Biological analysis of that cluster can then be performed by means of ensemble techniques. For instance, for performing taxonomic analysis of cluster C, the following criterion can be used in order to decide which taxID to associate to C. The set T of taxID's of the list of proteins that EGWproxy associates to C is considered. Then the final taxID t_{fin} of C is computed as

$$t_{fin} = \arg_{t \in T} \min \sum_{p \text{ with taxID equal to } t} \frac{w_p}{N_p}.$$

Figure 2 shows the output of EGWproxy on one of the datasets used in our experiments (M3, see Section 3): it plots the list (of proteins) size (x-axis) versus the number of lists of that size (y-axis). Similar trends are obtained on the other two datasets considered in our experiments. On these datasets the length of the resulting protein lists remains 1 for more than half of the clusters, while in some cases it becomes rather big (this is more likely to happen for small clusters).

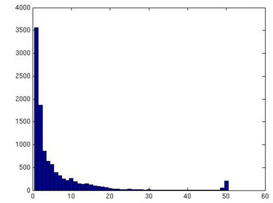

Fig. 2. Plot of size of protein lists (x-axis) versus number of lists of that size (y-axis) output by EGWproxy on dataset M3.

2.5 Comparison of Algorithms

GWproxy and LWproxy use different clustering heuristics: the first algorithm searches a clustering in a fixed searching space characterized by the sets of reads covered by each protein in R, while LWproxy constructs incrementally clusters of reads and of proteins. Furthermore, GWproxy scores proteins using bit and identities score, while LWproxy uses only bit score. Finally, GWproxy scores each protein globally, that is, using all the reads it covers, while LWproxy scores only the proteins of a cluster, where each protein is scored locally using the reads it covers that belong to that cluster.

Both EGWproxy and LWproxy associates to each cluster of reads one set of proteins. However, while LWproxy selects one protein as final representative of a cluster, EGWproxy employs an ensemble technique in order to exploit information of all the proteins of that set.

A drawback of LWproxy is that results may be affected by the choice of the read used in the first step of the algorithm, as illustrated by the following example.

Example 3. Consider the toy problem in Example 1. If LWproxy starts from r_1 as seed for C_1 then only r_2 is added to C_1, since r_2 (best hit of each of the other reads) does not occur in the list of protein associated to C_1. Then construction of a second cluster, say C_2 begins. C_2 is filled with the rest of the reads (r_3, r_4, r_5).

While in the experiments here conducted this drawback does not seem to affect the results, it remains to be investigated whether this drawback does not affect results in general.

A drawback of GWproxy is that it outputs only one solution, while in general there may be more "optimal" clustering of reads. This is because the weighted set covering problem seeks one optimal solution, not the set of all optimal solutions. EGWproxy tries to overcome this drawback by using a post-processing step, followed by the application of an ensemble technique for merging multiple solutions. However, the post-processing step acts only on the set of proteins, while the clusters of reads remain those produced by GWproxy. It remains to be investigated whether application of ensemble techniques also at the level of clusters of reads can improve the performance of the method.

3 Taxonomic Analysis of Metagenome Data

We consider three complex metagenome datasets introduced in [3], called in the following M1, M2 and M3. These datasets were generated, respectively, 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 were 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[3] (non-redundant) protein sequence database as reference database for BLASTx. The parameters of the external software we

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

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 we used Evalue cutoff $E = 10^{-6}$. Moreover, 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.

3.1 Results

We extract taxonomic information from each metagenome dataset as follows. For `LWproxy` and `GWproxy` each cluster of reads is represented by one protein. The taxID of such protein is used as taxonomic information of that cluster. For `EGWproxy` the list of proteins associated to each cluster is transformed into one taxID as described in Section 2.4.

In this way, the metagenomic data is transformed into a set of taxID's of proteins, one for each cluster of reads. Taxonomic information is then retrieved from the NCBI taxonomy (see `http://www.ncbi.nlm.nih.gov/Taxonomy/`). The NCBI Taxonomy database is a curated set of taxonomic classifications for all the organisms that are represented in GenBank. Each taxon in the database is associated with a numerical unique identifier called taxID. In the present analysis, the taxonomic information of these known proxygenes is used to determine the taxonomic content of the metagenomic data.

We visualize the resulting taxonomic information in two ways.

- Histogram of phylogenetic identities, as done e.g. in [6]. Shown are the percentages of the total of identifiable hits assigned to the phylogenetic groups obtained by means of the taxID of the proxygenes. Here analysis at the class taxonomic level is performed.
- Graph representation of taxonomic distribution of reads, as done e.g. in [14]. Here analysis at the taxonomic level of phylum and class is performed, where resulting taxa containing less than 10 reads are discarded.

We apply the above techniques to the proxygenes and taxid obtained from the considered algorithms, as well as to the known taxid's of the original metagenome data sets, provided by the producers of the benchmark data [3]. We use these latter results as "golden truth" (`GT` in short) to evaluate the methods.

Histograms of phylogenetic identities at the class level for the three data sets are shown in Figure 3. On dataset $M1$ `EGWproxy` achieves results most similar to `GT`. On $M2$ the three methods perform equally well, with results close to those of `GT`. On $M3$ there is a clear discrepancy in the percentages output by the three methods and `GT`, where `LWproxy` appears slightly closer to `GT` than the other methods.

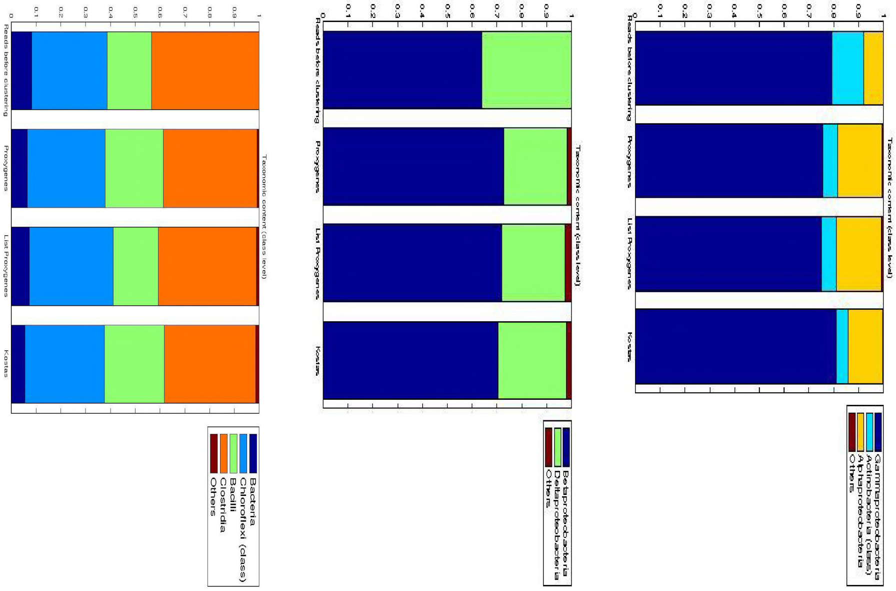

Fig. 3. Taxomonic distribution at taxonomic class level of the three datasets. From left to right: M1, M2 and M3. From top to bottom: "golden truth", `GWproxy`, `EGWproxy` and `LWproxy`.

For lack of space, we show graphs of the taxonomic distribution at phylum and class level only for $M1$ in Figure 4. Results indicate satisfactory consensus among the three methods, yielding similar type of graphs. The quality of results is satisfactory, with only one mismatching subtree. Indeed, the `GT` graph contains `Cyanobacteria` at phylum level, while all the graphs of the three clustering methods contain `Proteobacteria` at phylum level. This may be possibly justified by the fact that `Proteobacteria` is a phylum with more sequenced representatives than all other bacterial phylia combined [3]. However, a more thorough investigation of the reads assigned to this phylum is required, in order to check whether these reads are assigned to `Cyanobacteria` by `GT`. On $M2$, results show relative coherence of the taxonomic assignment of the three methods, and close similarity to `GT`. Indeed, at the phylum level, the three methods assign about 2% of the total amount of reads to an incorrect phylum (`Firmicutes`). On $M3$ the three methods assign about 0.1% of the total amount of reads to `Firmicutes`.

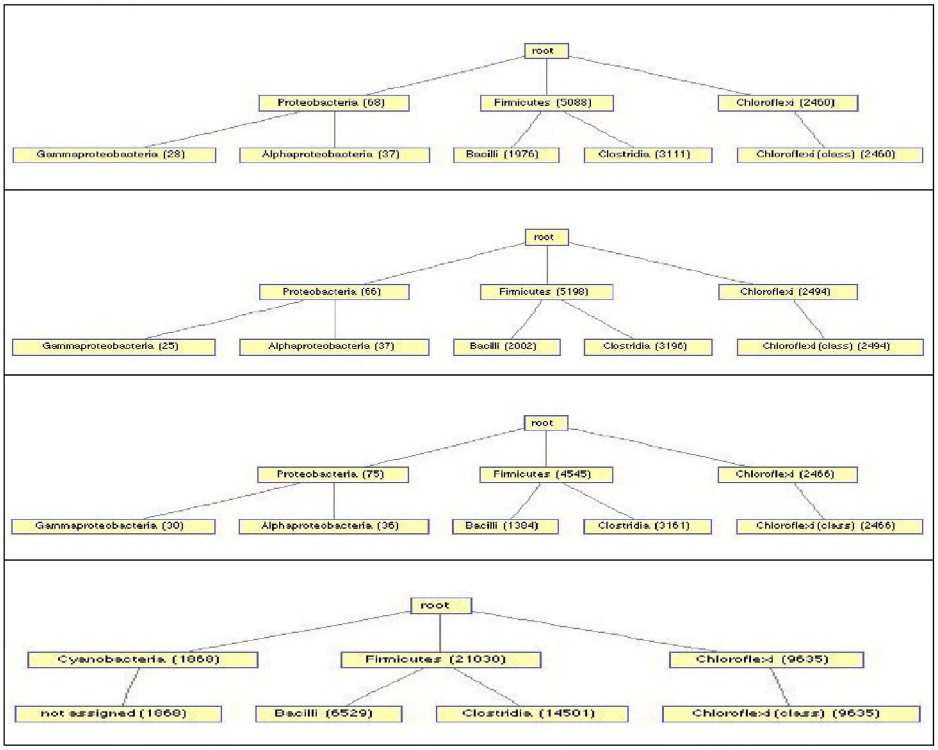

Fig. 4. Phylogenetic graph for M1. From top to bottom: `LWproxy`, `GWproxy`, `EGWproxy` and "golden truth".

4 Conclusion and Future Work

In this paper we compared three methods for clustering reads and their application to taxonomic analysis of metagenome data. We discuss advantages and drawbacks of the methods and applied them to perform taxonomic analysis of three real-life metagenome datasets with known taxonomic content. Results of such analysis indicate satisfactory consensus of all the three methods, and very good performance with respect to taxonomic distribution and phylogenetic content. In future work we intend to use the results of this investigation for designing even better clustering methods, in order to obtain fully reliable results. To this aim, we intend to introduce a statistical test for measuring significance of taxonomic assignment, in order to discard assignments possibly due to the composition of the reference proteome database used when applying BLASTx. Such a test will consider not only the number of reads assigned to a taxa, but also the divergence of their proxygenes as well as the nucleotide composition of the reads.

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. S. Yooseph et al. The sorcerer ii global ocean sampling expedition: Expanding the universe of protein families. *PLoS Biol*, 5(3):432–466, 2007.
5. G. Folino, F. Gori, M. Jetten, and E. Marchiori. Clustering metagenome short reads using weighted proteins. In *Proceedings of the Seventh European Conference on Evolutionary Computation, Machine Learning and Datamining in Bioinformatics*. Springer-Verlag, LNCS, 2009. In print.
6. Biddle J.F. and et al. Metagenomic signatures of the Peru margin subseafloor biosphere show a genetically distinct environment. *PNAS*, (105):10583–10588, 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. J.C. Venter and et al. Environmental genome shotgun sequencing of the sargasso sea. *Science*, (304):66–74, 2004.