Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Metagenomic reads binning with spaced seeds

Samuele Girotto, Matteo Comin *, Cinzia Pizzi *

Department of Information Engineering, University of Padova, Italy

ARTICLE INFO

Article history:
Received 23 February 2017
Received in revised form 16 May 2017
Accepted 21 May 2017
Available online xxxx

Keywords:
Spaced seeds
Clustering
Metagenomics

ABSTRACT

The growing number of sequencing projects in medicine and environmental sciences is creating new computational demands in the analysis and processing of these very large datasets. Recently we have proposed an algorithm called MetaProb that can accurately cluster metagenomic reads with a precision that is currently unmatched. The competitive advantage of MetaProb depends on the use of sequence signatures based on contiguous k-mers. Instead of using contiguous k-mers, in this work we explore the use of spaced seeds where mismatches are allowed at carefully predetermined positions. The experimental results show that the use of mismatches can further improve the accuracy and decrease the memory requirements.

Availability: https://bitbucket.org/samu661/metaprob.

1. Introduction

Metagenomics is the study of complex microbial communities without cultivation steps [1]. The samples can be taken from a variety of environments, from soil and water to gut and skin. One of the primary goals of metagenomic studies is to determine the taxonomical identity of the micro-organisms that are present in a sample [2], which involves the analysis of short reads obtained from sequencing a metagenomic sample.

This analysis can reveal the presence of unexpected bacteria and viruses in a microbial sample, and it also allows the identification and characterization of new bacterial and viral genomes. For example, in the case of the human body, imbalances in the microbiome are known to be related with several diseases, e.g. inflammatory bowel disease (IBD) [3] and colorectal cancer [4].

The taxonomic analysis of microbial communities can be carried out by a process referred to as binning, in which reads from the same species are grouped together without the use of reference genomes. By binning reads, researchers can identify the number and the abundance of species in the environment.

Binning tools are based on the observation that the k-mer distributions of the DNA fragments from the same genome are more similar than those from different genomes. Thus, one can determine if two reads are from genomes of similar species based on their k-mer distributions. For this reason, several methods classify reads based on their k-mer distributions: BiMeta [5], MetaCluster [6], AbundanceBin [7]. However, one of the major problems when processing metagenomic reads is the fact that the proportion of species in a sample, a.k.a. abundance rate, can vary greatly. For example AbundanceBin [7] works well for very different abundance ratios, but problems arise when some species have similar abundance ratios.

* Corresponding authors.
E-mail addresses: comin@dei.unipd.it (M. Comin), cinzia.pizzi@dei.unipd.it (C. Pizzi).

http://dx.doi.org/10.1016/j.tcs.2017.05.023
0304-3975/

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023

In this context, we have recently proposed MetaProb in [8] and demonstrated that it is one of the best performing methods for metagenomic read binning. MetaProb is an assembly-assisted method that groups reads by considering overlapping information combined with probabilistic sequence signatures, based on k-mers counts.

The work presented in this manuscript exploits a new approach to improve MetaProb's clustering performances. Here we show that the accuracy can be increased by allowing mismatches, in a limited number of positions, while counting the number of shared k-mers. This concept is also known as spaced seeds [9], where mismatches are allowed at carefully predetermined positions. Recently, spaced seeds have been successfully applied to the field of metagenomic [10,11].

In the following we briefly summarize the idea of spaced seeds and the main developments in this field. Then, we describe how spaced seeds can be used instead of contiguous k-mers and implemented in MetaProbS (S for spaced). We report an extensive comparison of MetaProbS, against the original method MetaProb, evaluating 9 different spaced seeds. We show that for several metrics MetaProbS outperforms MetaProb.

1.1. Background on spaced seeds

Contiguous k-mers counts are at the bases of many sequence comparison methods, the most widely used and notable example is BLAST [12]. BLAST uses the so-called "hit and extend" method, where a hit consists of a match of a 11-mers between two sequences. Then these matches are potential candidates to be extended and to form a local alignment. It can be easily noticed that not all local alignments include an identical stretch of length 11. As observed in [13] requiring that the matches are not consecutive increases the chances of finding alignments. This idea of optimizing the position of the required matches has been investigated in many studies, and it was used in PatternHunter [9], another popular similarity search software.

A spaced seed of length k and weight $w < k$ is a string over the alphabet $\{1, *\}$ that contains w '1' and $(k - w)$ '*' symbols. A spaced seed is a mask where the symbols '1' and '*' denote respectively match and don't care positions. The advantage of using spaced seeds, instead of k-mers, is due to the fact that spaced seeds can model mutation events, often required by many applications in biology. From the statistical point of view, it is observed that spaced seeds occurrences at neighboring sequence positions are statistically less dependent than occurrences of contiguous k-mers [14]. Much work has been dedicated to spaced seeds over the years, we refer the reader to [15] for a survey on the earlier work.

Spaced seeds are now routinely used in many problems involving sequence comparison like: multiple sequence alignment [16], protein classification [17], read mapping [18] and for alignment-free phylogeny reconstruction [19]. More recently also metagenome reads clustering and classification can benefit from the use of spaced seeds [10,20].

Despite the fact that spaced seeds are widely used, to find the best spaced seed or set of spaced seeds remains a challenging computational problem. Ideally one would like to maximize the sensitivity of a spaced seeds, however it has been shown that calculating the sensitivity of a spaced seed is NP-hard [14]. The sensitivity can be approximated using dynamic programming, but the running time remains exponential in the length of the seed [9]. In 2011 Ilie et al. introduced SpEED [21] a tool for computing candidate spaced seeds. SpEED is based on the notion of overlap complexity that is correlated with sensitivity but it can be computed in polynomial-time. Recently Morgenstern et al. proposed a new method to compute sets of spaced seeds called rasbhari [11]. rasbhari uses a hill-climbing algorithm to optimize the overlap complexity of the produced pattern sets.

2. Methods

In metagenomic, one of the basic assumption is that the k-mer frequency distributions of long fragments (or whole genome sequences) are unique to each genome. However the current NGS technologies cannot produce reads long enough to directly apply k-mers/compositional distances. In order to solve this issue MetaProb addresses the problem of metagenomic read binning into two phases, by simulating the availability of long fragments. To keep the paper self-contained here we briefly describe MetaProb and the required modifications to incorporate spaced seeds.

In Fig. 1 is reported an outline of the two phases of MetaProb. In the first phase reads are grouped together based on the extent of their overlap. Ideally the overlap would be computed using a reads overlap graph [22], but this approach is highly demanding in terms of RAM. Instead we use an alignment-free technique frequently used in de-novo genome assembly.

The overlap between reads is estimated by considering the number of shared k-mers between reads. This technique relies on the assumption that, by choosing a sufficiently large value for k, the probability that two k-mers are shared by different genomes is low. For example, a study presented in [5] shows that the average ratio of common k-mers between pairs of bacterial genomes is less than 1.02% when $k = 30$. Therefore, the presence of a shared k-mer between two reads should indicate that the two reads belong to the same species. Moreover, if several such k-mers are shared, this strengthens the probability the two reads overlap. For these reasons, in the first phase, we construct groups where two reads overlap if they share at least m common k-mers. As a result the reads in a group, because of their overlap, are likely to belong to the same species. However, reads from a same species might be distributed in different groups. Thus, further processing is needed to cluster the groups obtained in Phase 1 based on their similarity.

The similarity between groups can be defined in terms of k-mers frequency distribution within each group. However, by construction, the reads in a group must have a significant overlap. Because such overlaps might artificially inflate the count of some k-mers, we developed a strategy, based on independent sets of a graph, to select a subset of reads from a group in

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023

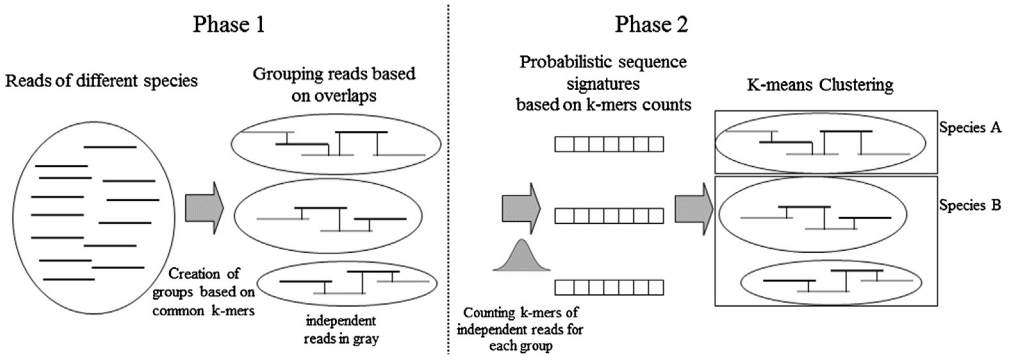


Fig. 1. An overview of the binning process of MetaProb. Phase 1 groups overlapping reads. Phase 2 merges the groups into clusters based on probabilistic sequence signatures of independent reads.

order to reduce the redundancy provided by large overlaps. In the second phase, each group is then represented by a set of independent reads on which the k-mers frequency distribution is computed. Because the Euclidean distance of k-mers distributions of different groups can be biased by the stochastic noise in each sequence [23,24], and by the possibly unbalanced size of the groups, we introduce a similarity measure based on self-standardized probabilistic sequence signatures. Thus in the second phase MetaProb clusters groups based on probabilistic sequence signatures (see for details [8]), so that reads from the same species are grouped together in one cluster.

In this work we explore a new idea of overlap between reads, not based on shared contiguous k-mers, but instead on shared spaced k-mers. A spaced seed is a mask over the alphabet $\{1, *\}$, where the two symbols '1' and '*' denote respectively match and don't care positions, and it is used to incorporate mismatches in several string comparison problems. We say that a spaced seed is of length k and it has weight $w < k$ if it contains w '1' and $(k - w)$ '*' symbols. If $k = w$, we have that the seed does not contain mismatches and it is equivalent to consider contiguous k-mers.

Let us consider for example the spaced seed $111 * 1$, with length 5 and weight 4. It has three consecutive matches followed by a don't care position (match or mismatch are allowed) and another match. By using this spaced seed, we can consider as equals the two sequences $TGAAG$ and $TGACG$, thus allowing for one mutation.

We need to choose carefully the parameters k and w of the spaced seed. For contiguous k-mers, precision increases as we increase k. However, the highest sensitivity occurs with somewhat shorter k-mers. For example Clark [25] is more precise for long contiguous k-mers (e.g., $k = 31$), but the highest sensitivity occurs for k-mers of length between 19 and 22. For these reasons a good compromise is to have spaced seeds where the number of required matches is $w = 22$, and the length is $k = 31$. These values have been reported also in [10,11] to improve the classification performance.

Also, the structure of spaced seeds is critical to achieve the highest possible precision and sensitivity. As discussed above, given k and w the problem of finding the optimal spaced seed, based on several evaluation criteria, can be computationally difficult [15]. However, the recent advancements in this field allow us to test spaced seeds optimized for different metrics. For example in [10] the authors report 3 spaced seeds, with length $k = 31$ and weight $w = 22$, that are designed to maximize the hit probability. Recently in [11] 6 new spaced seeds are proposed, 3 spaced seeds are computed minimizing the overlap complexity and another 3 spaced seeds to maximize the sensitivity. We selected these 9 spaced seeds for testing the impact on MetaProb:

1. from Clark-S [10], maximizing the hit probability:
   Q1 = 1111*111*111**1*111**1*11*11111
   Q2 = 11111*1*111**1*11*111**11*11111
   Q3 = 11111*1**111*1*11*11**111*11111
2. from rasbhari [11], minimizing overlap complexity:
   Q4 = 1111*1*111*1**11**111*11111*111
   Q5 = 111*111*111*1111*1**1*11**11111
   Q6 = 11111*1**1*111**11111*1*11*1111
3. from rasbhari [11], maximizing sensitivity:
   Q7 = 1111*1111**11*1*11111*1*1*11*11
   Q8 = 111*1*1*111*11**11*1**111111111
   Q9 = 111111*1*11*1*111**111*11**1111

In the context of MetaProb, spaced seeds allow to better detect the overlap between reads. In fact, in the original version of MetaProb, two reads are overlapping only if they share a contiguous stretch of length 30. With spaced seeds we require the two reads to share 22 matching positions that are not consecutive. For this reason spaced seeds can detect more overlaps between reads, by relaxing the requirements to have consecutive matches and by allowing for mutations.

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023

If more overlaps are discovered, then MetaProb can produce better estimate of the probabilistic sequence signature, and consequently improve the clustering performance.

2.1. Metrics

In order to estimate the performance of MetaProb using spaced seeds we used metrics to estimate both the quality of the binning, through precision, recall and F-measure, and the amount of computational resources needed, by measuring time and memory required for the analysis.

Furthermore, we evaluate the impact of spaced seeds in building the groups by measuring the average precision and group size.

We repeated these measures in two experimental frameworks: first we assume that the number of species is known and then we assume we do not have this information.

We now recall the definitions of precision, recall and F-measure as defined in [8]. Let m be the number of species in a metagenomic dataset, and k be the number of clusters returned by the binning algorithm. Let A_{ij} be the number of reads from species j assigned to cluster i. The ratio of reads assigned to a cluster that belong to the same species are called *precision* and is defined as follows:

$$precision = \frac{\sum_{i=1}^{k} \max_j A_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{m} A_{ij}} \tag{1}$$

If we consider groups as clusters, we can define the *groups precision* similarly to the just defined *precision*.

The ratio of reads from the same species that are assigned to the same cluster is called *recall*, and it is formally defined as:

$$recall = \frac{\sum_{j=1}^{m} \max_i A_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{m} A_{ij} + \#unassigned\_reads} \tag{2}$$

Finally, *F-measure* is a metric that emphasizes comprehensively both precision and recall:

$$F\text{-}measure = \frac{2 * precision * recall}{precision + recall} \tag{3}$$

For all metrics we have considered the average on all datasets, so that we can compare the overall behavior of different seeds.

3. Results and discussion

We compared the performances of MetaProb and MetaProbS on 10 different datasets that were also analyzed in previous works [5,8], using different spaced seeds and metrics, and assuming that the number of clusters is known. Furthermore, we also compared the two approaches on the subset of the datasets with a large number of species assuming no a-priori knowledge on the number of clusters, and thus also estimating its value. In both cases we compared the performances of MetaProbS with seeds Q1–Q9 against those of MetaProb with $k = 22$, which corresponds to the actual weight of our spaced seeds. Therefore, our comparison will show how, when considering the information coming from the same amount of symbols, using spaced seeds improves the overall performances of the binning process. Another interesting point of view is the one that considers the comparison between spaced seeds and k-mers of length 30. In such a context one can measure the impact of allowing some positions in the hashed segments not to be a match. As the performance of MetaProbS with respect to 30-mers are similar to those with respect to 22-mers we briefly discussed the differences in the text and reported the detailed figures in the appendix. Finally, we test MetaProbS on a real metagenomic dataset, a stool sample from the Human Microbiome Project.

All the experiments where run on a laptop equipped with an Intel i74510U CPU at 2 GHz, and 16 GB RAM. We start with a description of the datasets, and then discuss the results.

3.1. Datasets

For our experiments we considered 10 simulated bacterial genomes obtained with MetaSim [26]. These are the datasets $S_1 \ldots S_{10}$ and include paired-end short reads (length of approximately 80 bp) following an Illumina model with error rate of 1%. These datasets vary in terms of number of species (from 2 to 30), phylogenetic relationships among species, e.g. same family, and abundance ratio (balanced/unbalanced). We consider also a real stool metagenomic sample (SRR1804065) from the Human Microbiome Project. More detailed information on the composition of simulated and real datasets is shown in Table 5 in the Appendix.

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023

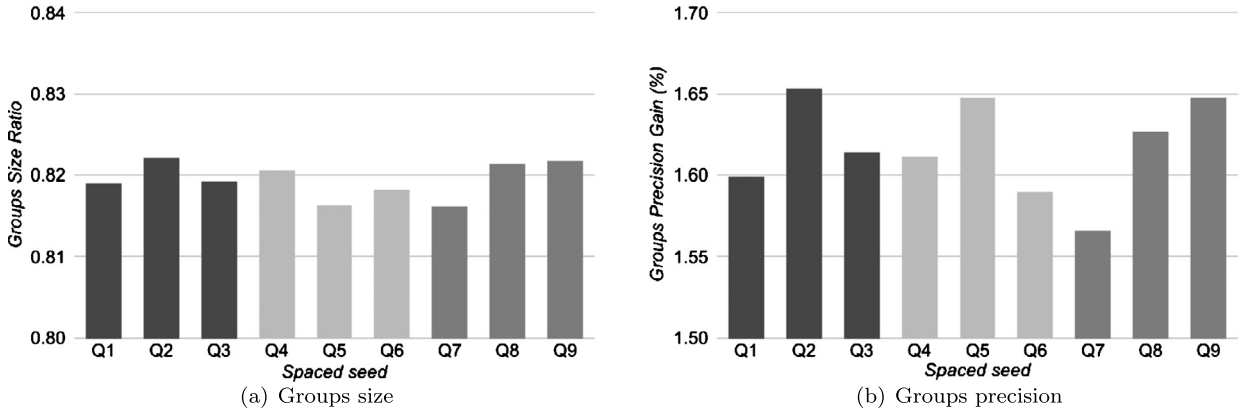


Fig. 2. Details of groups characteristics with spaced-seeds.

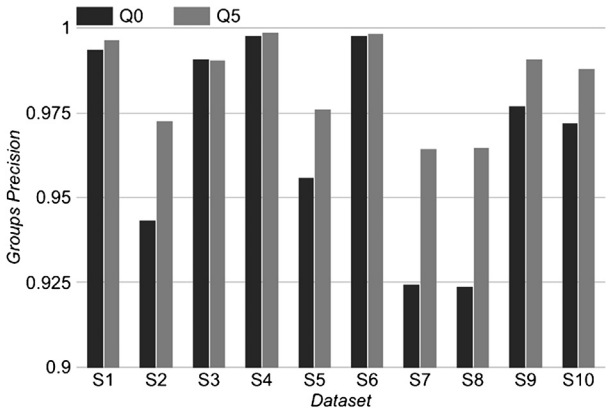


Fig. 3. Detailed groups precision between 22-mers Q0 and the spaced seed Q5.

3.2. Benchmark when the number of species is known

This set of experiments shows how the spaced seeds influence the construction of groups in the first phase of MetaProbS, and consequently how this in turn affect the binning algorithm. In the figures each bar corresponds to a spaced seed, and spaced seeds of the same type are filled with the same color: maximizing the hit probability (Q1, Q2, Q3, dark gray); minimizing overlap complexity (Q4, Q5, Q6, light gray); maximizing the sensitivity (Q7, Q8, Q9, medium gray). These experiments serve as a benchmark as we assume that the number of species in the sample is known a-priori.

3.2.1. Size and precision of groups

First of all we analyzed the features of the groups that are created with spaced seeds. Fig. 2 shows the average group size and group precision as defined in Section 2.1. We notice that, on average, and for any spaced seed, we obtain groups that are smaller (about 80%) but more precise (about 1.6%) than with MetaProb 22-mers. More in details, with respect to the group precision, we observed that in some cases the gain we obtain with seeds is small because for some datasets the baseline is actually already close to 100%.

As an example, we show in Fig. 3 the comparison of the group precision between the MetaProb baseline Q0, obtained with 22-mers, and what we obtain with the seed Q5 that, as we will see in the remaining of the discussion, is among those with the best performances overall. By using Q5 we have an improvement for all the datasets, with the only exception of S3 where the baseline is 99.09% and the loss is 0.03%, leading to a 99.06% precision, which is still very high. The most interesting aspects is that, although the precision is quite high also by using 22-mers, for the two datasets S7 and S8, where the baseline is around 92%, by using Q5 the gain is around 4%, bringing the overall precision above 96% for any dataset.

We recall that, by definition of groups precision, the gain refers to the whole dataset. So if we have 10 M reads, an increment of 1% in group precision implies that 100000 more reads are correctly grouped. We underline that this is not a peculiarity of seed Q5, but similar plots can be obtained also with the other spaced seeds, with some slight variations in terms of absolute values.

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023

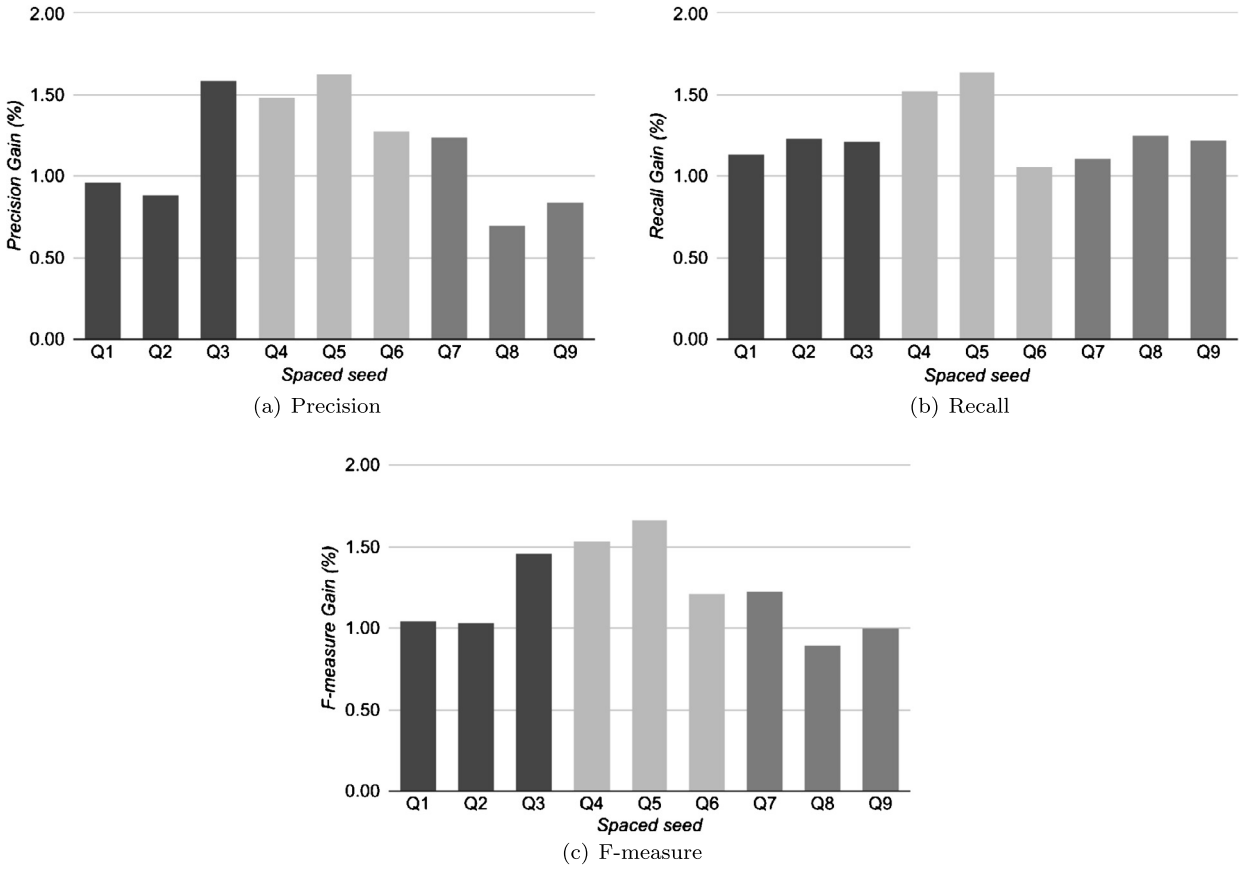


Fig. 4. Average difference of (a) precision, (b) recall, and (c) F-measure, when using Spaced-seeds (QX, $X = \{1, \ldots, 9\}$) with respect to the baseline Q0.

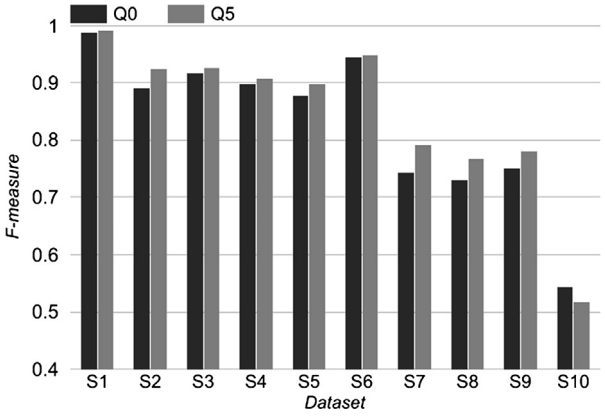


Fig. 5. Details of F-measure with seed Q5.

3.2.2. *Quality of the clustering*

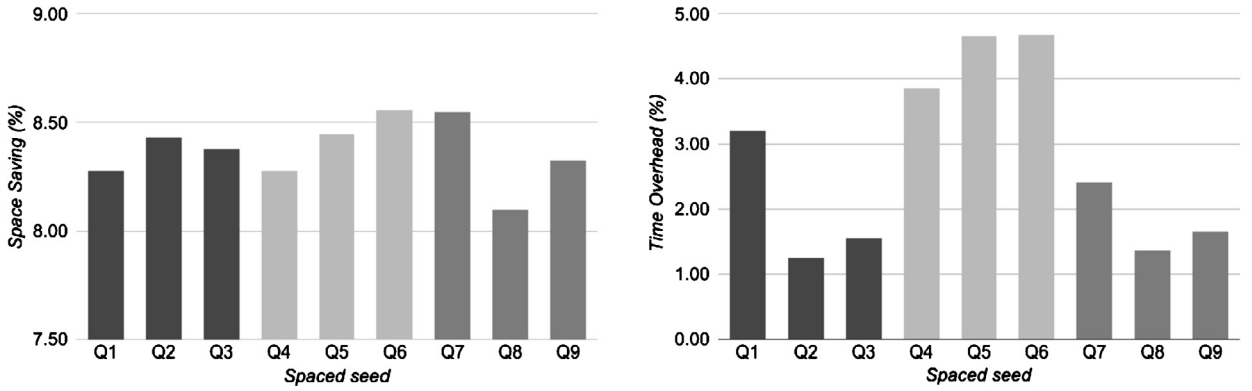
Having smaller and more precise groups has a direct impact on the probabilistic sequence signatures that are extracted from each group. Since these are the features according to which the groups are clustered together, using spaced seeds has an impact on the overall quality of the final clusters that we obtain.

Fig. 4 shows the gain in clustering quality obtained with space seeds with respect to the baseline MetaProb 22-mers.

First, we observe that precision, recall and F-measure all improve with respect to the baseline in a range that span from 1% to 1.6%. In general both the dataset composition and the spaced seed structure have an impact on the final quality of clustering. As before, we show in details what happens when we choose Q5 in Fig. 5.

The F-measure improves for all datasets, with the exception of S10. This dataset is among the most difficult to analyze as it has many species, a wide range of species abundance and phylogenetic relationships among them. Anyway, even for the

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023



(a) Space savings (b) Time overhead

Fig. 6. Impact on computational resources with spaced-seeds.

Table 1
Summary table of spaced seeds influence on MetaProbS with respect to MetaProb 22-mers. (Values in %.)

Seed	MinHitProb			MinOverlap			MaxSensitivity		
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Precision gain	0.96	0.88	1.59	1.48	1.63	1.28	1.24	0.70	0.84
Recall gain	1.14	1.23	1.21	1.52	1.64	1.06	1.11	1.25	1.22
F-measure gain	1.04	1.03	1.46	1.53	1.67	1.21	1.22	0.90	1.00
RAM saving	8.28	8.43	8.38	8.28	8.45	8.56	8.55	8.10	8.33
Time overhead	3.21	1.26	1.57	3.86	4.67	4.68	2.43	1.36	1.66

dataset S10 the F-measure obtained with Q5 is higher than the one obtained by other state of the art binning algorithms on the same dataset: Abundance Bin scores 0.14, MetaCluster 0.05, and BiMeta 0.43 [8]. As for the other datasets, the gain is relatively small for S1 to S6, where the baseline is close or above 90%. For S7, S8, and S9, where the baseline is around 75% the gain can reach up to 5%.

3.3. Computational resources

Fig. 6 shows the impact of using spaced seeds on the computational resources, namely the amount of memory and the time required for the computation.

We observe that for all spaced seeds we save between 8% and 8.5% of memory. This saving is possible because, although both with MetaProbS and MetaProb we index words of length 22, with spaced seeds we actually need to consider less positions, as the sliding window we use to hash the spaced seeds has length 31 rather than 22.

Using spaced seeds affect the time required for the computation at several steps. For example, the initial hashing of 22-mers with MetaProb is faster than hashing with spaced seeds because we can reuse the information about the previous k-mer to compute the next. The binning step is also affected by the k-mer distribution, which is necessarily different from the 22-mers as we still consider 22 positions but they are not necessarily contiguous. This in turn will affect the way the groups are built and the time needed for this step too. In general these differences do not necessarily imply a time overhead by using spaced seeds. In fact, for some datasets, MetaProbS is actually faster than MetaProb with any seed, for some other is slower with any seed and in some cases the behavior is very seed dependent. This said, on average, there is some time overhead with respect to MetaProb, in the range between 1.2% for Q2 and 4.68% for Q6. The MetaProb baseline with 22-mers span in a range from 14 seconds for processing S1 to 10 minutes for processing S10.

It is also worth noting as the spaced seeds for which the time overhead is higher is the class of seeds that minimize the overlap complexity, which are the ones with the best improvement in terms of quality of the clustering.

Table 1 summarizes the results we have shown so far given an overall view on the impact of seeds on the different aspects involved in the computation. Using spaced seeds leads to an improvement in terms of both overall quality of the final clustering and memory usage. Time overhead is expected when using spaced seeds rather than k-mers, as the model is intrinsically more complex. Nevertheless it is bounded within few units percentage.

In terms of class of spaced seeds we observe that those minimizing the overlap complexity are the ones with the best performances in terms of F-measure. However, they are also the ones with the highest time overhead. Among the other classes, we have that Q3, which belongs to the minimizing hit probability class, shows also good performance in terms of F-measure, slightly less than Q5 and Q4, but with a very small time overhead with respect to the baseline.

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023

Table 2
Summary table of spaced seed influence on MetaProbS with respect to MetaProb using 30-mers. (Values in %.)

Seed	MinHitProb			MinOverlap			MaxSensitivity		
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Precision gain	1.08	1.00	1.70	1.60	1.74	1.39	1.36	0.82	0.96
Recall gain	1.26	1.35	1.33	1.64	1.76	1.18	1.23	1.37	1.34
F-measure gain	1.16	1.15	1.58	1.65	1.79	1.33	1.34	1.02	1.12
RAM saving	2.91	3.06	3.01	2.90	3.09	3.20	3.19	2.72	2.96
Time overhead	4.61	2.48	2.87	5.30	6.13	6.07	3.74	2.62	2.92

Table 3
Summary table of spaced seed influence on MetaProbS. (Values in %.)

Seed	MinHitProb			MinOverlap			MaxSensitivity		
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
F-measure	2.38	3.02	1.93	2.58	2.27	2.33	0.82	3.64	2.07
RAM	12.08	12.07	12.13	11.86	12.06	12.10	12.28	11.82	12.06
Time	1.03	−2.29	−1.80	−0.25	−0.36	−2.24	5.15	−4.51	4.13
Grp_prec	2.68	2.76	2.67	2.75	2.77	2.63	2.61	2.68	2.70
Grp_size	0.82	0.82	0.82	0.82	0.82	0.82	0.81	0.82	0.82

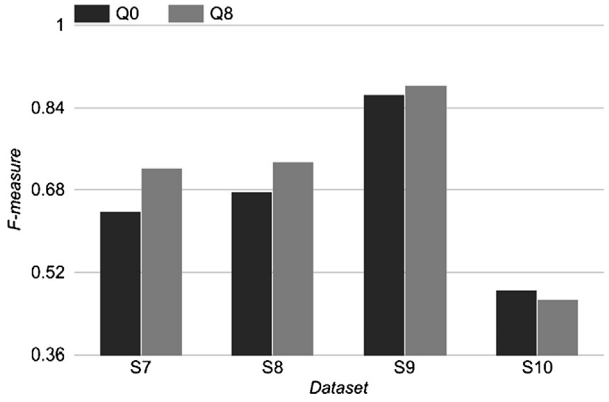


Fig. 7. Detailed F-measure comparison between 22-mers Q0 and the seed Q8.

The comparison with respect to MetaProb using 22-mers put in evidence how spaced seeds impact the performance of the analysis, by exploiting the same amount of information in terms of symbols that must match. Similarly, we performed the same set of experiments with respect to MetaProb using 30-mers. In this case the results put in evidence the difference when hashing segments of similar length but allowing, with the spaced seeds, that some positions do not necessarily match. The results are reported in Table 2.

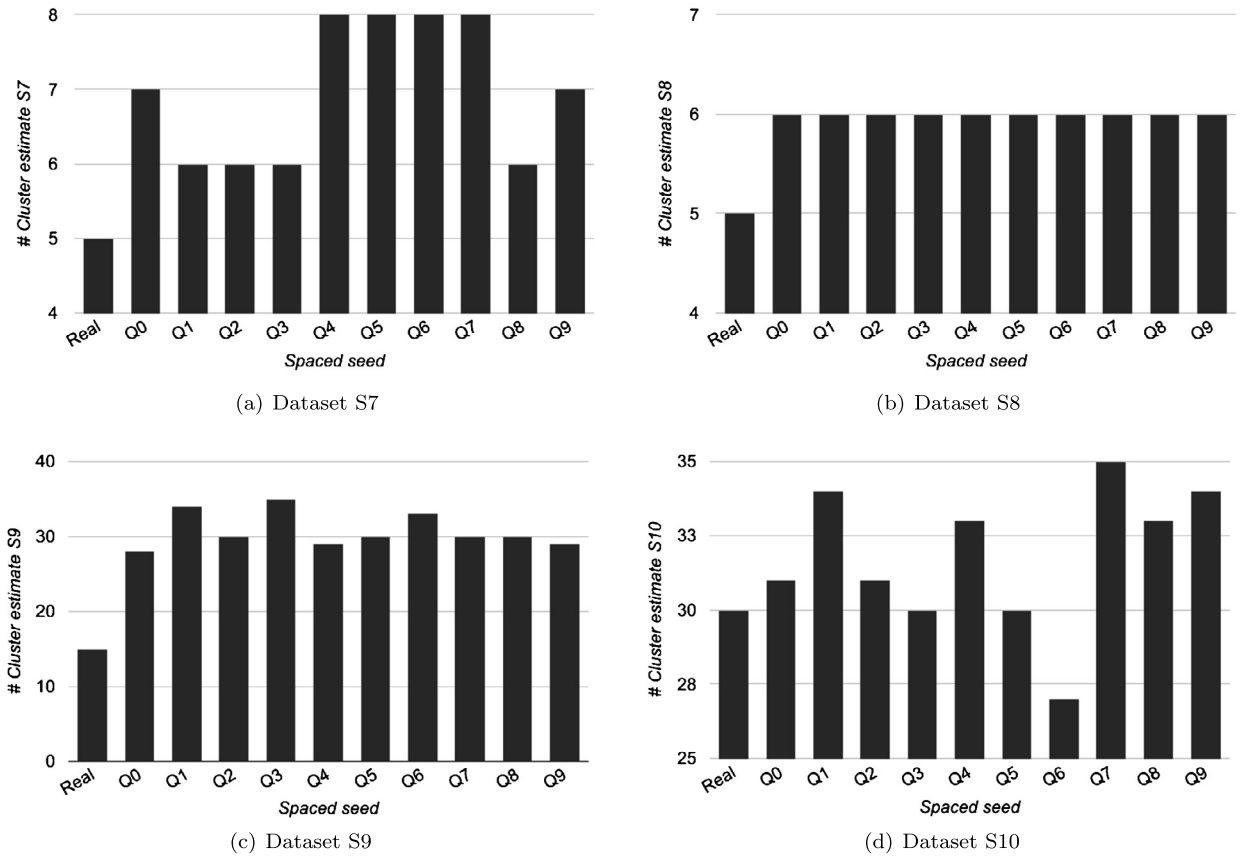
From comparison with Table 1 we can see that the improvements with spaced-seeds with respect to MetaProb using 30-mers, are comparable to those obtained with spaced-seeds with respect to MetaProb using 22-mers. More in details, the quality of the clustering slightly improves, the memory usage is still better than using MetaProb, although the gain is reduced, while the time overhead slightly increases. Figures with the trends for each measured parameter are reported in the Appendix.

3.4. Performance analysis when estimating the number of clusters

This experimental setting is more realistic, as we assumed no prior knowledge on the composition of the sample and limited the analysis to the datasets with more species: S7, S8, S9 and S10 with, respectively 5, 5, 15, and 30 species.

Table 3 shows the results of the impact of seeds in this framework. We first observe that the improvements in terms of F-measure, RAM savings, and group precision are higher than in the framework where the number of species is known. As for the F-measure, its value can vary substantially depending on the combinations dataset/seed that are considered. For example, the seed Q8 has both the maximum gain (more than 8% on dataset S7), and the maximum loss (about 1.6% on dataset S10). We report in Fig. 7 the detailed behavior of Q8 with respect to the analysis in the same framework with MetaProb 22-mers.

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023



(a) Dataset S7 (b) Dataset S8

(c) Dataset S9 (d) Dataset S10

Fig. 8. Estimation of the number of clusters with 22-mers Q0 and the seeds.

As for the usage of computational resources, there is an homogeneous trend of space saving, even more accentuated than in the previous framework. Moreover, there is a general speed up with respect to the baseline. The only relevant average slow downs are given by Q7 and Q9 that, although faster than the baseline in the analysis of S7–S9, are substantially slower in the processing of S10. This behavior is indeed shared also by the other seeds, but for the others the overhead for S10 is less accentuated.

For the groups composition, the observations are similar to the case when the number of species was known. Using spaced seeds we obtain smaller but more precise clusters.

Finally, we verified the performances in the estimation of the number of clusters with our approach using spaced seeds and using 22-mers. The results are shown in Fig. 8 for each of the datasets.

With respect to the actual number of clusters (first column in the plots), there is a generalized overestimation of the number of clusters. Some seeds perform better than the Q0 baseline, but some other do not, or it depends on the dataset. In any case the difference in using MetaProb with 22-mers or spaced seeds is in general quite limited.

3.5. Experiments on real data

We also evaluated the performance of MetaProbS on a real stool metagenomic sample (SRR1804065) from the Human Microbiome Project. Because there is no ground truth for this dataset, we use BLAST to find the reads that uniquely map to a genome and filter out all other reads. As a result the real metagenomic sample contains 775 distinct species and 1053741 reads.

We run MetaProbS on this dataset using the spaced seed Q3, one of the best seeds according to previous tests, and evaluate the quality of the clusters produced. We report the results only for this spaced seed because similar results can be obtained with the other seeds. On this dataset MetaProbS reports 9 clusters of various sizes. In Table 4 we show these clusters sorted by size.

For each cluster we report the majority species, the precision of the cluster, the abundance rate of the cluster, and the real abundance rate of the majority species in the input metagenome. The cluster abundance rate is computed as the size of clusters divided by the total number of reads. The abundance rate of the majority species is the number of reads assigned to the majority species divided by the total number of reads.

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023

Table 4
Results on a real stool metagenomic sample (SRR1804065). Clusters reported by MetaProbS in decreasing order of size.

Cluster	Majority species	Precision	Cluster Abund.	Species Abund.
1	Bacteroides Vulgatus	85.9%	46.7%	60.1%
2	Bacteroides Vulgatus	64.0%	18.4%	60.1%
3	Bacteroides Vulgatus	41.1%	7.5%	60.1%
4	Bacteroides Salanitronis	70.5%	7.4%	9.0%
5	Bacteroides Vulgatus	49.3%	5.3%	60.1%
6	Faecalibacterium Prausnitzii	71.5%	4.4%	4.3%
7	Parabacteroides Distasonis	72.4%	4.1%	7.0%
8	Parabacteroides Distasonis	41.5%	4.0%	7.0%
9	Odoribacter Splanchnicus	49.3%	2.3%	4.3%

First of all, we can see that 4 clusters have an high precision above 70%, and thus they can group together reads from the same species. In this real dataset the species abundance ratios are particular unbalanced, in fact 60% of reads belong to the species of Bacteroides Vulgatus. For this reason Bacteroides Vulgatus spans across 4 different clusters. The second most abundant species in the real dataset is Bacteroides Salanitronis (9.0%), that appears also to be the second most abundant species reported by MetaProbS, and it is associated to cluster number 4 with high precision. Overall MetaProbS is able to detect 5 species that are also the most abundant in the real dataset, and that account for 85% of the reads. These bacteria were also reported as the most abundant species in human feces [3].

4. Conclusions

In this paper we exploit the spaced seed model to increase the quality of metagenomic read binning. Several different type of spaced seeds are considered and used within the MetaProb approach instead of contiguous k-mers. Experiments on several datasets showed that using spaced seeds improves the quality of the binning, allowing also for a better use of computational resources. On a real fecal metagenomic data MetaProbS was able to detect the most abundant species with high precision.

Funding

This work has been supported by the PRIN project n. 20122F87B2.

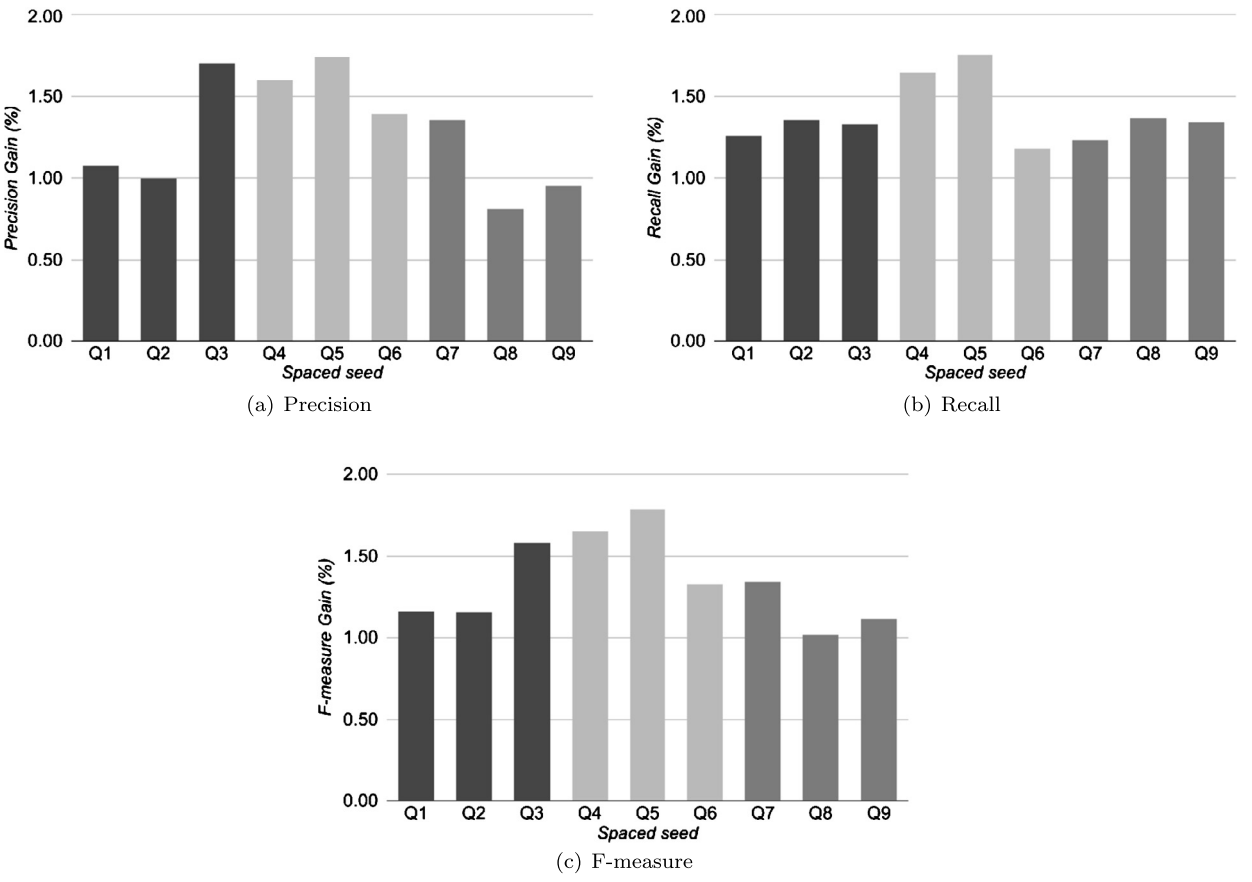
5. Appendices

5.1. Dataset description

Table 5 shows the details of the datasets used in the experiments.

Table 5
Description table of the datasets used in the experiments.

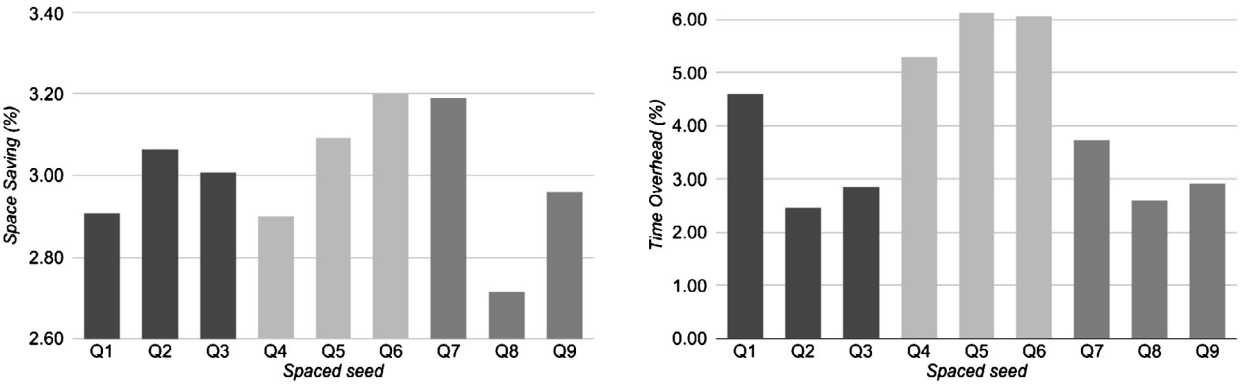
Dataset	N. species	Phylogenetic relationship	Abundance ratio	Tot. read
S1	2	Species	1:1	192734
S2	2	Species	1:1	390678
S3	2	Order	1:1	677450
S4	2	Phylum	1:1	750578
S5	3	Species and Family	1:1:1	650800
S6	3	Phylum and Kingdom	3:2:1	1426764
S7	5	Order, Order, Genus, Order	1:1:1:4:4	3307100
S8	5	Genus, Order, Order, Order	3:5:7:9:11	912448
S9	15	various relationships	1:1:1:1:1: 2:2:2:2:2: 3:3:3:3:3 4:4:4:4:4: 6:6:6:6:6:	4468336
S10_S	30	various relationships	7:7:7:7:7: 8:8:8:8:8: 9:9:9:9:9: 10:10:10:10:10	3000000
SRR1804065	775	various relationships	various abundances	1053741



(a) Precision (b) Recall

(c) F-measure

Fig. 9. Average difference of (a) precision, (b) recall, and (c) F-measure, when using MetaProbS with spaced seeds (QX, $X = \{1, \ldots, 9\}$) with respect to the baseline MetaProb using 30-mers.



(a) Space savings (b) Time overhead

Fig. 10. Impact on computational resources with MetaProbS with respect to MetaProb using 30-mers.

5.2. MetaProbS performances with respect to MetaProb using 30-mers

Figures 9 and 10 show the detailed performance analysis of MetaProbS with respect to MetaProb using 30-mers in terms of quality of the clustering in Fig. 9, and computational resources 10.

References

[1] P. Hugenholtz, G. Tyson, Microbiology: metagenomics, Nature 455 (2008) 481–483.

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023

[2] S.S. Mande, M.H. Mohammed, T.S. Ghosh, Classification of metagenomic sequences: methods and challenges, Brief. Bioinform. 13 (6) (2012) 669–681, http://dx.doi.org/10.1093/bib/bbs054, URL http://bib.oxfordjournals.org/content/13/6/669.
[3] J. Qin, R. Li, J. Raes, et al., A human gut microbial gene catalogue established by metagenomic sequencing, Nature 464 (2010) 59–65.
[4] G. Zeller, J. Tap, A.Y. Voigt, S. Sunagawa, J.R. Kultima, P.I. Costea, A. Amiot, J. Böhm, F. Brunetti, N. Habermann, R. Hercog, M. Koch, A. Luciani, D.R. Mende, M.A. Schneider, P. Schrotz-King, C. Tournigand, J. Tran Van Nhieu, T. Yamada, J. Zimmermann, V. Benes, M. Kloor, C.M. Ulrich, M. von Knebel Doeberitz, I. Sobhani, P. Bork, Potential of fecal microbiota for early-stage detection of colorectal cancer, Mol. Syst. Biol. 10 (11) (2014), http://msb.embopress.org/content/10/11/766.full.pdf, http://dx.doi.org/10.15252/msb.20145645, URL http://msb.embopress.org/content/10/11/766.
[5] L.V. Vinh, T.V. Lang, L.T. Binh, T.V. Hoai, A two-phase binning algorithm using l-mer frequency on groups of non-overlapping reads, Algorithms Mol. Biol. 10 (1) (2015) 1–12, http://dx.doi.org/10.1186/s13015-014-0030-4.
[6] Y. Wang, H.C. Leung, S.M. Yiu, F.Y. Chin, MetaCluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample, Bioinformatics 28 (2012), http://dx.doi.org/10.1093/bioinformatics/bts397.
[7] Y.-W. Wu, Y. Ye, A novel abundance-based algorithm for binning metagenomic sequences using l-tuples, J. Comput. Biol. 18 (3) (2011) 523–534, http://dx.doi.org/10.1089/cmb.2010.0245, URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3123841/.
[8] S. Girotto, C. Pizzi, M. Comin, MetaProb: accurate metagenomic reads binning based on probabilistic sequence signatures, Bioinformatics 32 (17) (2016) i567–i575, http://dx.doi.org/10.1093/bioinformatics/btw466, URL http://bioinformatics.oxfordjournals.org/content/32/17/i567.
[9] B. Ma, J. Tromp, M. Li, PatternHunter: faster and more sensitive homology search, Bioinformatics 18 (3) (2002) 440, http://dx.doi.org/10.1093/bioinformatics/18.3.440.
[10] R. Ounit, S. Lonardi, Higher classification sensitivity of short metagenomic reads with CLARK-S-s, Bioinformatics 32 (24) (2016) 3823, http://dx.doi.org/10.1093/bioinformatics/btw542.
[11] L. Hahn, C.-A. Leimeister, R. Ounit, S. Lonardi, B. Morgenstern, rasbhari: optimizing spaced seeds for database searching, read mapping and alignment-free sequence comparison, PLoS Comput. Biol. 12 (10) (2016) 1–18, http://dx.doi.org/10.1371/journal.pcbi.1005107.
[12] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, Basic local alignment search tool, J. Mol. Biol. 215 (3) (1990) 403–410, http://dx.doi.org/10.1016/S0022-2836(05)80360-2, www.sciencedirect.com/science/article/pii/S0022283605803602.
[13] J. Buhler, Efficient large-scale sequence comparison by locality-sensitive hashing, Bioinformatics 17 (5) (2001) 419, http://dx.doi.org/10.1093/bioinformatics/17.5.419.
[14] B. Ma, M. Li, On the complexity of the spaced seeds, J. Comput. System Sci. 73 (7) (2007) 1024–1034, http://dx.doi.org/10.1016/j.jcss.2007.03.008, URL www.sciencedirect.com/science/article/pii/S0022000007000268.
[15] D.G. Brown, M. Li, B. Ma, A tutorial of recent developments in the seeding of local alignment, J. Bioinform. Comput. Biol. 02 (04) (2004) 819–842, http://dx.doi.org/10.1142/S0219720004000983, http://www.worldscientific.com/doi/pdf/10.1142/S0219720004000983, URL http://www.worldscientific.com/doi/abs/10.1142/S0219720004000983.
[16] A.E. Darling, T.J. Treangen, L. Zhang, C. Kuiken, X. Messeguer, N.T. Perna, Procrastination Leads to Efficient Filtration for Local Multiple Alignment, Springer, Berlin, Heidelberg, Berlin, Heidelberg, 2006, pp. 126–137.
[17] T. Onodera, T. Shibuya, The gapped spectrum kernel for support vector machines, in: Proceedings of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM'13, Springer-Verlag, Berlin, Heidelberg, 2013, pp. 1–15.
[18] S.M. Rumble, P. Lacroute, A.V. Dalca, M. Fiume, A. Sidow, M. Brudno, Shrimp: accurate mapping of short color-space reads, PLoS Comput. Biol. 5 (5) (2009) 1–11, http://dx.doi.org/10.1371/journal.pcbi.1000386.
[19] C.-A. Leimeister, M. Boden, S. Horwege, S. Lindner, B. Morgenstern, Fast alignment-free sequence comparison using spaced-word frequencies, Bioinformatics 30 (14) (2014) 1991, http://dx.doi.org/10.1093/bioinformatics/btu177.
[20] K. Břinda, M. Sykulski, G. Kucherov, Spaced seeds improve k-mer-based metagenomic classification, Bioinformatics 31 (22) (2015) 3584, http://dx.doi.org/10.1093/bioinformatics/btv419.
[21] L. Ilie, S. Ilie, A. Mansouri Bigvand, Speed: fast computation of sensitive spaced seeds, Bioinformatics 27 (17) (2011) 2433, http://dx.doi.org/10.1093/bioinformatics/btr368.
[22] E. Myers, The fragment assembly string graph, Bioinformatics 21 (suppl. 2) (2005) ii79–ii85, http://dx.doi.org/10.1093/bioinformatics/bti1114.
[23] R.A. Lippert, H. Huang, M.S. Waterman, Distributional regimes for the number of k-word matches between two random sequences, Proc. Natl. Acad. Sci. USA 99 (22) (2002) 13980–13989, http://dx.doi.org/10.1073/pnas.202468099, http://www.pnas.org/content/99/22/13980.full.pdf, URL http://www.pnas.org/content/99/22/13980.abstract.
[24] K. Song, J. Ren, G. Reinert, M. Deng, M.S. Waterman, F. Sun, New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing, Brief. Bioinform. 15 (3) (2014) 343–353, http://dx.doi.org/10.1093/bib/bbt067, http://bib.oxfordjournals.org/content/15/3/343.full.pdf+html.
[25] R. Ounit, S. Wanamaker, T.J. Close, S. Lonardi, CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers, BMC Genomics 16 (2015) 236, http://dx.doi.org/10.1186/s12864-015-1419-2.
[26] D.C. Richter, F. Ott, A.F. Auch, R. Schmid, D.H. Huson, MetaSim: a sequencing simulator for genomics and metagenomics, PLoS ONE 3 (10) (2008) e3373, http://dx.doi.org/10.1371/journal.pone.0003373.

Please cite this article in press as: S. Girotto et al., Metagenomic reads binning with spaced seeds, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.05.023