QCluster: Extending Alignment-Free Measures with Quality Values for Reads Clustering

Matteo Comin, Andrea Leoni, and Michele Schimd

Department of Information Engineering, University of Padova, Padova, Italy
comin@dei.unipd.it

Abstract. The data volume generated by Next-Generation Sequencing (NGS) technologies is growing at a pace that is now challenging the storage and data processing capacities of modern computer systems. In this context an important aspect is the reduction of data complexity by collapsing redundant reads in a single cluster to improve the run time, memory requirements, and quality of post-processing steps like assembly and error correction. Several alignment-free measures, based on k-mers counts, have been used to cluster reads.

Quality scores produced by NGS platforms are fundamental for various analysis of NGS data like reads mapping and error detection. Moreover future-generation sequencing platforms will produce long reads but with a large number of erroneous bases (up to 15%). Thus it will be fundamental to exploit quality value information within the alignment-free framework.

In this paper we present a family of alignment-free measures, called D^q-type, that incorporate quality value information and k-mers counts for the comparison of reads data. A set of experiments on simulated and real reads data confirms that the new measures are superior to other classical alignment-free statistics, especially when erroneous reads are considered. These measures are implemented in a software called QCluster (`http://www.dei.unipd.it/~ciompin/main/qcluster.html`).

Keywords: alignment-free measures, reads quality values, clustering reads.

1 Introduction

The data volume generated by Next-Generation Sequencing (NGS) technologies is growing at a pace that is now challenging the storage and data processing capacities of modern computer systems [1]. Current technologies produce over 500 billion bases of DNA per run, and the forthcoming sequencers promise to increase this throughput. The rapid improvement of sequencing technologies has enabled a number of different sequencing-based applications like genome resequencing, RNA-Seq, ChIP-Seq and many others [2]. Handling and processing such large files is becoming one of the major challenges in most genome research projects.

D. Brown and B. Morgenstern (Eds.): WABI 2014, LNBI 8701, pp. 1–13, 2014.

Alignment-based methods have been used for quite some time to establish similarity between sequences [3]. However there are cases where alignment methods can not be applied or they are not suited. For example the comparison of whole genomes is impossible to conduct with traditional alignment techniques, because of events like rearrangements that can not be captured with an alignment [4–6]. Although fast alignment heuristics exist, another drawback is that alignment methods are usually time consuming, thus they are not suited for large-scale sequence data produced by Next-Generation Sequencing technologies (NGS)[7, 8]. For these reasons a number of alignment-free techniques have been proposed over the years [9].

The use of alignment-free methods for comparing sequences has proved useful in different applications. Some alignment-free measures use the patterns distribution to study evolutionary relationships among different organisms [4, 10, 11]. Several alignment-free methods have been devised for the detection of enhancers in ChIP-Seq data [12–14] and also of entropic profiles [15, 16]. Another application is the classification of protein remotely related, which can be addressed with sophisticate word counting procedures [17, 18]. The assembly-free comparison of genomes based on NGS reads has been investigated only recently [7, 8]. For a comprehensive review of alignment-free measures and applications we refer the reader to [9].

In this study we want to explore the ability of alignment-free measures to cluster reads data. Clustering techniques are widely used in many different applications based on NGS data, from error correction [19] to the discovery of groups of microRNAs [20]. With the increasing throughput of NGS technologies another important aspect is the reduction of data complexity by collapsing redundant reads in a single cluster to improve the run time, memory requirements, and quality of subsequent steps like assembly.

In [21] Solovyov *et. al.* presented one of the first comparison of alignment-free measures when applied to NGS reads clustering. They focused on clustering reads coming from different genes and different species based on k-mer counts. They showed that D-type measures (see section 2), in particular D_2^*, can efficiently detect and cluster reads from the same gene or species (as opposed to [20] where the clustering is focused on errors). In this paper we extend this study by incorporating quality value information into these measures.

Quality scores produced by NGS platforms are fundamental for various analysis of NGS data: mapping reads to a reference genome [22]; error correction [19]; detection of insertion and deletion [23] and many others. Moreover future-generation sequencing technologies will produce long and less biased reads with a large number of erroneous bases [24]. The average number of errors per read will grow up to 15%, thus it will be fundamental to exploit quality value information within the alignment-free framework and the *de novo* assembly where longer and less biased reads could have dramatic impact.

In the following section we briefly review some alignment-free measures. In section 3 we present a new family of statistics, called D^q-type, that take advantage of quality values. The software QCluster is discussed in section 4 and

relevant results on simulated and real data are presented in section 5. In section 6 we summarize the findings and we discuss future directions of investigation.

2 Previous Work on Alignment-Free Measures

One of the first papers that introduced an alignment-free method is due to Blaisdell in 1986 [25]. He proposed a statistic called D_2, to study the correlation between two sequences. The initial purpose was to speed up database searches, where alignment-based methods were too slow. The D_2 similarity is the correlation between the number of occurrences of all k-mers appearing in two sequences. Let X and Y be two sequences from an alphabet Σ. The value X_w is the number of times w appears in X, with possible overlaps. Then the D_2 statistic is:

$$D_2 = \sum_{w \in \Sigma^k} X_w Y_w.$$

This is the inner product of the word vectors X_w and Y_w, each one representing the number of occurrences of words of length k, *i.e.* k-mers, in the two sequences. However, it was shown by Lippert *et al.* [26] that the D_2 statistic can be biased by the stochastic noise in each sequence. To address this issue another popular statistic, called D_2^z, was introduced in [13]. This measure was proposed to standardize the D_2 in the following manner:

$$D_2^z = \frac{D_2 - \mathbb{E}(D_2)}{\mathbb{V}(D_2)},$$

where $\mathbb{E}(D_2)$ and $\mathbb{V}(D_2)$ are the expectation and the standard deviation of D_2, respectively. Although the D_2^z similarity improves D_2, it is still dominated by the specific variation of each pattern from the background [27, 28]. To account for different distributions of the k-mers, in [27] and [28] two other new statistics are defined and named D_2^* and D_2^s. Let $\tilde{X}_w = X_w - (n - k + 1) * p_w$ and $\tilde{Y}_w = Y_w - (n-k+1) * p_w$ where p_w is the probability of w under the null model. Then D_2^* and D_2^s can be defined as follows:

$$D_2^* = \sum_{w \in \Sigma^k} \frac{\tilde{X}_w \tilde{Y}_w}{(n - k + 1)p_w}.$$

and,

$$D_2^s = \sum_{w \in \Sigma^k} \frac{\tilde{X}_w \tilde{Y}_w}{\sqrt{\tilde{X}_w^2 + \tilde{Y}_w^2}}$$

This latter similarity measure responds to the need of normalization of D_2. These set of alignment-free measures are usually called D-type statistics. All these statistics have been studied by Reinert *et al.* [27] and Wan *et al.* [28] for the detection of regulatory sequences. From the word vectors X_w and Y_w several other measures can be computed like L_2, Kullback-Leibler divergence (KL), symmetrized KL [21] etc.

3 Comparison of Reads with Quality Values

3.1 Background on Quality Values

Upon producing base calls for a read x, sequencing machines also assign a *quality score* $Q_x(i)$ to each base in the read. These scores are usually given as *phred*-scaled probability [29] of the i-th base being wrong

$$Q_x(i) = -10 \log_{10} Prob\{\text{the base } i \text{ of read } x \text{ is wrong }\}.$$

For example, if $Q_x(i) = 30$ then there is 1 in 1000 chance that base i of read x is incorrect. If we assume that quality values are produced independently to each other (similarly to [22]), we can calculate the probability of an entire read x being correct as:

$$P_x\{\text{the read } x \text{ is correct}\} = \prod_{j=0}^{n-1} (1 - 10^{-Q_x(j)/10})$$

where n is the length of the read x. In the same way we define the probability of a word w of length k, occuring at position i of read x being correct as:

$$P_{w,i}\{\text{the word } w \text{ at position } i \text{ of read } x \text{ is correct}\} = \prod_{j=0}^{k-1} (1 - 10^{-Q_x(i+j)/10}).$$

In all previous alignment-free statistics the k-mers are counted such that each occurrence contributed as 1 irrespective of its quality. Here we can use the quality of that occurrence instead to account also for erroneous k-mers. The idea is to model sequencing as the process of reading k-mers from the reference and assigning a probability to them. Thus this formula can be used to weight the occurrences of all k-mers used in the previous statistics.

3.2 New D^q-Type Statistics

We extend here D-type statistics [27, 28] to account for quality values. By defining X_w^q as the sum of probabilities of all the occurrences of w in x:

$$X_w^q = \sum_{i \in \{i|\ w \text{ occurs in } x \text{ at position } i\}} P_{w,i}$$

we assign a weight (*i.e.* a probability) to each occurrence of w. Now X_w^q can be used instead of X_w to compute the alignment-free statistics. Note that, by using X_w^q, every occurrence is not counted as 1, but with a value in $[0, 1]$ depending of the reliability of the read. We can now define a new alignment-free statistic as :

$$D_2^q = \sum_{w \in \Sigma^k} X_w^q Y_w^q.$$

This is the extension of the D_2 measure, in which occurrences are weighted based on quality scores. Following section 2 we can also define the centralized k-mers counts as follows:

$$\widetilde{X_w^q} = X_w^q - (n-k+1)p_w E(P_w)$$

where $n = |x|$ is the length of x, p_w is the probability of the word w in the i.i.d. model and the expected number of occurrences $(n-k+1)p_w$ is multiplied by $E(P_w)$ which represents the expected probability of k-mer w based on the quality scores.

We can now extend two other popular alignment-free statistics:

$$D_2^{*q} = \sum_{w \in \Sigma^k} \frac{\tilde{X_w^q} \tilde{Y_w^q}}{(n-k+1)p_w E(P_w)}$$

and,

$$D_2^{sq} = \sum_{w \in \Sigma^k} \frac{\tilde{X_w^q} \tilde{Y_w^q}}{\sqrt{\tilde{X_w^q}^2 + \tilde{Y_w^q}^2}}$$

We call these three alignment-free measures D^q-type. Now, $E(P_w)$ depends on w and on the actual sequencing machine, therefore it can be very hard, if not impossible, to calculate precisely. However, if the set $\mathbb{D}$ of all the reads is large enough we can estimate the prior probability using the posterior relative frequency, *i.e.* the frequency observed on the actual set $\mathbb{D}$, similarly to [22]. We assume that, given the quality values, the error probability on a base is independent from its position within the read and from all other quality values (see [22]). We defined two different approximations, the first one estimates $E(P_w)$ as the average error probability of the k-mer w among all reads $x \in \mathbb{D}$:

$$E(P_w) \approx \frac{\sum_{x \in \mathbb{D}} X_w^q}{\sum_{x \in \mathbb{D}} X_w} \tag{1}$$

while the second defines, for each base j of w, the average quality observed over all occurrences of w in $\mathbb{D}$:

$$\overline{Q_w}[j] = \frac{\sum_{x \in \mathbb{D}} \sum_{i \in \{i|\ w \text{ occurs in } x \text{ at position } i\}} Q_x(i+j)}{\sum_{x \in \mathbb{D}} X_w}$$

and it uses the average quality values to compute the expected word probability.

$$E(P_w) \approx \prod_{j=0}^{k-1} (1 - 10^{-\overline{Q_w}(j)/10}) \tag{2}$$

We called the first approximation *Average Word Probability (AWP)* and the second one *Average Quality Probability (AQP)*. Both these approximations are implemented within the software QCluster and they will tested in section 5.

3.3 Quality Value Redistribution

If we consider the meaning of quality values it is possible to further exploit it to extend and improve the above statistics. Let's say that the base A has quality 70%, it means that there is a 70% probability that the base is correct. However there is also another 30% probability that the base is incorrect. Let's ignore for the moment insertion and deletion errors, if the four bases are equiprobable, this means that with uniform probability 10% the wrong base is a C, or a G or a T. It's therefore possible to redistribute the "missing quality" among other bases. We can perform a more precise operation by redistributing the missing quality among other bases in proportion to their frequency in the read. For example, if the frequencies of the bases in the read are A=20%, C=30%, G=30%, T=20%, the resulting qualities, after the redistribution, will be: A=70%, $C = 30\%*30\%/(30\%+30\%+20\%) = 11\%$, $G = 30\%*30\%/(30\%+30\%+20\%) = 11\%$, $T = 30\% * 20\%/(30\% + 30\% + 20\%) = 7,5\%$.

The same redistribution, with a slight approximation, can be extended to k-mers quality. More in detail, we consider only the case in which only one base is wrong, thus we redistribute the quality of only one base at a time. Given a k-mer, we generate all neighboring words that can be obtained by substitution of the wrong base. The quality of the replaced letter is calculated as in the previous example and the quality of the entire word is again given by the product of the qualities of all the bases in the new k-mers. We increment the corresponding entry of the vector X^q_w with the score obtained for the new k-mer. This process is repeated for all bases of the original k-mer. Thus every time we are evaluating the quality of a word, we are also scoring neighboring k-mers by redistributing the qualities. We didn't consider the case where two or more bases are wrong simultaneously, because the computational cost would be too high and the quality of the resulting word would not appreciably affect the measures.

4 QCluster: Clustering of Reads with D^q-Type Measures

All the described algorithms were implemented in the software QCluster. The program takes in input a fastq format file and performs centroid-based clustering (k-means) of the reads based on the counts and the quality of k-mers. The software performs centroid-based clustering with KL divergence and other distances like L_2 (Euclidean), D_2, D_2^*, symmetrized KL divergence etc. When using the D^q-type measures, one needs to choose the method for the computation of the expected word probability, AWP or AQP, and the quality redistribution.

Since some of the implemented distances (symmetrized KL, D_2^*) do not guarantee to converge, we implemented a stopping criteria. The execution of the algorithm interrupts if the number of iterations without improvements exceeds a certain threshold. In this case, the best solution found is returned. The maximum number of iterations may be set by the user and for our experiments we use the value 5. Several other options like reverse complement and different normalization are available. All implemented measures can be computed in linear time

and space, which is desirable for large NGS datasets. The QCluster[1] software has been implemented in C++ and compiled and tested using GNU GCC.

5 Experimental Results

Several tests have been performed in order to estimate the effectiveness of the different distances, on both simulated and real datasets. In particular, we had to ensure that, with the use of the additional information of quality values, the clustering improved compared to that produced by the original algorithms.

For simulations we use the dataset of human mRNA genes downloaded from NCBI[2], also used in [21]. We randomly select 50 sets of 100 sequences each of human mRNA, with the length of each sequence ranged between 500 and 10000 bases. From each sequence, 10000 reads of length 200 were simulated using Mason[3] [30] with different parameters, *e.g.* percentage of mismatches, read length. We apply QCluster using different distances, to the whole set of reads and then we measure the quality of the clusters produced by evaluating the extent to which the partitioning agrees with the natural splitting of the sequences. In other words, we measured how well reads originating from the same sequence are grouped together. We calculate the recall rate as follows, for each mRNA sequence S we identified the set of reads originated from S. We looked for the cluster C that contains most of the reads of S. The percentage of the S reads that have been grouped in C is the recall value for the sequence S. We repeat the same operation for each sequence and calculate the average value of recall rate over all sequences.

Several clustering were produced by using the following distance types: D_2^*, D_2, L_2, KL, symmetrized KL and compared with D_2^{*q} in all its variants, using the expectation formula (1) AWP or (2) AQP, with and without quality redistribution (q-red). In order to avoid as much as possible biases due to the initial random generation of centroids, each algorithm was executed 5 times with different random seeds and the clustering with the lower distortion was chosen.

Table 1 reports the recall while varying error rates, number of clusters and the parameters k. For all distances the recall rate decreases with the number of clusters, as expected. For traditional distances, if the reads do not contain errors then D_2^* preforms consistently better then the others D_2, L_2, KL. When the sequencing process becomes more noisy, the KL distances appears to be less sensitive to sequencing errors. However if quality information are used, D_2^{*q} outperforms all other methods and the advantage grows with the error rate. This confirms that the use of quality values can improve clustering accuracy. When the number of clusters increases then the advantage of D_2^{*q} becomes more evident. In these experiments the use of AQP for expectation within D_2^{*q} is more stable and better performing compared with formula AWP. The contribution of

[1] `http://www.dei.unipd.it/~ciompin/main/qcluster.html`
[2] `ftp://ftp.ncbi.nlm.nih.gov/refseq/H-sapiens/mRNA-Prot/`
[3] `http://seqan.de/projects/mason.html`

Table 1. Recall rates of clustering of mRNA simulated reads (10000 reads of length 200) for different measures, error rates, number of clusters and parameter k

Distance	No Errors	3%	5%	10%	No Errors	3%	5%	10%
	2 clusters				2 clusters			
D_2^*	**0,815**	0,813	0,810	0,801	**0,822**	0,819	0,814	0,794
D_2^{*q} AQP	**0,815**	**0,815**	**0,813**	**0,810**	**0,822**	**0,822**	**0,820**	**0,809**
D_2^{*q} AQP q-red	**0,815**	**0,815**	**0,813**	**0,810**	**0,822**	**0,822**	**0,820**	0,807
D_2^{*q} AWP	0,809	0,806	0,805	0,802	0,809	0,807	0,805	0,802
D_2^{*q} AWP q-red	0,809	0,806	0,805	0,802	0,809	0,807	0,805	0,802
L_2	0,811	0,807	0,806	0,801	0,810	0,806	0,805	0,801
KL	0,812	0,809	0,807	0,802	0,812	0,809	0,807	0,802
Symm, KL	0,812	0,809	0,807	0,802	0,812	0,808	0,806	0,802
D_2	0,811	0,807	0,806	0,801	0,809	0,806	0,805	0,800
	3 clusters				3 clusters			
D_2^*	**0,695**	0,689	0,683	0,662	**0,717**	0,707	0,697	0,668
D_2^{*q} AQP	**0,695**	**0,696**	**0,696**	0,689	**0,717**	0,711	**0,705**	0,679
D_2^{*q} AQP q-red	**0,695**	**0,696**	**0,696**	**0,691**	**0,717**	**0,712**	0,704	**0,681**
D_2^{*q} AWP	0,653	0,646	0,646	0,638	0,668	0,662	0,655	0,646
D_2^{*q} AWP q-red	0,653	0,646	0,645	0,637	0,668	0,662	0,655	0,644
L_2	0,682	0,673	0,671	0,657	0,685	0,677	0,674	0,663
KL	0,694	0,687	0,685	0,672	0,696	0,689	0,687	0,675
Symm, KL	0,693	0,686	0,684	0,669	0,695	0,688	0,685	0,673
D_2	0,675	0,668	0,662	0,654	0,675	0,671	0,665	0,655
	4 clusters				4 clusters			
D_2^*	**0,623**	0,613	0,606	0,574	0,627	0,616	0,591	0,551
D_2^{*q} AQP	0,622	0,621	0,618	0,602	**0,628**	**0,617**	0,602	0,572
D_2^{*q} AQP q-red	0,622	**0,622**	**0,619**	**0,605**	**0,628**	**0,617**	**0,603**	**0,573**
D_2^{*q} AWP	0,580	0,563	0,566	0,535	0,582	0,571	0,572	0,555
D_2^{*q} AWP q-red	0,580	0,560	0,565	0,533	0,582	0,570	0,570	0,555
L_2	0,554	0,551	0,547	0,540	0,568	0,565	0,553	0,543
KL	0,555	0,548	0,545	0,536	0,566	0,558	0,547	0,537
Symm, KL	0,556	0,549	0,546	0,538	0,562	0,554	0,547	0,539
D_2	0,553	0,547	0,547	0,538	0,556	0,549	0,548	0,540
	5 clusters				5 clusters			
D_2^*	0,553	0,539	0,532	0,500	0,560	0,534	0,512	0,462
D_2^{*q} AQP	**0,554**	**0,545**	**0,551**	0,532	0,560	0,544	0,524	**0,489**
D_2^{*q} AQP q-red	0,553	0,544	0,550	**0,533**	**0,561**	**0,545**	**0,531**	0,487
D_2^{*q} AWP	0,483	0,475	0,470	0,463	0,509	0,494	0,485	0,470
D_2^{*q} AWP q-red	0,483	0,475	0,470	0,461	0,509	0,494	0,482	0,470
L_2	0,478	0,472	0,465	0,453	0,500	0,495	0,486	0,465
KL	0,498	0,488	0,484	0,468	0,507	0,501	0,492	0,476
Symm, KL	0,498	0,488	0,484	0,468	0,507	0,500	0,491	0,474
D_2	0,470	0,464	0,457	0,449	0,488	0,482	0,476	0,455
	$k = 2$				$k = 3$			
	(a)				(b)			

quality redistribution (q-red) is limited, although it seems to have some positive effect with the expectation AQP.

The future generation sequencing technologies will produce long reads with a large number of erroneous bases. To this end we study how read length affects these measures. Since the length of sequences under investigation is limited we keep the read length under 400 bases. In Table 2 we report some experiments for the setup with 4 clusters and $k = 3$, while varying the error rate and read length. If we compare these results with Table 1, where the read length is 200, we can observe a similar behavior. As the error rate increases the improvement with respect to the other measures remains evident, in particular the difference in terms of recall of D_2^{*q} with the expectations AQP grows with the length of reads when compared with KL (up to 9%), and it remains constant when compared with D_2^*. With the current tendency of the future sequencing technologies to produce longer reads this behavior is desirable. These performance are confirmed also for other setups with larger k and higher number of clusters (data not shown).

Table 2. Recall rates for clustering of mRNA simulated reads (10000 reads, $k = 3$, 4 clusters) for different measures, error rates and read length

Distance	No Errors	3%	5%	10%	No Errors	3%	5%	10%
	4 clusters				4 clusters			
D_2^*	**0,680**	0,667	0,658	0,625	**0,713**	0,700	0,697	0,672
D_2^{*q} AQP	**0,680**	**0,672**	**0,673**	**0,650**	**0,713**	**0,712**	0,710	0,693
D_2^{*q} AQP q-red	**0,680**	0,671	**0,673**	**0,650**	**0,713**	0,711	**0,711**	**0,694**
D_2^{*q} AWP	0,616	0,610	0,608	0,601	0,643	0,636	0,632	0,623
D_2^{*q} AWP q-red	0,616	0,610	0,607	0,602	0,643	0,635	0,631	0,622
L_2	0,610	0,600	0,602	0,581	0,638	0,630	0,624	0,614
KL	0,617	0,604	0,601	0,577	0,649	0,632	0,628	0,618
Symm, KL	0,613	0,603	0,599	0,576	0,647	0,632	0,627	0,616
D_2	0,601	0,593	0,588	0,575	0,626	0,618	0,615	0,604
	read length=300				read length=400			
	(a)				(b)			

5.1 Boosting Assembly

Assembly is one of the most challenging computational problems in the field of NGS data. It is a very time consuming process with highly variable outcomes for different datasets [31]. Currently large datasets can only be assembled on high performance computing systems with considerable CPU and memory resources. Clustering has been used as preprocessing, prior to assembly, to improve memory requirements as well as the quality of the assembled contigs [20, 21]. Here we test if the quality of assembly of real read data can be improved with clustering. For the assembly component we use Velvet [32], one of the most popular assembly tool for NGS data. We study the *Zymomonas mobilis* genome and download

as input the reads dataset *SRR017901* (454 technology) with 23.5Mbases corresponding to 10× coverage. We apply the clustering algorithms, with $k = 3$, and divide the dataset of reads in two clusters. Then we produce an assembly, as a set of contigs, for each cluster using Velvet and we merged the generated contigs. In order to evaluate the clustering quality, we compare this merged set with the assembly, without clustering, using of the whole set of reads. Commonly used metrics such as number of contigs, $N50$ and percentage of mapped contigs are presented in Table 3. When merging contigs from different clusters, some contig might be very similar or they can cover the same region of the genome, this can artificially increase these values. Thus we compute also a less biased measure that is the percentage of the genome that is covered by the contigs (last column).

Table 3. Comparison of assembly with and without clustering preprocess ($k = 3$, 2 clusters). The assembly with Velvet is evaluated in terms of mapped contigs, N50, number of contigs and genome coverage. The dataset used is SRR017901 (23.5M bases, 10x coverage) that contains reads of *Zymomonas mobilis*.

Distance	Mapped Contigs	N50	Number of Contigs	Genome Coverage
No Clustering	93.55%	112	22823	0,828
D_2^*	93.97%	138	28701	0,914
D_2^{*q} AQP	94.09%	141	29065	**0,921**
D_2^{*q} AQP q-red	94.13%	**141**	**29421**	0,920
D_2^{*q} AWP	**94.36%**	137	28425	0,907
D_2^{*q} AWP q-red	**94.36%**	137	28549	0,908
L_2	94.24%	135	28297	0,904
KL	94.19%	135	28171	0,903
Symm, KL	94.27%	134	27999	0,902
D_2	94.33%	134	28019	0,903

In this set of experiments the introduction of clustering as a preprocessing step increases the number of contigs and the N50. More relevant is the fact that the genome coverage is incremented by 10% with respect to the assembly without clustering. The relative performance between the distance measures is very similar to the case with simulated data. In fact D_2^{*q} with expectation AQP and quality redistribution is again the best performing. More experiments should be conducted in order to prove that assembly can benefit from the clustering preprocessing. However this first preliminary tests show that, at least for some configuration, a 10% improvement on the genome coverage can be obtained.

The time required to performed the above experiments are in general less than a minute on a modern laptop with an Intel i7 and 8Gb of ram. The introduction of quality values typically increases the running time by 4% compared to standard alignment-free methods. The reads dataset *SRR017901* is about 54MB and the memory required to cluster this set is 110MB. Also in the other experiments the memory requirements remain linear in the input size.

6 Conclusions

The comparison of reads with quality values is essentials in many genome projects. The importance of quality values will increase in the near future with the advent of future sequencing technologies, that promise to produce long reads, but with 15% errors. In this paper we presented a family of alignment-free measures, called D^q-type, that incorporate quality value information and k-mers counts for the comparison of reads data. A set of experiments on simulated and real reads data confirms that the new measures are superior to other classical alignment-free statistics, especially when erroneous reads are considered. If quality information are used, D_2^{*q} outperforms all other methods and the advantage grows with the error rate and with the length of reads. This confirms that the use of quality values can improve clustering accuracy.

Preliminary experiments on real reads data show that the quality of assembly can also improve when using clustering as preprocessing. All these measures are implemented in a software called QCluster. As a future work we plan to explore other applications like genome diversity estimation and meta-genome assembly in which the impact of reads clustering might be substantial.

Acknowledgments. M. Comin was partially supported by the Ateneo Project CPDA110239 and by the P.R.I.N. Project 20122F87B2.

References

1. Medini, D., Serruto, D., Parkhill, J., Relman, D., Donati, C., Moxon, R., Falkow, S., Rappuoli, R.: Microbiology in the post-genomic era. Nature Reviews Microbiology 6, 419–430 (2008)
2. Jothi, R., et al.: Genome-wide identification of in vivo protein-DNA binding sites from ChIP-Seq data. Nucleic Acids Res. 36, 5221–5231 (2008)
3. Altschul, S., Gish, W., Miller, W., Myers, E.W., Lipman, D.: Basic local alignment search tool. Journal of Molecular Biology 215(3), 403–410 (1990)
4. Sims, G.E., Jun, S.-R., Wu, G.A., Kim, S.-H.: Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. PNAS 106(8), 2677–2682 (2009)
5. Comin, M., Verzotto, D.: Whole-genome phylogeny by virtue of unic subwords. In: Proc. 23rd Int. Workshop on Database and Expert Systems Applications (DEXA-BIOKDD 2012), pp. 190–194 (2012)
6. Comin, M., Verzotto, D.: Alignment-free phylogeny of whole genomes using underlying subwords. BMC Algorithms for Molecular Biology 7(34) (2012)
7. Song, K., Ren, J., Zhai, Z., Liu, X., Deng, M., Sun, F.: Alignment-Free Sequence Comparison Based on Next-Generation Sequencing Reads. Journal of Computational Biology 20(2), 64–79 (2013)
8. Comin, M., Schimd, M.: Assembly-free Genome Comparison based on Next-Generation Sequencing Reads and Variable Length Patterns. Accepted at RECOMB-SEQ 2014: 4th Annual RECOMB Satellite Workshop at Massively Parallel Sequencing. Proceedings to appear in BMC Bioinformatics (2014)

9. Vinga, S., Almeida, J.: Alignment-free sequence comparison – a review. Bioinformatics 19(4), 513–523 (2003)
10. Gao, L., Qi, J.: Whole genome molecular phylogeny of large dsDNA viruses using composition vector method. BMC Evolutionary Biology 7(1), 41 (2007)
11. Qi, J., Luo, H., Hao, B.: CVTree: a phylogenetic tree reconstruction tool based on whole genomes. Nucleic Acids Research 32 (Web Server Issue), 45–47 (2004)
12. Goke, J., Schulz, M.H., Lasserre, J., Vingron, M.: Estimation of pairwise sequence similarity of mammalian enhancers with word neighbourhood counts. Bioinformatics 28(5), 656–663 (2012)
13. Kantorovitz, M.R., Robinson, G.E., Sinha, S.: A statistical method for alignment-free comparison of regulatory sequences. Bioinformatics 23(13), 249–255 (2007)
14. Comin, M., Verzotto, D.: Beyond fixed-resolution alignment-free measures for mammalian enhancers sequence comparison. Accepted for presentation at The Twelfth Asia Pacific Bioinformatics Conference. Proceedings to appear in IEEE/ACM Transactions on Computational Biology and Bioinformatics (2014)
15. Comin, M., Antonello, M.: Fast Computation of Entropic Profiles for the Detection of Conservation in Genomes. In: Ngom, A., Formenti, E., Hao, J.-K., Zhao, X.-M., van Laarhoven, T. (eds.) PRIB 2013. LNCS, vol. 7986, pp. 277–288. Springer, Heidelberg (2013)
16. Comin, M., Antonello, M.: Fast Entropic Profiler: An Information Theoretic Approach for the Discovery of Patterns in Genomes. IEEE/ACM Transactions on Computational Biology and Bioinformatics 11(3), 500–509 (2014)
17. Comin, M., Verzotto, D.: Classification of protein sequences by means of irredundant patterns. Proceedings of the 8th Asia-Pacific Bioinformatics Conference (APBC), BMC Bioinformatics 11(Suppl.1), S16 (2010)
18. Comin, M., Verzotto, D.: The Irredundant Class method for remote homology detection of protein sequences. Journal of Computational Biology 18(12), 1819–1829 (2011)
19. Hashimoto, W.S., Morishita, S.: Efficient frequency-based de novo short-read clustering for error trimming in next-generation sequencing. Genome Research 19(7), 1309–1315 (2009)
20. Bao, E., Jiang, T., Kaloshian, I., Girke, T.: SEED: efficient clustering of next-generation sequences. Bioinformatics 27(18), 2502–2509 (2011)
21. Solovyov, A., Lipkin, W.I.: Centroid based clustering of high throughput sequencing reads based on n-mer counts. BMC Bioinformatics 14, 268 (2013)
22. Heng, L., Jue, R., Durbin, R.: Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Research 18, 1851–1858 (2008)
23. Albers, C., Lunter, G., MacArthur, D.G., McVean, G., Ouwehand, W.H., Durbin, R.: Dindel: accurate indel calls from short-read data. Genome Research 21(6), 961–973 (2011)
24. Carneiro, M.O., Russ, C., Ross, M.G., Gabriel, S.B., Nusbaum, C., DePristo, M.A.: Pacific biosciences sequencing technology for genotyping and variation discovery in human data. BMC Genomics 13, 375 (2012)
25. Blaisdell, B.E.: A measure of the similarity of sets of sequences not requiring sequence alignment. PNAS USA 83(14), 5155–5159 (1986)
26. Lippert, R.A., Huang, H.Y., Waterman, M.S.: Distributional regimes for the number of k-word matches between two random sequences. Proceedings of the National Academy of Sciences of the United States of America 100(13), 13980–13989 (2002)
27. Reinert, G., Chew, D., Sun, F., Waterman, M.S.: Alignment-free sequence comparison (I): statistics and power. Journal of Computational Biology 16(12), 1615–1634 (2009)

28. Wan, L., Reinert, G., Chew, D., Sun, F., Waterman, M.S.: Alignment-free sequence comparison (II): theoretical power of comparison statistics. Journal of Computational Biology 17(11), 1467–1490 (2010)
29. Ewing, B., Green, P.: Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Research 8(3), 186–194 (1998)
30. Holtgrewe, M.: Mason–a read simulator for second generation sequencing data. Technical Report FU Berlin (2010)
31. Birney, E.: Assemblies: the good, the bad, the ugly. Nature Methods 8, 59–60 (2011)
32. Zerbino, D.R., Birney, E.: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Research 18, 821–829 (2008)