Assemble CRISPRs from metagenomic sequencing data

Jikai Lei and Yanni Sun*

Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

*To whom correspondence should be addressed.

Abstract

Motivation: Clustered regularly interspaced short palindromic repeats and associated proteins (CRISPR-Cas) allows more specific and efficient gene editing than all previous genetic engineering systems. These exciting discoveries stem from the finding of the CRISPR system being an adaptive immune system that protects the prokaryotes against exogenous genetic elements such as phages. Despite the exciting discoveries, almost all knowledge about CRISPRs is based only on microorganisms that can be isolated, cultured and sequenced in labs. However, about 95% of bacterial species cannot be cultured in labs. The fast accumulation of metagenomic data, which contains DNA sequences of microbial species from natural samples, provides a unique opportunity for CRISPR annotation in uncultivable microbial species. However, the large amount of data, heterogeneous coverage and shared leader sequences of some CRISPRs pose challenges for identifying CRISPRs efficiently in metagenomic data.

Results: In this study, we developed a CRISPR finding tool for metagenomic data without relying on generic assembly, which is error-prone and computationally expensive for complex data. Our tool can run on commonly available machines in small labs. It employs properties of CRISPRs to decompose generic assembly into local assembly. We tested it on both mock and real metagenomic data and benchmarked the performance with state-of-the-art tools.

Availability and Implementation: The source code and the documentation of metaCRISPR is available at https://github.com/hangelwen/metaCRISPR.

Contact: yannisun@msu.edu

1 Introduction

Clustered regularly interspaced short palindromic repeats (CRISPRs), together with the associated cas genes, constitute a prokaryotic adaptive immune system that protects against exogenous genetic elements such as viruses and plasmids (Andersson and Banfield, 2008; Barrangou et al., 2007; Cui et al., 2008; Horvath and Barrangou, 2010; Horvath et al., 2008; Marraffini and Pennisi, 2013; Sontheimer, 2008). CRISPRs are found in a large number of prokaryotic genomes (approximately 40% of sequenced bacteria and almost all sequenced archaea (Grissa et al., 2007)). Recent advances of studying CRISPRs in prokaryotic organisms have led to promising biotechnology applications such as artificial immunization and accurate gene editing (silencing, enhancing or changing specific genes) (Cong et al., 2013; DeCarlo et al., 2013; Friedland et al., 2013; Hale et al., 2012; Hwang et al., 2013; Jiang et al., 2013; Larson et al., 2013; Sorek et al., 2008; Wang et al., 2013).

Each CRISPRs locus contains a set of variable spacers regularly separated by direct repeats (DRs), which are highly conserved within species and have typical size range of 23–47 bp. A 100 bp–500 bp long DNA sequence called ‘leader’ is always found on one side of the CRISPR. The leader acts as a promoter and directs the transcription of the CRISPR, which is essential for the CRISPR system to confer specific resistance to viruses whose genomic sequences match one spacer (Andersson and Banfield, 2008; Haft et al., 2005; Pougach et al., 2010). Figure 1 shows an example of a cas-CRISPR system.

Most known CRISPRs are annotated from sequenced genomes of bacteria and archaea. However, only a small percentage of bacterial can be cultured and sequenced in labs, limiting a comprehensive characterization of CRISPRs. Metagenomic data, which contain community shotgun sequences of a large number of uncultured microbial species from variable natural samples, provides a unique opportunity to thoroughly annotate cas-CRISPRs and to enhance our understanding of the phage-host interactions (Andersson and Banfield, 2008; Rho et al., 2012; Skennerton et al., 2013). A number of groups searched for CRISPRs in metagenomic data such as those sequenced from human body (Rho et al., 2012), human gut (Stern et al., 2012), saliva (Pride et al., 2011), natural acidophilic biofilms (Andersson and Banfield, 2008), ocean (Sorokin et al., 2010), etc.
As existing tools for identifying CRISPRs are developed for sequenced genomes (Bland et al., 2007; Edgar, 2007; Rho et al., 2012) rather than short reads, most studies in metagenomic data still rely on the combination of generic de novo assembly and genome-wide CRISPR finding. This combined pipeline is convenient to use but is not optimized for detecting CRISPRs in metagenomic data. One reported problem is that CRISPRs loci found in metagenomic assemblies may only represent the most dominant strain in the community, and not indicative of the true spacer diversity (Skennerton et al., 2013). In addition, some special properties of CRISPRs are not carefully considered by the generic assembly programs. For example, many CRISPRs that share the same repeat in a species have highly conserved leader sequences (Jansen et al., 2002; Karginov and Hannon, 2010; Wei et al., 2015). As the leader sequence can be longer than a read, a very tangled assembly graph can be formed. Finally, generic assembly has high demand on memory and CPU resources (Martin and Wang, 2011), particularly for complicated metagenomic data such as those sequenced from soil.

Thus it is ideal to have dedicated tools for identifying CRISPRs in metagenomic data without assembling whole genomes. The major tool in this category is Crass (Skennerton et al., 2013). Crass identifies reads that belong to CRISPRs using the repeat structural characteristic of CRISPRs and uses a graph based algorithm to recover the spacer arrangement of CRISPRs that share the same direct repeat sequence. As a final step to output the full-length CRISPR sequences, Crass requires users to parse the information output by the previous step and input it to existing de novo assembly programs such as Velvet (Zerbino and Birney, 2008). This step is not straightforward for complex data. In addition, the supported assemblers Velvet and cap3 (15) are genome assemblers and are not designed to address specific assembly challenges for CRISPRs. Finally, our empirical results show that Crass tends to fail to identify CRISPR repeat when read length is short or when repeat-spacer unit is long (Ben-Bassat and Chor, 2015). Thus, there is a need for alternative algorithms and tools that can output assembled CRISPR sequences from metagenomic data automatically.

In this study, we design and develop an efficient CRISPR identification method (named metaCRISPR) that can run on commonly available machines in small labs. We use a divide-and-conquer strategy to decompose generic assembly into local assembly for individual CRISPRs. Our methods can identify and assemble sequences of CRISPRs that contain both known and novel repeats in metagenomic data.

2 Methods

Figure 2 sketches the main algorithmic components in metaCRISPR. It consists of three major components: filtration, clustering and assembly. First, we reduce the problem size by identifying CRISPR reads from the original metagenomic dataset. This starts with identifying DR sequences using the repetitive property of DR sequences and paired-end read information. Then all reads that contain the identified DR sequences and also known repeats such as those in CRISPRdb (Grissa et al., 2007) are kept as input to the next stage. All other reads will be removed. This step removes a large portion of reads that are not part of CRISPRs and significantly reduces input size for downstream analysis. In the second component, we cluster reads belonging to the same CRISPRs using reads’ overlaps and paired-end reads support. Clustering will further reduce memory usage for de novo assembly and enables parallel assembly on all clusters. In the final stage, we apply overlap graph-based iterative assembly in order to recover CRISPRs of heterogeneous coverage. After assembly, a simple validation step is applied to discard contigs that do not satisfy properties of CRISPRs.

2.1 Screening reads sequenced from CRISPRs

Our pipeline first identifies reads that are likely sequenced from CRISPR regions. Direct repeats (DRs), which share very high sequence similarity and are evenly spaced in a CRISPR locus, is the main feature for identifying reads from CRISPRs. One simple and commonly used method for identifying DRs from short reads is to search for repetitive sequences that meet properties of DRs.

2.1.1 De novo repeat identification using paired-end reads

Crass identifies repeats by detecting identical k-mers in a read, where the default k-mer size is 23. Then the identical k-mers are extended until a mismatch is found. The extensions are putative DRs. The length of the repeat sequences and the distance between them are checked to make sure they meet requirement of CRISPRs.
This method is effective but it fails when repeat-spacer unit is relatively long compared to the read length. In that case, the assumption that a read must contain at least two repeats will not hold and those repeats will not be detected. The problem associated with this approach was observed by previous study (Ben-Bassat and Chor, 2015).

In this study, we extend Crass’ DR identification method by using information from paired-end reads. When performing paired-end sequencing, one chooses a target insert size (the distance from the first base of one end to the last base of the other end). For example, for Illumina TrueSeq sequencer, the library preparation reference guide provides two recommended insert sizes, 350 nt and 550 nt, which is long enough to cover part of a CRISPR repeat-spacer array for many CRISPRs genes. In our approach, instead of searching DR in a single read without considering paired-end read information, we search DRs in both ends. Specifically, we do the following:

- For each paired-end reads \( r_1 \) and \( r_2 \), and a user specified k-mer size \( k \), metaCRISPR first searches for a pair of matching k-mers of size \( k \) between \( r_1 \) and \( r_2 \), starting from the beginning of the two ends. If such two k-mers are found, they are extended until the extended regions contain two mismatches. The extended regions are regarded as putative DRs if their lengths satisfy the length requirement of DRs specified by the user.
- If the first step fails (e.g. no matching k-mers are found between \( r_1 \) and \( r_2 \)), \( r_1 \) and \( r_2 \) are examined individually without considering paired-end information. If one end or both ends contain two copies of a k-mer, the same extension strategy as in the first step is conducted. In addition to checking the length of the repeat, the distance between the two repetitive regions needs to meet the requirement of the length of spacers.

Searching repeats in paired-end reads enables metaCRISPR to detect DR sequences when read length is short or DR-spacer unit is long. Our experimental results show that this strategy can recover most DRs.

Including DRs from existing databases

While the de novo repeat identification process can recover a majority of repeat containing reads, some DRs may be missed if they do not satisfy the specified parameters such as the length ranges of DRs or spacers. MetaCRISPR also allows users to input known repeats such as those in CRISPRdb. Direct repeats in annotated CRISPRs loci are effective templates for identifying CRISPRs reads because repeats are well-conserved among CRISPRs in the same species or even between different species (Kunin et al., 2007). By including repeats from known databases, we are able to identify some of the repeats that are missed by the de novo repeat identification process and thus further increase the sensitivity of the pipeline.

2.1.2 Screen metagenomic data using k-mer hash table

Using the previous step, we can identify repeats and recruit reads that contain at least two copies of DRs or paired-end reads with both ends containing DRs. However, there are still a number of reads containing only part of a DR. It’s also possible that for a paired-end read, one end of the read contains DR, the other end does not contain the complete DR. These reads cannot be identified using the previous method. But as we have already detected DRs in this dataset, we can recruit these reads by searching for identified and known DRs in them.

To build the efficient procedure, we use a k-mer hash table based method to detect reads that contain only one copy of DR or partial DR. We construct a hash table for all the k-mers of the known and newly identified direct repeats. For each k-mer, both the k-mer and the reverse complementary sequence of the k-mer are kept in the hash table. The original metagenomic dataset is scanned to identify reads that contain at least one k-mer in the hash table. For paired-end reads, if one end has a matching k-mer, the other end is automatically included regardless of whether it has a matching k-mer. All reads from this step are passed to the next step. This step can rapidly identify additional reads that are likely to be sequenced from CRISPRs.

It should be noted that both the repeat identification step and the k-mer hash table based filtration step inevitably recruit a small portion of reads that do not belong to any CRISPR. These false positive reads slightly increase the input size to the next step but will be removed during either the clustering step or the assembly/validation step.

2.2 Clustering of CRISPR reads

The CRISPR reads identification step produces a mixture of reads that are likely sequenced from CRISPRs in different species or different CRISPRs of the same species. To further reduce the memory usage of de novo assembly and enable parallel processing, we divide reads into different clusters with the goal of each cluster representing one CRISPR loci.

2.2.1 Overlap based CRISPR read clustering

We cluster reads if their overlap is larger than a given threshold. To separate CRISPRs, we use two important properties specific to CRISPRs:

- Spacers are usually unique. For a CRISPR locus, the array of spacers is quite different from each other. They also have very low similarity with the repeat sequence. In addition, spacers are not conserved across different CRISPRs.
- Leader sequences of CRISPRs that share the same repeat in one species may have high similarity, but they are usually quite different for CRISPRs from different species, even when these CRISPRs share the same repeat sequence (Jansen et al., 2002; Karginov and Hannon, 2010; Wei et al., 2015).

From the first property, we know that random overlaps caused by near identical repeats cannot exceed the length of a typical repeat. For reads that start or end within repeats, they tend to form random overlaps with each other. However, as the spacers are usually unique, the overlap rarely extends beyond the repeats. Thus, by applying an overlap threshold that is much larger than the typical repeat size, a majority of random overlaps will be removed. In addition, it is rare that two different CRISPRs will produce the same read unless the read is completely sequenced from the repeats. In practice, as the length of reads is usually much longer than repeats, the read will cover at least part of the spacer, which can distinguish reads from different loci.

From the second property, we know that reads sequenced from the leader regions of different CRISPRs that share repeat sequence in the same species may form large overlaps. We are unable to separate those even with a large overlap threshold. However, because overlap identification does not allow mismatches, most CRISPRs in the same species can still be separated into different clusters as long as the leader sequences are not exactly the same. This is true for a large portion of CRISPRs that share repeat sequence in the same
species (Karginov and Hannon, 2010). Second, if several CRISPRs are clustered into the same cluster, we will separate them in the assembly stage.

Based on these CRISPR specific properties, we build the clusters with a large overlap threshold (The default value is 80 nt or 80% of the length of the reads, whichever is smaller).

### Merging clusters using paired end reads support

We are aware that reads from the same CRISPR may be separated into several initial clusters. This happens for the following reasons. First, there are species of heterogeneous abundance in a metagenomic sample. For rare species, the sequencing depth is much lower than the abundant ones. The reads sequenced from rare species tend to form small overlaps and thus cannot be clustered into the same set. Second, even for a highly abundant CRISPR, lowly sequenced regions may exist locally and prevent all reads from the CRISPR joining a single cluster. Thus, we conduct a merging step to combine clusters whose reads are sequenced from the same CRISPR based on paired-end reads support.

Specifically, we build an undirected graph in which a node represents an initial cluster whose size is larger than one. An edge is formed between two nodes $v_1$ and $v_2$ if the corresponding clusters have enough paired-end reads support. A read pair supports nodes $v_1$ and $v_2$ if the two ends are located in $v_1$ and $v_2$, respectively. Then all initial clusters in one connected component in the graph form a new cluster.

#### 2.3 Assemble CRISPRs of heterogeneous coverage

Overlap graph based clustering divides reads into different clusters. Reads in each cluster are assembled separately. Ideally each cluster should contain reads sequenced from the same CRISPR, making de novo assembly much more straightforward than whole genome assembly. While most clusters belong to this category, some can contain reads from multiple CRISPRs of the same species because these CRISPRs share similar leader sequences. In addition, note that some reads in the same cluster do not share large overlaps because they are merged into the same cluster based on paired-end read support. Thus, in order to distinguish and assemble different CRISPRs with heterogeneous coverage, we now need to build an overlap using a small threshold, leading to a highly tangled graph that could even contain cycles.

One such example is shown in Figure 3. The graph contains three CRISPRs, of which one has high sequence coverage (red), and the other two have low sequence coverage. These three CRISPRs are in the same species and share leader sequences and very similar repeat sequences. Thus, using a small overlap threshold, there are many edges generated by random overlaps in the initial overlap graph. These random edges in the same CRISPR and between different CRISPRs loci lead to highly tangled and cyclic graphs. We will illustrate the major steps of our algorithm using this example. Our goal is to recover the three correct paths corresponding to the three CRISPRs loci. We formulate CRISPR assembly problem as finding the longest simple paths with support from paired-end reads in the overlap graph.

From the CRISPR specific properties stated in Section 2.2.1, we know that by applying an overlap threshold larger than the typical repeat size, a majority of random overlaps can be removed. We refer edges, whose overlap size is larger than threshold $\tau$ as **confident edges**, where $\tau >$ repeat size. In addition, once a read is used in assembled CRISPRs, we remove them from the graph. As read length is usually larger than the typical repeat size, it is highly unlikely that the same read is sequenced from two different CRISPRs. Based on these observations, we propose the following method to assemble CRISPRs.

#### 2.3.1 Step 1: clean graph

First, we apply existing graph pruning techniques to remove tips and bubbles (Zerbino and Birney, 2008), which are usually caused by sequencing errors. Second, we clean the graph using confident edges for nodes with many bifurcations. For a node $n$ with multiple outgoing edges, only confident ones will be kept. For example, in Figure 3, the edge from $b_3$ to $b_4$ is a confident edge, thus edge from $b_3$ to $c_2$ and the edge from $b_3$ to $b_2.2$ can be removed.

#### 2.3.2 Step 2: break cycles

For almost all datasets, the sequencing depths of different species are highly uneven. This leads to the coexistence of both highly sequenced and lowly sequenced CRISPRs in a dataset. A lowly sequenced CRISPR has a lot true overlaps that have similar sizes with the random overlaps, and cannot be easily distinguished using coverage information. Even worse, the random overlaps formed by repeats within the same loci or between two CRISPRs of low coverage may form cycles in a highly tangled graph, which makes finding longest path difficult. In this step, we will further simplify the cyclic graph $G_2$ by breaking cycles using information provided by paired end reads. First, we make the following assumptions:

- **Assumption 1:** for a cycle, there exists at least one read pair such that $R.1$ (or $R.2$) is a node in the cycle, and $R.2$ (or $R.1$) is a node out of the cycle. When the cycle has this property, we can break the cycle using paired-end read information.
- **Assumption 2:** if reads $R.1$, $R.2$ are paired-end reads, and $R.1$ and $R.2$ are in the same weakly connected component of an overlap graph, then there must be a path between $R.1$ and $R.2$. A directed graph is weakly connected if replacing all of its directed
edges with undirected edges produces a connected undirected graph. This assumption is not strong because the overlap graph at this stage is constructed by small overlap threshold. Thus, there usually is a path between paired-end reads as they are sequenced from the same fragment.

If the first assumption does not hold for some cycles, we simply do a deep first traversal of the graph and remove all back-track edges.

In order to remove a cycle, we find the longest path between paired-end reads whose two ends are in and out of the cycle, respectively. Once the path is identified, all edges in the cycle but not in the path will be removed.

Figure 4 shows two cases in cycle breaking and the steps to break the cycle in case one. The two cases differ by which end of the read pair is in the cycle. In case 1, the read in the cycle (a1.1) is at the 5′ end of the fragment respective to its mate (i.e. the other end of the read pair). In case 2, the read in the cycle (a1.2) is at the 3′ end of the fragment respective to its mate. For case 1, we will need to find the longest path from a1.1 to a1.2 and thus will break cycle by temporarily removing all edges entering a1.1. In case 2, as a1.2 is at the 3′ end, we will temporarily remove all edges leaving a1.2. All the other operations for two cases are exactly the same.

The right panel of Figure 4 shows an example of how to break a cycle in case 1. e2 is temporarily removed because it is an edge entering a1.1. The graph becomes a DAG, the longest path between a1.1 and a1.2 can be easily found (the path formed by red edges). e5 is removed because it’s a branching edge on the path. After adding e2 back, we can find a longest path that contains e2, e1, e4 and e3.

2.3.3 Step 3: iterative assembly
After all cycles in the graph are broken, the graph becomes a DAG that harbors one or more CRISPRs. MetaCRISPR assembles CRISPRs by iteratively finding longest paths in the DAG. Given the observation that different CRISPRs do not share the same reads, all the reads, their corresponding nodes, and associated edges from recovered CRISPRs can be removed from the graph. In addition, we use paired-end reads support to guide the path finding. The path that is not well supported by paired end reads will not be output. For example, in Figure 3, two confident edges leave node a6. The path containing edge a6 to a7 will finally be chosen because it contains paired end supports (one example is given between a4.1 and a4.2).

A post-processing can be done to handle a very rare case when the same CRISPR is disconnected (after only keeping confident edges) because the overlap between two nodes is smaller than $r$. The paths that represent segments of a disconnected CRISPR can be connected using paired-end reads information. For example, in Figure 3, node a7 and a8 can be connected using information from paired-end reads a4.1 and a4.2.

2.3.4 Post-processing
After the assembly, all contigs longer than a user defined threshold $L$ is validated. For each contig, the validation step first identifies repeat sequences in the contig, then validates the size of the repeats, the size of the spacers and the number of repeat sequences. If all of them meet the user defined values, the contig is classified as a CRISPR. MetaCRISPR uses a modified version of CTR to do the validation. Users can also use other genome-wide CRISPR detection tools such as PILER-CR, CRISPRFinder, to validate the contigs.

3 Results
To evaluate the performance of the proposed method, we applied our implementation, metaCRISPR, to three datasets. The three datasets contain a mock community dataset that was sequenced from synthetic communities (mock), a complex corn Rhizosphere soil metagenomic dataset (soil) and a human gut dataset (gut). The number of reads, read length and the size of each dataset can be found in Table 1.

| Dataset | Reads | Read length | Size | Recruited reads | Sensitivity | Filtration Rate | Time (min) | Memory (g) |
|---------|-------|-------------|------|----------------|-------------|----------------|------------|------------|
| mock    | 105m  | 101         | 34G  | 1 194 816      | 98%         | 1.14%          | 19.3       | 7.9        |
| soil    | 271m  | 150         | 117G | 5 642 746      | N/A$^a$     | 2.08%          | 108        | 13.5       |
| gut     | 28m   | 151         | 7.3G | 159 219        | N/A$^b$     | 0.57%          | 5.7        | 1.4        |

$^a$Walltime using four threads, average of three runs.
$^b$No ground truth so sensitivity cannot be calculated.
3.1 Evaluating the performance of identifying reads sequenced from CRISPRs
The first step (i.e. filtration step) of our method is to identify reads that are sequenced from CRISPRs. In this section, we evaluate the sensitivity of the filtration step and also quantify the data size reduction after filtration. The mock data was sequenced from synthetic communities. It contains 64 archaea/bacteria members (Shakya et al., 2013). Of the 64 species in the dataset, 48 of them have 199 CRISPR annotations in CRISP Rdb with at least three spacers. The dataset contains about 105 million of 101 bp paired-end reads. The size of the fastq file is 34G. As the species, their genomes, and the dataset contains about 105 million of 101 bp paired-end reads. The size of the fastq file is 34G. As the species, their genomes, and the genomes available. However, we cannot exclude the possibility that the spacer unit in this dataset, which mainly consists of species with full sensitivity of CRISPR read identification. The increase is modest because two sequences of the repeats in one read for such CRISPRs with long repeat-spacer units. Using paired-end information, the pipeline failed to recruit any reads for four CRISPRs whose lengths of repeat-spacer unit are larger than 80 nt. CRISPRs. The performance of the two runs was very similar for many CRISPRs in the dataset. However, without utilizing paired-end read information, the pipeline failed to recruit any reads for four CRISPRs whose lengths of repeat-spacer unit are larger than 80 nt. As the read length of the dataset is 101 bp, it is impossible to see two copies of the repeats in one read for such CRISPRs with long repeat-spacer units. Using paired-end read information, the filtration step was able to recruit most of the reads for these CRISPRs. This showed that utilizing paired-end read information can increase the sensitivity of CRISPR read identification. The increase is modest because there are not many characterized CRISPRs with long repeat-spacer units in this dataset, which mainly consists of species with full genomes available. However, we cannot exclude the possibility that many more novel CRISPRs with longer repeat-spacer units exist in accumulating metagenomic data.

To further investigate the performance of using paired end information in filtration, we evaluated the sensitivity of CRISPR read identification for each CRISPR sequence. Specifically, for each CRISPR in the mock dataset, we compared the sensitivity of the filtration step with and without using paired end read information. We ran the filtration step twice. In the first run, we used both single read and paired-end information. In the second run, we did not use paired-end information. In Figure 5, x-axis shows the index of 199 CRISPRs in the dataset. The CRISPRs are sorted by the size of the repeat-spacer units. The figure shows that the read identification step can identify a large proportion of true positive reads for most CRISPRs. The performance of the two runs was very similar for most CRISPRs in the dataset. However, without utilizing paired-end read information, the pipeline failed to recruit any reads for four CRISPRs whose lengths of repeat-spacer unit are larger than 80 nt. As the read length of the dataset is 101 bp, it is impossible to see two copies of the repeats in one read for such CRISPRs with long repeat-spacer units. Using paired-end read information, the filtration step was able to recruit most of the reads for these CRISPRs. This showed that utilizing paired-end read information can increase the sensitivity of CRISPR read identification. The increase is modest because there are not many characterized CRISPRs with long repeat-spacer units in this dataset, which mainly consists of species with full genomes available. However, we cannot exclude the possibility that many more novel CRISPRs with longer repeat-spacer units exist in accumulating metagenomic data.

There are a few CRISPRs with read identification sensitivity below 0.80. We took a closer look at five sequences with the lowest sensitivity. The major reason behind the low sensitivity is that these CRISPRs contain DRs that are not similar to conserved ones. Some of them have four mismatches. Thus reads containing only non-conserved DRs will not be recruited. However, our tool does not need all sequenced reads to recover a CRISPR. Four of the five CRISPRs are fully assembled and reported by our tool.

For soil data and gut data, as we do not know all the ground truth species in the datasets, we cannot report the accuracy. Table 1 shows that the filtration can discard a significant amount of reads rapidly. For the three datasets, the sizes of the reduced datasets range from 0.57% to 2.08% of the raw datasets. As shown in the table, running time and memory usage of the read identification step is small with respect to the size of the raw datasets.

To further investigate the performance of using paired end information in filtration, we evaluated the sensitivity of CRISPR read identification for each CRISPR sequence. Specifically, for each CRISPR in the mock dataset, we compared the sensitivity of the filtration step with and without using paired end read information. We ran the filtration step twice. In the first run, we used both single read and paired-end information. In the second run, we did not use paired-end information. In Figure 5, x-axis shows the index of 199 CRISPRs in the dataset. The CRISPRs are sorted by the size of the repeat-spacer units. The figure shows that the read identification step can identify a large proportion of true positive reads for most CRISPRs. The performance of the two runs was very similar for most CRISPRs in the dataset. However, without utilizing paired-end read information, the pipeline failed to recruit any reads for four CRISPRs whose lengths of repeat-spacer unit are larger than 80 nt. As the read length of the dataset is 101 bp, it is impossible to see two copies of the repeats in one read for such CRISPRs with long repeat-spacer units. Using paired-end read information, the filtration step was able to recruit most of the reads for these CRISPRs. This showed that utilizing paired-end read information can increase the sensitivity of CRISPR read identification. The increase is modest because there are not many characterized CRISPRs with long repeat-spacer units in this dataset, which mainly consists of species with full genomes available. However, we cannot exclude the possibility that many more novel CRISPRs with longer repeat-spacer units exist in accumulating metagenomic data.

![Fig. 5. Sensitivity of the CRISPR reads identification step. CRISPRs are sorted along the x-axis by the size of the repeat-spacer unit. The blue line with squared marker shows the sensitivity of the run with paired-end information. The red line with triangle marker shows the sensitivity of the run without using paired-end information. The line with cycle marker shows the average read and paired-end information. In the second run, we did not use paired-end information. The major reason behind the low sensitivity is that these CRISPRs contain DRs that are not similar to conserved ones. Some of them have four mismatches. Thus reads containing only non-conserved DRs will not be recruited. However, our tool does not need all sequenced reads to recover a CRISPR. Four of the five CRISPRs are fully assembled and reported by our tool.](https://academic.oup.com/bioinformatics/article-abstract/32/17/i520/2450786)

3.2 The performance of read clustering
The goal of the read clustering step is to separate different CRISPRs as well as to reduce false positive reads recruited in the first step. With known reference genomes, we used the mock dataset to evaluate the performance of read clustering. The cluster step produced 1548 clusters on the filtered mock data. Many of them contain only FP reads and will be easily removed by later stages because the assembled contigs do not satisfy the properties of CRISPRs. Table 2 shows that after the clustering step, most of the reads sequenced from CRISPR regions (TP reads) were kept, while a large portion of FP reads were removed. There were still over 236k FP reads in the clusters. Most of them can be identified and removed in the assembly step or the validation step after assembly for the following reasons. First, some of these reads form random overlaps with true CRISPR reads, they can be removed in the graph cleaning stage using topology-based pruning. Second, for contigs that are generated by FP reads in clusters, the final validation step can remove these contigs because they would not meet the sequence properties of CRISPRs.

We then evaluated the ability of metaCRISPR to separate different CRISPRs. After running the read clustering step on the identified CRISPR reads in mock dataset, reads belonging to one CRISPRs may go to several clusters. In addition, one cluster may harbor multiple CRISPRs. Figure 6 quantifies the number of CRISPRs (blue line with squared marker) in each cluster and also the number of clusters for each CRISPR sequence (red line with triangle marker). Here we only included 175 clusters, each of which has more than 5 TP reads. According to Figure 6, most CRISPRs are separated to their own clusters. For the 199 CRISPRs, only 13 of them have more than one cluster. 29 out of 175 clusters have more than one CRISPRs. Of these 29 clusters, 28 of them contain CRISPRs from one species.

We analyzed the cluster with the most CRISPRs to understand why multiple CRISPRs are clustered to the same set. This cluster contains 15 CRISPRs from two species. The identifiers for these two species are NC_010483 and NC_009486, respectively. We generated the left and right flanking sequences of length 200 nt for each CRISPR in the two species, and conducted pairwise alignment of the sequences.

### Table 2. The performance of clustering on mock dataset

| Read type | Reads after filtration | Reads after clustering | Reduction |
|-----------|------------------------|------------------------|-----------|
| TP reads  | 232 399                | 232 211                | 0.08%     |
| FP reads  | 962 419                | 236 714                | 75.4%     |
| Total     | 1 194 816              | 468 925                |           |
flanking sequences. It is clear that a lot of flanking sequences have very long common substrings: for 30 flanking sequences from 15 CRISPRs, there are 100 pairs of flanking sequences whose overlap sizes are larger than 50. There are 56 pairs of flanking sequences whose overlap sizes are larger than 100. Because flanking sequences of these 15 CRISPRs share long substrings, reads from these CRISPRs are classified to the same cluster.

### 3.3 Performance of CRISPR assembly on the mock dataset

Next, we applied the assembly algorithm to the clusters of reads, and compared the performance of metaCRISPR with IDBA-ud/CRISPRFinder and Crass. The overlap threshold for creating the initial graph and the confident overlap threshold for the assembly were set to 30 and 50, respectively. For IDBA-ud/CRISPRFinder, we first ran IDBA-ud with default parameters on the full metagenomic dataset, the assembled contigs were uploaded to the CRISPRFinder website for CRISPR prediction. All CRISPRs predicted by CRISPRFinder were downloaded and used as the output of the pipeline.

For Crass, we ran it on the raw metagenomic dataset with default parameters. Crass only reports spacers and their arrangement. It requires users to manually check the output files, find out which segments should be assembled together, and give the information to users to manually check the output files, find out which segments should be assembled together, and give the information to

The summary of the performance comparison was shown in Table 3. The numbers of fully recovered CRISPRs for metaCRISPR and IDBA-ud/CRISPRFinder are very similar: metaCRISPR fully recovered 145 CRISPRs while IDBA-ud/CRISPRFinder fully recovered 147. IDBA-ud partially recovered 38 CRISPRs, while metaCRISPR partially recovered 36 CRISPRs. IDBA-ud only produced one chimeric CRISPR. metaCRISPR was only slightly worse than IDBA-ud: it produced three chimeric CRISPRs. Overall, the accuracy of metaCRISPR is comparable to the accuracy of IDBA-ud/CRISPRFinder pipeline on the raw metagenomic dataset. However, as for resource usage, our pipeline was much faster and used much less memory than IDBA-ud/CRISPRFinder. The mock sample is a medium-sized metagenomic dataset. To assemble it, IDBA-ud used about 34GB memory and took more than 18h. The resources required by IDBA-ud are not available for most of today’s desktop machines. metaCRISPR was much more resource efficient: it only took 1.56h and used 4.1 GB memory. This makes it more convenient for identifying CRISPRs on large scale metagenomic dataset on even a personal computer. Crass with cap3 generated 632 CRISPRs. This is much larger than the number of annotated CRISPRs (199). It only fully recovered 84 and partially recovered 103 of the annotated CRISPRs. The performance for Crass with velvet was the lowest among all the benchmarked pipelines.

Another aspect to evaluate the performance of the benchmarked tools is to check how many of the identified spacers overlap with spacers in the annotated CRISPRs in CRISPRdb. Table 3 shows that the numbers of spacers generated by metaCRISPR and IDBA-ud/CRISPRFinder are quite similar. For metaCRISPR, 82.6% of the identified spacers overlapped with annotations. For IDBA-ud/CRISPRFinder, the number is 83.4%. Crass generated similar number of spacers as IDBA-ud/CRISPRFinder. However, the number of spacers that are in the annotated CRISPRs were lower than both metaCRISPR and IDBA-ud/CRISPRFinder: only 79.3% of the identified spacers overlapped with annotations. Further work is needed to determine whether Crass detected false spacers.

### 3.4 Experimental result on the soil dataset

In order to test the performance of our tool in large and complex metagenomic datasets sequenced from real environmental niches, we applied metaCRISPR to a large soil metagenomic dataset. The
dataset contains 271 million of 150 nt paired-end reads and the fastq file of the dataset is about 117G. We tried to assemble the dataset using IDBA-ud on a high performance computing node with 120G memory but failed. The program was killed because it used more memory than requested. We ran Crass on the raw dataset. It was killed too after 24 h. Thus, here we reported the results of IDBA-ud/CRISPRFinder and Crass on the filtered dataset produced by our read identification step. For Crass, we ran it with cap3 because the first experiment showed that cap3 had better performance than velvet.

Unlike the first experiment, the genomes in the dataset are unknown. Instead of comparing with annotated CRISPRs, we used the following metrics: the number of CRISPRs identified, the number of spacers identified, max length of the identified CRISPRs, average length of the identified CRISPRs and resource usage. The result was shown in Table 4. metaCRISPR detected 17 more CRISPRs and 137 more spacers than IDBA-ud/CRISPRFinder on this dataset. The overlap between the spacers detected by metaCRISPR and IDBA-ud/CRISPRFinder was 655, which is about 68.7% of all the spacers detected by metaCRISPR. The maximum length of the CRISPRs output by metaCRISPR and IDBA-ud/CRISPRFinder was the same. On average, metaCRISPR output shorter CRISPRs than IDBA-ud/CRISPRFinder.

Crass with cap3 predicted 811 CRISPRs. The number was about eight times the number of CRISPRs predicted by IDBA-ud/CRISPRFinder. The average length of the predicted CRISPRs output by Crass was less than half of the average CRISPR length of the other tools. Crass detected 2479 spacers, which was much larger than the output of metaCRISPR and IDBA-ud/CRISPRFinder. We mapped the spacers predicted by Crass to the input reads data. About half of the spacers can only be mapped to one read, and 1741 of the spacers can only be mapped to less than three reads. The read counts for these spacers are too low to distinguish them between true spacers from lowly sequenced CRISPRs and false positives.

3.5 Experimental result on the human gut dataset

In the last experiment, we applied metaCRISPR to a human gut metagenomic dataset. The dataset was sequenced using Illumina from family members of a patient with diarrhea by our collaborator.

The dataset contains sequences from both bacteria and virus. We analyzed six samples. One was sequenced from the patient with illness, another sample (followup) was sequenced after the patient had recovered from the illness. The other four samples (control) were sequenced from other healthy family members. The numbers of reads in the samples range from 3 million to 7 million.

As this dataset contains sequences of both bacteria and virus, we evaluated the reliability of the identified spacers by comparing detected spacers against assembled viral genomes. In the cas-CRISPR system, the first stage in the immune response to an invading phage involves capturing a short part of the DNA of the phage, inserting the sequence to one end of a CRISPR locus, and forming a spacer. Thus if a spacer was newly acquired from an invading phage in the environment, the spacer sequence should match part of the genome of a phage in the environment. By searching the viral contigs assembled from the metagenomic dataset, it’s possible to validate whether the predicted spacers are reliable or not.

We first predicted CRISPRs using metaCRISPR and IDBA-ud/CRISPRFinder on all six samples. The number of CRISPRs and spacers predicted were shown in Table 5. In most samples, metaCRISPR predicted more CRISPRs and more spacers. For all the assembled contigs by IDBA-ud, we used Metavir (Roux et al., 2014) to annotate viral metagenomic sequences. Metavir is able to annotate each sequence as virome or non-virome based on similarity score with known viromes. The number of virome-like contigs for each sample was shown in Table 5. We used Bowtie to map the predicted spacers to the virome-like contigs in each sample. The numbers of hits are shown in the ‘hits’ column of Table 5. For four out of six samples, there were some proportions of spacers mapped to the virome-like contigs assembled from the same sample. For all the four samples, metaCRISPR had more hits than IDBA-ud/CRISPRFinder. In particular for the sample sequenced from the patient, and the sample sequenced from a healthy family member (healthy4), IDBA-ud/CRISPRFinder had no hits, while metaCRISPR had seven and six hits, respectively. The experimental results provide additional evidence that spacers identified by metaCRISPR are reliable.

### Table 4. The performance of the benchmarked tools on the soil dataset

| Tool                  | Predictions | Spacers | Max Length | Average Length | Time (h) | Memory (G) |
|-----------------------|-------------|---------|------------|----------------|----------|------------|
| metaCRISPR            | 120         | 954     | 5030       | 531            | 1.12     | 3.3        |
| IDBA-ud/CRISPRFinder  | 103         | 817     | 5030       | 572            | 1.2      | 3.3        |
| Crass (cap3)          | 811         | 2479    | 4847       | 242            | 1.07     | 0.91       |

*IDBA-ud was run on the raw dataset. The process was killed because it used more memory than requested.

*Crass was run on the raw dataset and was killed after 24 h.

*IDBA-ud and Crass were run on the filtered dataset.

### Table 5. The performance of the benchmarked tools on the human gut dataset

| Sample    | Virome contigs | metaCRISPR CRISPRs | Spacers | Hits | IDBA-ud/CRISPRFinder CRISPRs | Spacers | Hit |
|-----------|----------------|---------------------|---------|------|-------------------------------|---------|-----|
| Patient   | 6535           | 18                  | 101     | 7    | 4                            | 4       | 37  |
| Follow-up | 4680           | 15                  | 131     | 0    | 14                           | 14      | 133 |
| Healthy1  | 9416           | 35                  | 364     | 27   | 39                           | 39      | 288 |
| Healthy2  | 7620           | 35                  | 343     | 3    | 28                           | 28      | 262 |
| Healthy3  | 36             | 30                  | 293     | 0    | 20                           | 20      | 255 |
| Healthy4  | 2738           | 20                  | 138     | 6    | 21                           | 21      | 149 |
We also searched predicted spacers against the public virus database from NCBI. Less matches can be identified than using Metavir. This is not surprising as most of the reads sequenced from virus community cannot be mapped to the public virus genome database. Metavir is likely to identify some ‘new’ virus fragments that have not been included in the public databases. But even with Metavir, there are still limited number of matches between predicted spacers and virome-like contigs. Although the spacers were originally obtained from invading viruses, they go through various changes such as mutations or loss of the whole spacers. In addition, the viral genomes mutate as well. Thus, most spacers cannot find exact or near-exact matches in viral genomes. It was reported that only the most recently acquired spacers match coexisting viruses (Andersson and Banfield, 2008).

Acknowledgements
We would like to thank Jiarong Guo and Dr. James Tiedje for providing us the soil data. We thank Dr. Pallavi Singh and Dr. Shannon Manning for providing us the human gut metagenomic data.

References
Andersson, A.F. and Banfield, J.F. (2008) Virus population dynamics and acquired virus resistance in natural microbial communities. Science, 320, 1047–1050.
Barrangou, R. et al. (2007) CRISPR provides acquired resistance against viruses in prokaryotes. Science, 315, 1709–1712.
Ben-Bassat, L. and Chor, B. (2015) CRISPR detection from short reads using partial overlap graphs. Lect. Notes Comput. Sci., 9029, 16–27.
Bland, C. et al. (2007) Crispr recognition tool (crt): a tool for automatic detection of clustered regularly interspaced palindromic repeats. BMC Bioinformatics, 8, 209.
Cong, L. et al. (2013) Multiplex genome engineering using CRISPR/Cas systems. Science, 339, 819–823.
Cui, Y. et al. (2008) Insight into microevolution of Yersinia pestis by clustered regularly interspaced short palindromic repeats. PLoS One, 3, e2652.
DiCarlo, J.E. et al. (2013) Genome engineering in Saccharomyces cerevisiae using CRISPR-Cas systems. Nucleic Acids Res., 41, 4336–4343.
Edgar, R.C. (2007) PIler-CR: fast and accurate identification of CRISPR repeats. BMC Bioinformatics, 8, 18.
Friedland, A.E. et al. (2013) Heritable genome editing in C. elegans via a CRISPR-Cas9 system. Nat. Methods, 10, 741–743.
Grissa, I. et al. (2007) The CRISPRdb database and tools to display CRISPRs and to generate dictionaries of spacers and repeats. BMC Bioinformatics, 8, 172.
Hafner, D.H. et al. (2005) A guild of 45 CRISPR-associated (Cas) protein families and multiple CRISPR/Cas subtypes exist in prokaryotic genomes. PLoS Comput. Biol., 1, e60.
Hale, C.R. et al. (2012) Essential features and rational design of CRISPR RNAs that function with the Cas RAMP module complex to cleave RNAs. Mol. Cell, 45, 292–302.
Horvath, P. and Barrangou, R. (2010) CRISPR/Cas, the immune system of bacteria and archaea. Science, 327, 167–170.
Horvath, P. et al. (2008) Diversity, activity, and evolution of CRISPR loci in Streptococcus thermophilus. J. Bacteriol., 190, 1401–1412.
Huang, X. and Madan, A. (1999) Cap3: a DNA sequence assembly program. Genome Res., 9, 868–877.
Hwang, W.Y. et al. (2013) Efficient genome editing in zebrafish using a CRISPR-Cas system. Nat. Biotechnol., 31, 227–229.
Jansen, R. et al. (2002) Identification of a novel family of sequence repeats among prokaryotes. Omics, 6, 23–33.
Jiang, W. et al. (2013) Demonstration of CRISPR/Cas9 sgRNA-mediated targeted gene modification in Arabidopsis, tobacco, sorghum and rice. Nucleic Acids Res., 41, e188.
Karginov, F.V. and Hannon, G.J. (2010) The CRISPR system: small RNA-guided defense in bacteria and archaea. Mol. Cell, 37, 7–19.
Kunin, V. et al. (2007) Evolutionary conservation of sequence and secondary structures in CRISPR repeats. Genome Biol., 8, R61.
Larson, M.H. et al. (2013) CRISPR interference (CRISPRs) for sequence-specific control of gene expression. Nat. Protoc., 8, 2180–2196.
Marraffini, L.A. and Sontheimer, E.J. (2008) CRISPR interference limits horizontal gene transfer in staphylococci by targeting DNA. Science, 322, 1843–1845.
Martin, J.A. and Wang, Z. (2011) Next-generation transcriptome assembly. Nat. Rev. Genet., 12, 671–682.
Pennisi, E. (2013) The CRISPR craze. Science, 341, 833–836.
Pougach, K. et al. (2010) Transcription, processing and function of CRISPR cassettes in Escherichia coli. Mol. Microbiol., 77, 1367–1379.
Pride, D.T. et al. (2011) Analysis of streptococcal CRISPRs from human saliva reveals substantial sequence diversity within and between subjects over time. Genome Res., 21, 126–136.
Rho, M. et al. (2012) Diverse CRISPRs evolving in human microbiomes. PLoS Genet., 8, e1002441.
Roux, S. et al. (2014) Metavir 2: new tools for viral metagenome comparison and assembled virome analysis. BMC Bioinformatics, 15, 76.
Shakya, M. et al. (2013) Comparative metagenomic and rRNA microbial diversity characterization using archaeal and bacterial synthetic communities. Environ. Microbiol., 15, 1882–1899.
Skenerton, C.T. et al. (2013) Crass: identification and reconstruction of CRISPR from unassembled metagenomic data. Nucleic Acids Res., 41, e105.
Sorek, R. et al. (2008) CRISPR—a widespread system that provides acquired resistance against phages in bacteria and archaea. Nat. Rev. Microbiol., 6, 181–186.
Sorokin, V.A. et al. (2010) Evolutionary dynamics of clustered irregularly interspaced short palindromic repeat systems in the ocean metagenome. Appl. Environ. Microbiol., 76, 2136–2144.
Stern, A. et al. (2012) CRISPR targeting reveals a reservoir of common phages associated with the human gut microbiome. Genome Res., 22, 1985–1994.
Wang, H. et al. (2013) One-step generation of mice carrying mutations in multiple genes by CRISPR/Cas-mediated genome engineering. Cell, 153, 910–918.
Wei, Y. et al. (2015) Sequences spanning the leader-repeat junction mediate CRISPR adaption to phage in streptococcus thermophilus. Nucleic Acids Res., 43, 1749–1758.
Yuan, C. et al. (2015) Reconstructing 16s rna genes in metagenomic data. Bioinformatics, 31, i35–i43.
Zerbino, D.R. and Birney, E. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res., 18, 821–829.