Improving ancient DNA genome assembly

Alexander Seitz[1] and Kay Nieselt[1]

[1]**Center for Bioinformatics (ZBIT), Integrative Transcriptomics,
Eberhard-Karls-Universität Tübingen**

ABSTRACT

Most reconstruction methods for genomes of ancient origin that are used today require a closely related reference. In order to identify genomic rearrangements or the deletion of whole genes, *de novo* assembly has to be used. However, because of inherent problems with ancient DNA, its *de novo* assembly is highly complicated. In order to tackle the diversity in the length of the input reads, we propose a two-layer approach, where multiple assemblies are generated in the first layer, which are then combined in the second layer. We used this two-layer assembly to generate assemblies for an ancient sample and compared the results to current *de novo* assembly approaches. We are able to improve the assembly with respect to the length of the contigs and can resolve more repetitive regions.

Keywords: ancient DNA, de novo assembly, genome reconstruction

INTRODUCTION

The introduction of next generation sequencing (NGS) made large scale sequencing projects feasible (Bentley et al., 2008). Their high throughput allows fast and cheap sequencing of arbitrary genomic material. It revolutionized modern sequencing projects and made the study of ancient genomes possible (Der Sarkissian et al., 2015). However, the resulting short reads pose several challenges for the reconstruction of the desired genome when compared to the longer Sanger reads (Li et al., 2010). For modern DNA samples, the problem of having only short reads can be mitigated by the sheer volume of sequenced bases and usage of long fragments with paired-end sequencing. The insert size is used to determine the distance between the forward and the reverse read, which are sequenced from both ends of the fragments. These distances can be important for *de novo* assembly as they are used for repeat resolution and scaffolding. They provide the same information as using one long Sanger read as the bases in between can be derived from other reads. However, samples from ancient DNA (aDNA) mostly contain only very short fragments between 44 and 172 bp (Sawyer et al., 2012). Paired-end sequencing of these short fragments therefore often results in overlapping forward and reverse reads (thus actually negative inner mate pair distances). Additionally, post-mortem damage of aDNA, most importantly the deamination of cytosine to uracil, can result in erroneous base incorporation (Rasmussen et al., 2010). Using reference based approaches, these errors can be detected, as they always occur at the end of the fragments. This is not possible using *de novo* assembly approaches and these errors can lead to mistakes in the assembly. Deeper sequencing does not yield better results as the amount of endogenous DNA contained in aDNA samples is often very low (Sawyer et al., 2012).

In order to achieve a higher content of endogenous DNA, samples are often subject to enrichment using capture methods (Avila-Arcos et al., 2011). The principle of these capture methods relies on selection by hybridization (Maricic et al., 2010). Regions of interest are fixed to probes prior to sequencing. These probes can be immobilized on glass slides, called array capture (Hodges et al., 2007), or recovered by affinity using magnetic beads, referred to as in-solution capture (Gnirke et al., 2009). Using these capture methods, only DNA fragments that can bind to the probes are used for amplification, which increases the amount of the desired DNA. However, as these methods only amplify sequences that are contained in the probes, regions that were present in ancient samples and lost over time cannot be amplified and thus cannot be identified. Nevertheless, most of the current aDNA projects use these capture methods.

Currently, there are two ways to reconstruct a genome from sequencing data. If there is a known, closely related genome, it can be used as a reference. Mapping programs like BWA (Li and Durbin, 2009) can then be used to align the reads against the reference genome. Single nucleotide variations (SNVs) or short indels between the DNA sequence of the sample and reference can be identified after all reads are aligned.

Because of the inherent characteristics of aDNA, specialized mapping pipelines for the reconstruction of aDNA genomes, such as EAGER (Peltzer et al., 2016) and PALEOMIX (Schubert et al., 2014), have recently been published. The mapping against a reference genome allows researchers to easily eliminate non-endogenous DNA and identify erroneous base incorporations. These errors can be identified after the mapping and used to verify that the sequenced fragments stem from ancient specimen.

The reference-based mapping approaches cannot detect large insertions or other genomic architectural rearrangements. In addition, if the ancient species contained regions that are no longer present in the modern reference, these cannot be identified via mapping against modern reference genomes. In these cases a *de novo* assembly of the genome should be attempted. This is also true for modern samples, if no closely related reference is available. If the ancient sample was sequenced after amplification through capture arrays, genomic regions that are not contained on the probes also can't be identified. Using shotgun sequencing, sequences that stem from species that migrated into the sample post-mortem are often more abundant (Knapp and Hofreiter, 2010). However, if shotgun data is available an effort for assembly can be made to identify longer deletions or genomic rearrangements. The introduction of NGS has lead to new assembly programs that can handle short reads such as SOAPdenovo2 (Luo et al., 2012), SPADES (Bankevich et al., 2012) and many more.

The assembly of modern NGS data is still a hard problem (Chao et al., 2015) and methods to improve them are constantly developed. Among these is ALLPATHS-LG (Gnerre et al., 2011), arguably the winner of the so-called Assemblathon (Earl et al., 2011). ALLPATHS-LG uses the information provided by long fragments from paired-end and mate-pair sequencing to improve the assembly, and has therefore been shown to be one of the best assembly programs that are available today (Utturkar et al., 2014). However, because of the short fragments contained in aDNA samples, this approach is not feasible for aDNA samples and other methods have to be employed.

De Bruijn graph assemblers highly rely on the length of the k-mer to generate the graph (Li et al., 2012). The choice of an optimal value is already a hard problem for modern sequencing projects (Durai and Schulz, 2016).

Because of the short fragments of aDNA samples, the sequencing adapter is often partially or fully sequenced. After the adapter is removed, the length of the resulting read is then equal to the length of the fragment. Furthermore, overlapping forward and reverse reads can be merged to generate longer reads, which is usually done in aDNA studies to improve the sequence quality (Peltzer et al., 2016). Thus the length distribution of reads from aDNA samples is often very skewed. This implies that the choice of one single fixed k for the k-mer in *de Bruijn* graph-based assembly approaches is not ideal in aDNA studies. Long k-mers miss all reads that are shorter than the value of k and shorter k-mers cannot resolve repetitive regions.

We have developed a two-layer assembly approach where in the first layer, the contigs are assembled from short reads using a *de Bruijn* graph approach with multiple k-mers. These contigs are then used in the second layer in order to combine overlapping contigs contained in the different assemblies resulting from the first layer. This is done using an overlap-based approach.

Outline   This article is organized as follows. The next section contains the methods we used to improve the *de novo* assembly for aDNA samples. In short, we used multiple assemblies with different k-mers and then merge these assemblies into longer contigs. In the results section, we used our two-layer assembly to improve the assembly of the sample Jorgen625 published by Schuenemann et al. (2013). Finally, we conclude our findings and give an outlook.

METHODS

The general structure of our two-layer approach is as follows: In the first layer, the raw `fastq` files are preprocessed, followed by a *de Bruijn* graph-based assembly using multiple k-mer sizes to generate several different, yet similar assemblies. All produced contigs are quality filtered before they are combined and used in the second layer. There, an overlap-based approach is used to identify contigs in the different assemblies that represent the same genomic region. These can be merged into longer contigs. Afterwards small contigs are removed from the result. The rest of this section explains these steps in more detail.

We used the tool Clip & Merge (Peltzer et al., 2016) to remove the sequencing adapters. It was also used to quality trim all bases in reads below a minimum phred score of 20. This threshold was left at this default value as the low-quality ends of the reads are merged and thus the base call is confirmed by two reads. The value was not changed for the unmerged reads in order to be able to compare the experiments. In order to evaluate how different preprocessing affects the assembly, the reads were
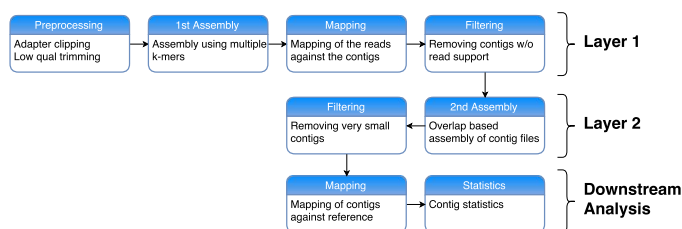
Figure 1. Workflow of our two-layer assembly approach. First the reads are preprocessed by removing sequenced adapters and clipping low-quality bases. After that, multiple *de novo* assemblies are generated using a *de Bruijn* graph approach with multiple values for *k*. The reads are then mapped back against each of these resulting contigs and the contigs with no read support are filtered out. In Layer 2, these filtered contigs are then combined and assembled again using an Overlap-Layout-Consensus approach. Very short contigs are removed. The resulting contigs are mapped against a reference genome and contig statistics are calculated in order to assess the quality of the assembly.

treated using three different methods: First, the reads were only adapter clipped and trimmed. Reads that no longer have a partner were removed. These reads were then used in a paired-end assembly. Second, after the reads were adapter clipped and quality trimmed, all resulting forward and reverse reads were combined into one file, each read given a unique identifier so that they could be used in a single-end assembly. Third, after the adapter clipping the forward and reverse reads were merged into longer reads whenever possible. For the merging of the reads, we used the standard parameters of Clip & Merge defining a minimum overlap length of 10 bp with a maximum mismatch rate of 5%. The resulting reads were then quality trimmed as described above. Unique identifiers were assigned to forward and reverse reads that could not be merged and added to the resulting `fastq` file. These reads were then used in a single-end assembly. In all three sets, resulting reads that were shorter than 25 bp were removed before the assembly.

After the preprocessing, the resulting reads are of different lengths. The reason for this are the different fragment lengths contained in the sample. This is why we propose assembly of aDNA using a two-layer approach. In the first layer, we use a *k*-mer based assembly program like SOAPdenovo2 (Luo et al., 2012), MEGAHIT (Li et al., 2014), or any other assembly program for which different values for *k* can be chosen.

De Bruijn based programs first generate all possible *k*-mers based on the input reads. Matching *k*-mers are used to generate the *de Bruijn* graph. This can lead to random overlaps of *k*-mers contained in different reads and therefore to infeasible contigs. To filter out these contigs, the reads are mapped back against the resulting contigs. This can be done by using modern mapping programs like BWA-MEM (Li, 2013). Contigs that are not supported by any read are removed before the next step.

To combine the results of the different assemblies, each contig is given a unique identifier before they are combined into one file. This file is the input of the second layer assembly. Here, the assembly is based on string overlaps instead of *k*-mers. An assembly program that uses this approach is the String Graph Assembler (SGA) (Simpson and Durbin, 2012). It efficiently calculates all overlaps of the input using suffix arrays (Manber and Myers, 1993). These overlaps are then used to generate an overlap graph and the final contigs are generated based on this graph. We used this method to merge the contigs from the different assemblies based on their overlap.

As SGA uses string-based overlaps and modern sequencing techniques are not error-free, it provides steps to correct for these errors. There is a preprocessing step that removes all bases that are not A,G,C or T. There is also a correction step that performs a *k*-mer based error correction and a filtering step that removes input reads with a low *k*-mer frequency. Because the input for SGA are already pre-assembled contigs, these errors are already averaged out and these steps are not used for the assembly of the second layer. However, the assemblies with the different *k*-mers produce similar contigs, which is why the duplicate removal step of SGA is performed. SGA can also use the Ferragina Manzini (FM) index (Ferragina and Manzini, 2000) to merge unambiguously overlapping sequences, which is used to further remove duplicate information. Afterwards the overlap graph is calculated and the new contigs are assembled. All these steps are performed using the standard parameters provided by SGA. Afterwards, contigs shorter than 1 000 bp are removed from the final assembly. In order to evaluate our two-layer assembly method, the resulting contigs are then aligned

3/10

with the reference genome of interest. We use again BWA-MEM for this step. Finally various statistics for the assembly are computed.

An overview of this methodology can be seen in Figure 1.

RESULTS

To evaluate our two-layer assembly, we applied it to a published ancient sample containing DNA from *Mycobacterium leprae*. We used the sample Jorgen625 published by Schuenemann et al. (2013). The bones from which the DNA was extracted, are approximately 700 years old. Two different sequencing libraries are available for this sample. In order to get an overview of the two libraries, we used the EAGER pipeline (Peltzer et al., 2016) to map the two libraries against the reference genome of *Mycobacterium leprae TN*. One of the two libraries contained relatively long fragments with a mean fragment length of 173.5 bp and achieved an average coverage on the reference genome of 102.6X. The other library was sequenced on an Illumina MiSeq with a read length of 151 bp. It was produced from shorter fragments with a mean fragment length of 88.1 bp and a mean coverage of 49.3X. With its shorter fragments and lower achieved coverage, the second library better reflects typical sequencing libraries generated from aDNA samples (Sawyer et al., 2012), so we focused our experiments on this library.

Figure 2. Read length distribution for the different preprocessed `fastq` files. Blue: merged reads, red: single reads.

The distribution of the different read lengths after the different preprocessing steps were performed is shown in Figure 2. There are many reads that were clipped, trimmed or merged and thus not of equal length.

Each of the three input read files (generated from the three different preprocessing methods) were then subject to our two-layer assembly approach. We used both SOAPdenovo2 (Version 2.04) and MEGAHIT (v1.0.4-beta-3-g027c6b6) in the first layer of the assembly. In order to cover a broad range of k-mers representing both short and long reads contained in the input, we used ten different k-mer sizes $(37, 47, 57, \ldots, 127)$. After removing contigs with no read support, the contigs were then reassembled with SGA. To identify contigs that belong to the genome of *Mycobacterium leprae*, the results were mapped against the reference sequence of *Mycobacterium leprae TN*. Contigs that could be mapped against the reference were extracted and used to compare the assemblies generated in the different layers.

Table 1 shows statistical results of the contigs that could be mapped against the reference genome of *Mycobacterium leprae TN*. The results that were generated in the second layer are shown as well as the assembly that generated the longest contig in the first layer using the respective assembly program. Additionally, results from SGA directly on the `fastq` files as well as results from programs that can use multiple k-mers in their assembly are shown. It can be seen that when using SOAPdenovo2 in the first layer, the longest contig, the N50 and the mean contig length could be improved by using SGA to merge the different assemblies in the second layer. Here, the overall best assembly was derived with the preprocessing method using the combined trimmed and clipped reads for a single-end assembly in the first layer. SOAPdenovo2 can also generate its graph using multiple k-mers. The result of this method is better than using only one k-mer but not as good as our two-layer approach. Using MEGAHIT, the merging in the second layer with SGA also improved the assemblies generated in the first layer. MEGAHIT also provides the possibility to generate an assembly using multiple k-mers. As with SOAPdenovo2, they improve the assembly compared to using only one k-mer but the result is worse than out two-layer methodology. Another assembly program that can use multiple k-mers to generate a result is the "interactive *de Bruijn* graph de novo assembler" (IDBA) (Peng et al., 2010). Its results are very good but not as good as the second layer assembly with SOAPdenovo2 in the first layer.

The length distribution of the resulting contigs is shown in Figure 3. After the second layer assembly, the number of contigs at the upper end of the length distribution has increased, compared to the first layer. With MEGAHIT, this is also true, even though it is not as pronounced as in the assembly using SOAPdenovo2. Using MEGAHIT, the total number of contigs that could be mapped
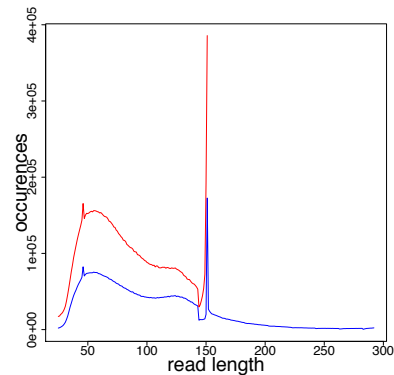
4/10

Table 1. Results using our two-layer assembly with SOAPdenovo2 and MEGAHIT as primary assemblies compared with the standard assemblies of SGA, SOAPdenovo2, MEGAHIT and IDBA on the short fragment library. The results show only values for contigs that could be mapped against the genome of *Mycobacterium leprae*. Here only the best assemblies (based on the longest mapped contig) for the different preprocessing methods and *k*-mers are shown. "SOAP" alone represents the results using the parameter (-m) resulting in an assembly using multiple different *k*-mers for the generation of their underlying graph structure. "MEGAHIT" and "IDBA" alone also represent an assembly using multiple internal *k*-mers. "SOAP K57" and "MEGAHIT K77" represent the best assemblies in the first layer of our pipeline using the respective *k*-mers of 57 and 77. "SOAP SGA" and "MEGAHIT SGA" show the results of the second layer using SOAPdenovo2 and/or MEGAHIT in the first layer. The column "preprocessing" describes the preprocessing method that was used to generate the result. Values in bold represent the best value that could be achieved. All other statistical values can be found in the supplementary material.

	name	prepro-cessing	# contigs	N50	mean con-tig length	longest contig	# gaps
Layer 1	SOAP	single	249	21909	13210.3	99866	103
	MEGAHIT	merged	175	28410	16777.5	91499	106
	IDBA	paired	164	35419	20152.7	118220	118
	SGA	single	1157	2199	1997.3	8640	952
	SOAP K57	single	215	24962	14918.6	72345	120
	MEGAHIT K77	merged	253	21863	12765.4	87880	108
Lvl 2	SOAP SGA	single	**133**	**42136**	**25225.0**	**135656**	88
	MEGAHIT SGA	merged	668	19758	12245.3	109259	**80**

against the reference genome after the second layer assembly with SGA is significantly higher than in the individual assemblies of the first layer. There are several more short contigs, whereas using SOAPdenovo2 in the first layer leads to fewer shorter contigs and more longer contigs after the second layer. Using SGA directly on the preprocessed `fastq` files did not result in good assembly results.

Since one normally is interested in one genome of interest (here the genome of the leprosy causing bacterium), we computed the genome coverage after mapping all contigs of length at least 1000 bases against *Mycobacterium leprae TN*. We used Qualimap2 (Okonechnikov et al., 2015) for the analysis of the mapping. The percentage of the genome that could be covered using only contigs longer than $1\,000, 1\,500, \dots, 10\,000$ bp is shown in Figure 4. It can be seen that the percentage of the

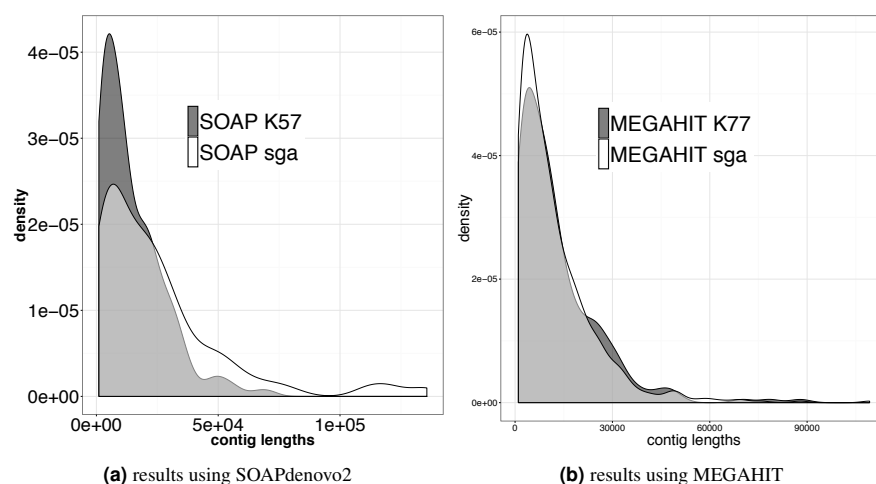**(a)** results using SOAPdenovo2

(b) results using MEGAHIT

Figure 3. Distribution of the length of the contigs generated by the different assemblies. The results generated by the second layer assembly with SGA is shown in white. The results of one first layer assembly is shown in dark grey. The light grey part represents values that belong to both methods. In 3a, the results using SOAPdenovo2 in the first layer are described. The results using MEGAHIT in this layer are shown in 3b.
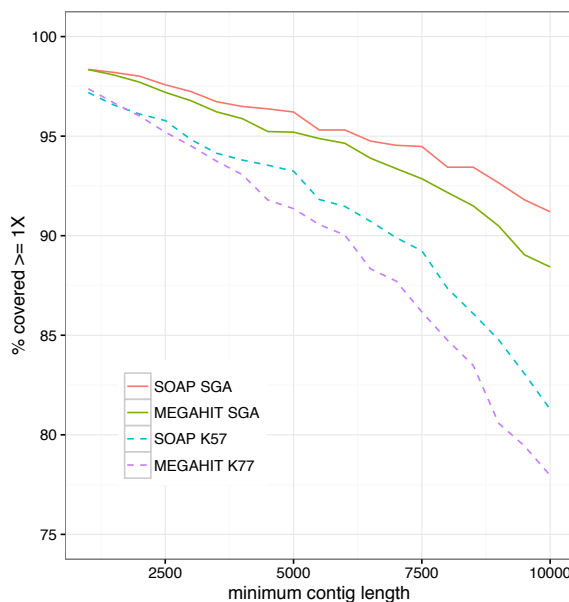
Figure 4. The percentage that could be covered with contigs longer than the minimum contig length

genome that could be covered using different cutoffs for the minimum length of the contigs is always higher after the second layer assembly using SGA than using only the results generated in the first layer assemblies. This becomes more and more pronounced with increasing filter threshold for the minimum contig lengths. When using only contigs longer than 1 000 bp, the results are almost the same. Using only contigs longer than 10 000 bp, around 90% of the genome can be covered using the second layer assembly with SGA, whereas at most 80% of the genome is covered by contigs from assemblies generated in the first layer.

The percentage of the genome that was covered at least twice is around 1% for the assemblies generated in the first layer with SOAPdenovo2 and MEGAHIT. This value has increased after the second layer assembly where the contigs were assembled again with SGA, showing that not all overlapping contigs could be identified and merged by SGA.

In order to be able to merge more contigs, we performed a new experiment that also uses the internal error correction of SGA that were described in the previous section. The resulting assembly contained contigs of length $\geq 400,000$ bp that could be mapped against the reference genome. However, when analyzing these contigs, only subsequences of at most 500 bp actually mapped to the genome. The beginning and the end of these contigs were soft-clipped by BWA-MEM and did not map anywhere else on the reference genome. When analyzing the contigs from the assemblies generated without this internal error correction of SGA, the whole contig (with some small insertions and deletions) could be mapped against the reference genome.

The mapping of the contigs generated by the first layer assemblies of SOAPdenovo2 and MEGAHIT against the reference genome resulted in approximately 115 gaps. This value is re-duced to around 84 gaps for the contigs generated by the second layer assembly with SGA (see Table 1). These gaps, together with annotated repeat regions of *Mycobacterium leprae*, are shown in Figure 5. It can be seen that the gaps in the mapping of the contigs mainly coincide with annotated repeat regions in the reference genome, as already shown by Schuenemann et al. (2013). It can also be seen that there are repetitive regions that can be resolved after assembling the different contigs in the second layer with SGA.

Up until now we showed that we were able to generate long, high quality contigs that can be mapped against the reference of *Mycobacterium leprae TN*. In order to show that the assembled contigs actually belong to the species of *Mycobacterium leprae* and not to other *Mycobacteria*, we took the ten longest contigs from each assembly and used BLASTN (Altschul et al., 1990) available on the NCBI webserver to align the contigs with all the genomes available from the genus *Mycobacterium*. The hits that generated the highest score for all of these contigs always belonged to a strain of *Mycobacterium leprae*.

While previous analyses confirmed the specificity of mapped contigs, there were several long
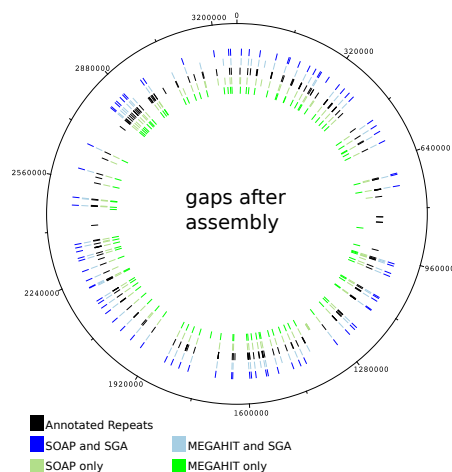
Figure 5. Gaps in the mapping of the contigs against the reference genome of *Mycobacterium leprae TN* together with annotated repeat regions in the reference genome. The outer ring represents the gaps that occur after the mapping of the contigs that were generated by the second layer assembly with SGA after a first layer assembly with SOAPdenovo2. The second outer ring shows the same but for a first layer assembly using MEGAHIT. The middle ring represents the annotated repeat regions of the reference genome. The second inner ring represents the gaps after using only SOAPdenovo2 with a *k*-mer size of 57 and the inner most ring represents the gaps after using only MEGAHIT with a *k*-mer size of 77.

contigs that could not be mapped. This is not surprising, because DNA from ancient bones is often mixed with other DNA and thus a metagenomic sample. For this experiment, the longest contig that could not be mapped against the reference genome of *Mycobacterium leprae TN* was used. This contig was aligned against the whole nr/nt database with BLASTN. The best hits achieved only a query coverage of approximately 13%. These regions on the query are not consecutive and map to different genes. The most promising gene that can be identified is the heat shock protein 70, which is a highly conserved gene among several bacteria (Bukau and Horwich, 1998). The same is true for the very long contigs generated using the correction steps of SGA or the iterative graph construction approach of MEGAHIT. There is not one species in the database where more than 15% of these queries could be aligned to.

In order to see how this two-layer assembly handles sequencing libraries of lower mean coverage of the desired species, we performed several experiments of different samples that showed a mean coverage between 2X and 7X after being mapped with the EAGER pipeline against the respective reference gnome. Here we could not achieve any meaningful results.

Furthermore we evaluated the scalability of our pipeline through subsampling. We used the library from the Jorgen625 sample with the longer fragments as it contained more than twice as many reads ($2 \times 15,101,591$ instead of $2 \times 6,751,711$ reads). We evaluated the whole pipeline using 1, 2, 5, 10 and all 15.1 million reads. The calculations were performed on a server with 500GB memory and 32 CPUs of type Intel® XEON® E5-416 v2 with 2.30 GHz using four threads wherever parallelization was possible. The results shown in Figure 6 show that the runtime scales linearly with the number of input reads. The time it would take to assemble a human genome using our two-layer approach can be estimated using this linear model. The ancient human LBK/Stuttgart sample published by Lazaridis et al. (2014) was sequenced using eight lanes, each containing between 200 and 230 million reads. The assembly of one such lane would take approximately one week and the assembly of all 1.74 billion reads almost two months.

DISCUSSION AND CONCLUSIONS

With ancient genome assembly one faces a number of challenges. The underlying dataset stems from a metagenomic sample with short fragments. When performing a paired-end sequencing experiment, this results in mostly overlapping forward and reverse reads. Because of the highly different read lengths after the necessary preprocessing steps, including adapter removal and quality trimming, typical *de Bruijn* approaches using a fixed *k*-mer size cannot sufficiently assemble the sample. On
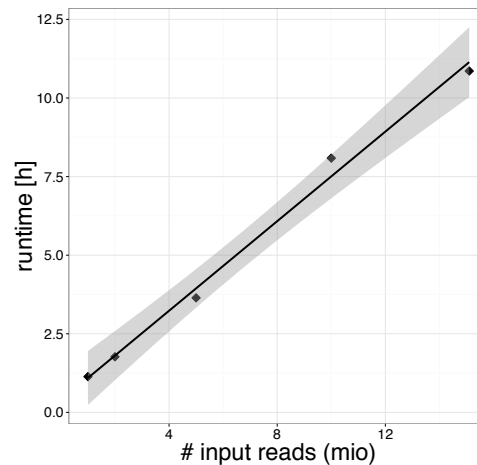
7/10

Figure 6. Runtime scaling of the two-layer assembly approach. The black dots show the runtime in minutes using 1, 2, 5, 10 and all 15.1 million input reads. The black line shows the fitted linear regression and the grey area represents the 95% confidence region.

the other hand overlap-based approaches alone are also inferior. Our two-layer approach combining various assemblies using different k-mer sizes followed by a second assembly based on string overlaps is able to fuse the contigs generated in the first layer into longer contigs and reduce the redundancy. Additionally, we could show that longer, high quality contigs are generated after the second layer assembly. In particular, at least for our example of a *Mycobacterium leprae* genome, these longer contigs are able to close more gaps, mainly spanning repetitive regions. The different values for k that are used in the first layer assembly lead to similar contigs that can be combined in the second layer assembly. The percentage of the genome that is covered more than once is increased after the second layer assembly. This proves that SGA is not able to identify and merge all overlapping contigs. One reason for this could be the underlying metagenomic sample. Multiple species in the sample share similar but not identical sequences. As SGA is not designed to assemble metagenomic samples, these differences cannot be distinguished from different sequences of the same genome containing small errors. One possibility to solve this could be to optimize the parameters that SGA provides, as the current parameters for SGA cannot merge all relevant contigs. This probably has to be adapted for each sample. However, we showed that when using the steps to account for sequencing errors, the resulting contigs became worse, when considering the specificity of the contigs (of the organism of interest). We believe that it could be a problem of multiple *Mycobacteria* in the sample that share similar sequences which are then combined to sequences that are built-up out of fragments of different species in the sample. The contigs that are generated without these steps are of high quality and map almost perfectly against the reference sequence that is known to be highly similar to the desired genome (Mendum et al., 2014). When assembling metagenomic and especially aDNA samples, the results always have to be regarded critically in order to avoid mistakes. Another possibility could be sequencing errors in the sample, leading to distinct contigs using different k-mers. However, these errors should be averaged out by the different assemblies (Schatz et al., 2010). Erroneous base incorporations can be ruled out as the sample was treated with *Uracil-DNA Glycosylase* (UDG), removing these errors.

An important step is the preprocessing of the raw reads. We compared the performance using all reads as single reads, as paired reads or as merged reads. However, at least from our study, we can conclude that the results highly depends on the first layer assembler and probably also on the dataset itself. What is interesting is the fact that SOAPdenovo2 produces better results when using all input reads in a single-end assembly than in a paired-end assembly. One possible explanation is that the information between the pairs does not contain additional information as almost all paired-end reads overlap and can be merged. It is possible that the program then disregards some overlaps in order to fulfill the paired-end condition. Overlaps that were disregarded this way could be used in the single-end assembly leading to a better assembly. Additionally, reads that did not have a partner were removed before the paired-end assembly. These reads are of course available in the single-end assembly. It could be that they contained some relevant information.

The mapping of the assembled contigs against the reference show that in our case,all gaps align

with annotated repeat regions. Using our two-layer assembly approach, more of these regions could be resolved, but many still remain. In sequencing projects of modern DNA, repetitive regions are resolved using other sequencing technologies such as PacBio. It can produce much longer sequences that span these regions. However, these technologies are not applicable to aDNA as most of the fragments contained in the sample are even shorter than the sequences that can be produced using the Illumina platforms.

In general, it can be concluded that assembly of aDNA is highly dependent on the amount of endogenous DNA in the sample. We are able to improve results generated by current assembly programs. However, the information gain generated by the second layer assembly is dependent on the quality of the first layer assemblies. Thus if the first layer assemblies are of low quality, the second layer assembly cannot significantly improve them.

The runtime scales linearly with the number of input reads, which is no problem for small bacterial datasets. However, big projects like the assembly of human specimen does not seem to be feasible. Nevertheless it has to be kept in mind that the current pipeline currently consists of bash scripts that have not been optimized for parallelization. Using more threads on optimized code might make this approach feasible even for large genomes.

We have shown that the concept of our two-layer approach can improve the assembly of aDNA samples. The results in this study were generated using several scripts. In order to facilitate other researchers to use our two-layer approach, we are currently developing an automated pipeline containing all the steps described above. In the meantime, we provide a shell script that can perform the two-layer assembly with SOAPdenovo2 in the first layer up to the removal of small contigs after the second layer assembly. This script as well as the supplementary material can be downloaded from https://lambda.informatik.uni-tuebingen.de/gitlab/seitz/MADAM.

ACKNOWLEDGMENTS

REFERENCES

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–10.

Avila-Arcos, M. C., Cappellini, E., Romero-Navarro, J. A., Wales, N., Moreno-Mayar, J. V., et al. (2011). Application and comparison of large-scale solution-based DNA capture-enrichment methods on ancient DNA. *Sci. Rep.*, 1:74.

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. a., Dvorkin, M., et al. (2012). SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. *Journal of Computational Biology*, 19(5):455–477.

Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., et al. (2008). Accurate whole human genome sequencing using reversible terminator chemistry. - Supplement. *Nature*, 456(7218):53–9.

Bukau, B. and Horwich, A. L. (1998). The Hsp70 and Hsp60 chaperone machines. *Cell*, 92(3):351–366.

Chao, R., Yuan, Y., and Zhao, H. (2015). Recent advances in DNA assembly technologies. *FEMS Yeast Research*, 15(1):1–9.

Der Sarkissian, C., Allentoft, M. E., Ávila-Arcos, M. C., Barnett, R., Campos, P. F., et al. (2015). Ancient genomics. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 370(1660):20130387.

Durai, D. A. and Schulz, M. H. (2016). Informed kmer selection for de novo transcriptome assembly. *Bioinformatics*, page btw217.

Earl, D., Bradnam, K., St. John, J., Darling, A., Lin, D., et al. (2011). Assemblathon 1: A competitive assessment of de novo short read assembly methods. *Genome Research*, 21(12):2224–2241.

Ferragina, P. and Manzini, G. (2000). Indexing Compressed Text. *Journal of the ACM*, 52(4):552–581.

Gnerre, S., Maccallum, I., Przybylski, D., Ribeiro, F. J., Burton, J. N., et al. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proceedings of the National Academy of Sciences of the United States of America*, 108(4):1513–8.

9/10

Gnirke, A., Melnikov, A., Maguire, J., Rogov, P., Leproust, E. M., et al. (2009). Solution Hybrid Selection with Ultra-long Oligonucleotides for Massively Parallel Targeted Sequencing. 27(2):182–189.

Hodges, E., Xuan, Z., Balija, V., Kramer, M., Molla, M. N., et al. (2007). Genome-wide in situ exon capture for selective resequencing. *Nature genetics*, 39(12):1522–7.

Knapp, M. and Hofreiter, M. (2010). Next generation sequencing of ancient DNA: Requirements, strategies and perspectives. *Genes*, 1(2):227–243.

Lazaridis, I., Patterson, N., Mittnik, A., Renaud, G., Mallick, S., et al. (2014). Ancient human genomes suggest three ancestral populations for present-day Europeans. *Nature*, 513(7518):409–413.

Li, D., Liu, C. M., Luo, R., Sadakane, K., and Lam, T. W. (2014). MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, 31(10):1674–1676.

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv*, 00(00):3.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760.

Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., et al. (2010). De novo assembly of human genomes with massively parallel short read sequencing. *Genome Research*, 20(2):265–272.

Li, Z., Chen, Y., Mu, D., Yuan, J., Shi, Y., et al. (2012). Comparison of the two major classes of assembly algorithms: Overlap-layout-consensus and de-bruijn-graph. *Briefings in Functional Genomics*, 11(1):25–37.

Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., et al. (2012). SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience*, 1(1):18.

Manber, U. and Myers, G. (1993). Suffix Arrays: A New Method for On-Line String Searches. *SIAM Journal on Computing*, 22(5):935–948.

Maricic, T., Whitten, M., and Pääbo, S. (2010). Multiplexed DNA sequence capture of mitochondrial genomes using PCR products. *PLoS ONE*, 5(11):9–13.

Mendum, T. a., Schuenemann, V. J., Roffey, S., Taylor, G. M., Wu, H., et al. (2014). Mycobacterium leprae genomes from a British medieval leprosy hospital: towards understanding an ancient epidemic. *BMC genomics*, 15(1):270.

Okonechnikov, K., Conesa, A., and García-Alcalde, F. (2015). Qualimap 2: Advanced multi-sample quality control for high-throughput sequencing data. *Bioinformatics*, 32(2):292–294.

Peltzer, A., Jäger, G., Herbig, A., Seitz, A., Kniep, C., Krause, J., and Nieselt, K. (2016). EAGER: Efficient Ancient Genome Reconstruction. *Genome Biology*, 17(1):60.

Peng, Y., Leung, H. C. M., Yiu, S. M., and Chin, F. Y. L. (2010). IDBA - A practical iterative De Bruijn graph De Novo assembler. In *Annual International Conference on Research in Computational Molecular Biology*, volume 6044, pages 426–440. Springer.

Rasmussen, M., Li, Y., Lindgreen, S., Pedersen, J. S., Albrechtsen, A., et al. (2010). Ancient human genome sequence of an extinct Palaeo-Eskimo. *Nature*, 463(7282):757–762.

Sawyer, S., Krause, J., Guschanski, K., Savolainen, V., and Pääbo, S. (2012). Temporal patterns of nucleotide misincorporations and DNA fragmentation in ancient DNA. *PLoS ONE*, 7(3).

Schatz, M. C., Delcher, A. L., and Salzberg, S. L. (2010). Assembly of Large Genomes using Cloud Computing. *Genome research*, 20(9):1165–1173.

Schubert, M., Ermini, L., Der Sarkissian, C., Jónsson, H., Ginolhac, A., et al. (2014). Characterization of ancient and modern genomes by SNP detection and phylogenomic and metagenomic analysis using PALEOMIX. *Nature Protocols*, 9(5):1056–1082.

Schuenemann, V. J., Singh, P., Mendum, T. A., Krause-Kyora, B., et al. (2013). Genome-Wide Comparison of Medieval and Modern Mycobacterium leprae. *Science*, 179(July):179–184.

Simpson, J. T. and Durbin, R. (2012). Efficient de novo assembly of large genomes using compressed data structures. *Genome Research*, 22(3):549–556.

Utturkar, S. M., Klingeman, D. M., Land, M. L., Schadt, C. W., Doktycz, M. J., Pelletier, D. A., and Brown, S. D. (2014). Evaluation and validation of de novo and hybrid assembly techniques to derive high-quality genome sequences. *Bioinformatics*, 30(19):2709–2716.