APPLICATIONS OF HIGH THROUGHPUT TECHNOLOGIES IN MODERN GENOMICS

A Dissertation

Presented in Partial Fulfillment of the Requirements for the

Degree of Doctorate of Philosophy

with a

Major in Bioinformatics and Computational Biology

in the

College of Graduate Studies

University of Idaho

by

Samuel S. Hunter

April 2014

Major Professor: Larry Forney, Ph.D.

Authorization to Submit Dissertation

This dissertation of Samuel S. Hunter, submitted for the degree of doctorate of philosophy with a major in Bioinformatics and Computational Biology and titled "Applications of high throughput technologies in modern genomics" has been reviewed in final form. Permission, as indicated by the signatures and dates below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor: ________________________________ Date: __________________
Dr. Larry Forney

Committee
Member: ________________________________ Date: __________________
Dr. James J. Nagler

Committee
Member: ________________________________ Date: __________________
Dr. Terence Soule

Committee
Member: ________________________________ Date: __________________
Dr. Barrie Robison

Program
Administrator: ________________________________ Date: __________________
Dr. Eva M. Top

Discipline's
College Dean: ________________________________ Date: __________________
Dr. Paul Joyce

Final Approval and Acceptance by the College of Graduate Studies:

________________________________ Date: __________________
Dr. Jie Chen

Abstract

DNA microarrays and high throughput sequencing (HTS) have allowed investigators to characterize nucleic acids (DNA and RNA) across tissues, treatments, samples, and species, providing a wealth of information in efficient and cost-effective ways. This dissertation presents an application of DNA microarrays and two novel methods for analysis of HTS data.

Bacterial and fungal species that live in and on the human body are believed to play important roles in the maintenance of health and prevention of disease. To study these communities in the human vagina, we developed the VChip, a DNA microarray with probes representing 313 strains of bacteria as well as 716 human immunity genes. This array was validated using mock bacterial communities and tested using DNA and cDNA from vaginal swabs. The VChip produced results that accurately reflected the composition of the mock bacterial communities, and produced results similar to those obtained from 16S rRNA amplicon pyrosequencing.

Assembly by Reduced Complexity (ARC), is a software package that facilitates iterative, reference seeded assembly of HTS datasets. This strategy is useful for datasets that can be divided into several discreet subsets that can each be assembled independently. A set of reference or “target” sequences is used to recruit initial subsets of reads, each subset is assembled independently into contigs, these contigs are then used to recruit a new set of reads. This process is iterated, to grow assemblies until stopping conditions are met. I showed that ARC works well even with moderately divergent references, and is not plagued by reference bias, a serious limitation of mapping based strategies.

StopGap is a strategy for improving genome assemblies. Gap spanning Pacific Biosciences continuous long reads are identified and used to guide assembly of high quality Illumina or 454 reads with the ARC pipeline. Two assembly merging programs were tested for their ability to take advantage of these gap-bridging contigs. I show that this approach was able to produce more contiguous assemblies and better represent repeated sequences within the assembly. Although StopGap was used here to improve the assembly of a bacterial genome, this approach could be used in the assembly of more complex eukaryotic genomes as well.

Vita

Education

- 2006-2014 PhD Bioinformatics & Computational Biology (In Progress) from the University of Idaho.
- 2006-2010 MS Statistics from the University of Idaho. GPA 3.85 out of 4.0.
- 1998-2003 BS Double major in Biology and Math/Computer Science from College of Idaho. GPA 3.66 out of 4.0

Publications

1. **Hunter SS**, Yano H, Loftie-Eaton W, Hughest J, De Gelder L, Stragier P, De Vos P, Settles ML, Top EM. Draft Genome Sequence of Pseudomonas moraviensis R28-S. Genome Announcements. 2014 Feb 20;2(1).
2. Sherpa T, Lankford T, McGinn TE, **Hunter SS**, Frey RA, Sun C, Ryan M, Robison BD, Stenkamp DL, Retinal regeneration is facilitated by the presence of surviving neurons. Developmental Neurobiology. 2014 Feb 1.
3. Zhbannikov IY, **Hunter SS**, Settles LM, Foster James A. SlopMap: a software application tool for quick and flexible identification of similar sequences using exact k-mer matching. ArXiv:1307.8407. 2013 July 31.
4. Johnson TJ, Abrahante JE, **Hunter SS**, Hauglund M, Tatum FM, Maheswaran SK, Briggs RE. Comparative genome analysis of an avirulent and two virulent strains of avian *Pasteurella multocida* reveals candidate genes involved in fitness and pathogenicity. BMC Microbiology. 2013 May 13:106.
5. Abrahante JE, Johnson TJ, **Hunter SS**, Maheswaran SK, Hauglund MJ, Bayles DO, Tatum FM, Briggs RE. Draft Genome Sequences of Two Virulent Serotypes of Avian *Pasteurella multocida*.. Genome Announcements. 2013 Jan; 1(1): e00058-12.
6. Sherpa T, **Hunter SS**, Frey RA, Robison BD, Stenkamp DL. Retinal proliferation response in the buphthalmic zebrafish, bugeye. Experimental Eye Research. 2011 Oct;93(4):424-36.
7. Nagler JJ, Cavileer T, **Hunter S**, Drew R, Okutsu T, Sakamoto T, Yoshizaki G. Non-sex specific genes associated with the secondary mitotic period of primordial germ cell proliferation in the gonads of embryonic rainbow trout (*Oncorhynchus mykiss*). Molecular Reproduction and Development 2011 Mar;78(3):181-7.
8. T. Cavileer, **S. Hunter**, T. Okutsu, G. Yoshizaki, J.J. Nagler. Identification of Novel Genes Associated with Molecular Sex Differentiation in the Embryonic Gonads of Rainbow Trout (*Oncorhynchus mykiss*). Sexual Development**.** 2009; 3(4):214-224.

Publications in review

1. Paulraj Lawrence, Russel Bey, Danielle McKeown, **Samuel Hunter**, Randy Simonson. Insights into Streptococcus suis Genome: Map-Based Comparative Genomic Analysis. BMC Microbiology. (submitted).
2. Tim Cavileer, **Sam Hunter**, Jeffery Olsen, John Wenburg, and James J. Nagler. A sex determining gene (sdY) assay shows discordance between phenotypic and genotypic sex in wild populations of Chinook Salmon Oncorhynchus tshawytscha. (submitted for review)

Publications in preparation

1. **Samuel S. Hunter**, Robert T. Lyon, Brice Sarver, Kayla Hardwick, Larry J. Forney, Matthew L. Settles. ARC: Assembly by Reduced Complexity. (in prep)

Acknowledgements

I greatly appreciate the time and effort my advisor Larry Forney has invested in helping me to complete this dissertation. Without his insights, encouragement, support completing this degree would not have been possible.

I would like to thank my other committee members, Terry Soule, Barrie Robison, and James Nagler. I have had the opportunity to work closely with each of them on a variety of projects during my time in the program and have benefited greatly from these experiences.

I would like to acknowledge the IBEST community which I have had the great privilege to be involved with.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The use of DNA microarray and high throughput DNA sequencing (HTS) technologies have driven major developments in biological research and stimulated collaborations with experts in computer science and statistics. The large and often complicated datasets produced by these technologies have made it necessary to combine expertise from each of these disciplines to extract biological meaning from otherwise obtuse data. Both DNA microarrays and HTS technologies produce large and complex data sets that pose significant challenges for analysis, requiring the development of many new methods and creating a growing need for scientists with the necessary skills and training to analyze these data. Interestingly, both of these technologies attempt to gain insight into the biology of a sample through analysis of nucleic acid sequences (DNA and RNA), the information storage and transfer systems [1] thought to be common to all of known of life. Although both of these technologies are routinely used to study nucleic acids, microarray analysis ultimately results in a matrix of numeric values representing nucleic acid abundance, while HTS produces millions of short sequences known as 'reads'. These very different types of data require equally different approaches for analysis.

Early developments in DNA sequencing [2, 3] and the invention and marketing of automated DNA sequencing instruments in 1986 by Applied Biosystems [4] occurred concomitantly with the development of early microarray-like strategies [5]. However, microarrays proved to be much easier and cheaper to scale up. This was achieved either by using robots that could print (or “spot”) DNA probe sequences onto glass slides quickly and efficiently [6] or through processes of synthesizing oligonucleotide sequences directly on a substrate [7, 8]. A comparably compact and automated commercial solution for HTS did not appear until much later with the first publicly available release of a pyrosequencing system from 454 Life Sciences in 2004 [9].

Eventually, microarrays became well established as a standard tool in biological laboratories and a consensus for best practices regarding experimental design and data analysis emerged [10]. In contrast, no consensus for best practices has emerged for the analysis of HTS data as evidenced by Assemblathon 2, a recently published competition in which strategies for assembly of three vertebrate genomes were compared [11]. The authors conclude that “the clear take-home message from this exercise was the lack of consistency between assemblies”. Similarly, this conclusion was also drawn from another comparison of assemblers for bacterial genomes [12] in which assemblies were quite variable across eight sequence assemblers tested on twelve data sets. At the same time, other authors have noted serious discrepancies between algorithms used for identifying single nucleotide variants [13], and the need for continued development of algorithms for identifying insertions and deletions [14].

In this thesis, I present results from studies involving both microarray, and HTS technologies. Chapter 2 details a study in which we leveraged mature microarray design, fabrication, and analysis technologies and developed a novel platform, the VChip, for studying the gene content of bacterial communities within the human vagina. This array was developed using recent data on the species composition of vaginal microbiota in a large cohort of healthy women [15, 16]. It targets coding sequence from 313 bacterial strains representing 184 species, as well as probes for 716 genes that encode components of the human immune, stress response, and other systems. The array was designed using 1.4 million probes with the NimbleGen three-plex 4.2 million probe custom format, making it possible to hybridize three samples per slide. We validated the VChip by characterizing species and gene composition of mock communities and with human vaginal community samples.

Chapter 3 describes the development of a novel method and software package for the analysis of HTS data called Assembly by Reduced Complexity (ARC). In many cases, only a subset of the reads produced as the result of high throughput sequencing are of interest. For example, a sequencing library may be prepared from total DNA extracted from a tissue, but only an assembly of reads derived from the mitochondrial genome is of interest (in the context of HTS, the term 'library' is used to indicate a set of DNA fragments prepared for sequencing).

Alternatively, in the case where “target enrichment” strategies such as sequence capture [17] are employed, a set of DNA probes (targets) are used to enrich a library for a specific subset of template sequences through hybridization. This “captured” library contains a mix of sequences, some of which are associated with the DNA probes of interest and others of which simply were not washed away completely during the capture process. After sequencing, the reads associated with the targets of interest are referred to as “on target” while the rest are “off target”.

A major challenge for analysis of these types of datasets is the large number of reads that are not of interest. These reads make *de novo* assembly time consuming and in some cases intractable. An alternative strategy is to align (map) reads against an existing reference sequence, however mapping software is sensitive to differences (divergence) between sample and reference sequences making it difficult to map reads in regions of the reference where divergence exist [14].

The ARC approach was designed for situations like this, in which the primary objective is not the assembly of entire genomes. It facilitates recruitment and assembly of subsets of reads associated with a set of reference sequences (called “targets” here) and uses an iterative approach to account for divergent regions. ARC was shown to work well, even when references are divergent enough to completely preclude a traditional mapping based approach. I describe the algorithm as an iterative, reference seeded approach for assembly of homologous sequences because ARC only uses the target sequences on the first iteration and works well with divergent targets.

In Chapter 4 I describe an alternative use of the ARC pipeline as a strategy for improving genome assemblies. Currently, one can expect that assembly algorithms will produce assemblies that only poorly represent the original DNA sequence. Instead of a single long sequence representing each chromosome, assemblies tend to be “fragmented” into multiple sequences (contigs) and have little or no information about the order and orientation of these contigs. The typical cause for assembly fragmentation is repeated sequences that commonly occur in most genomes [18, 19]. The emergence of single molecule real time sequencing [currently only available from Pacific Biosciences (PacBio)], presents the opportunity to solve this problem due to the exceptionally long read length produced by this technology. However, PacBio reads have such low quality it is difficult to directly incorporate them into genome assemblies. To overcome this, I explored the feasibility of using ARC to combine the length of PacBio reads and Illumina or 454 reads to generate high-quality contig assemblies that bridge repeats, making it possible to incorporate them into the genome assembly, thereby closing gaps and reducing DNA sequence fragmentation overall.

In summary, Chapter 2 describes the VChip, a microarray designed and validated for characterizing the gene content of the vaginal microbiome. Chapter 3 describes a software tool designed for reference seeded assembly of homologous sequences and shows that this tool is effective for assembly of mitochondrial genomes under a variety of circumstances. Chapter 4, presents a strategy to reduce the fragmentation of a genome assembly by using long PacBio reads in combination with Illumina or 454 short reads.

Chapter 2

VChip, a DNA microarray for exploratory analysis of the vaginal microbiome

Abstract

We report on the development and validation of the VChip, a DNA microarray for exploratory analysis of the species and gene composition of vaginal microbiota. The three-plex glass slide microarray consists of 1.4 million 60-mer probes per sub-array derived from the coding genome sequences of 313 strains representing 184 bacterial species commonly found in the vagina and 716 human immunity genes. We performed a series of validation experiments with the VChip to demonstrate its efficacy with both mock bacterial communities and clinical vaginal swabs. Through these experiments we confirmed that the VChip correctly detected expected species in DNA from mock communities and in both DNA and cDNA from vaginal swabs. Furthermore, it was highly sensitive for both high- and low-abundance bacterial taxa and human immunity genes. While not strictly quantitative, the VChip produced similar overall patterns of species and gene presence as high-throughput sequencing methods including 16S rRNA amplicon pyrosequencing and Illumina shotgun sequencing of vaginal swab metagenomes and metatranscriptomes. We conclude the VChip is suitable for exploratory qualitative analysis of the vaginal microbiome in both research and clinical settings with a variety of potential applications including association of gene expression with health outcomes or rapidly screening a large number of samples for patterns of interest.

Introduction

The vaginal microbiota is a complex assemblage of microbes believed to play a significant role in the maintenance of health and prevention of disease [1, 2]. Numerous studies employing cultivation-independent techniques have found that bacterial communities in the vaginas of healthy, reproductive-age women vary markedly, both within and among individuals as well as over time [3-8]. Community structure and composition differ considerably, with some communities highly skewed toward a single predominant species (most often Lactobacillus spp.) and others characterized by a more even distribution of assorted anaerobic bacterial taxa. There appear to be at least five major community types among reproductive age women, and their prevalence varies somewhat across racial and ethnic groups [6]. Furthermore, vaginal microbial community composition is dynamic in many women with patterns ranging from constant to rapidly fluctuating, and those patterns may vary over time within an individual [8, 9]. The causes and consequences of variation in community composition and structure are presently not well understood and are thus an active area of research.

Along with other advancements in understanding the human microbiome, recent discoveries about the vaginal microbiome have been facilitated by high-throughput sequencing technologies, decreased costs and improved efficiency. Most cultivation-independent studies to date have relied primarily on phylogenetic analysis of partial 16S rRNA gene sequences or other conserved genes to survey bacterial taxa [10-15]. More recently, however, there has been growing interest in determining not only which taxa are present in various host-associated microbial communities, but also their function and how they modulate human health. Analyses of this nature warrant a more comprehensive approach that takes into account not just a single gene, but rather a larger suite of genes present or expressed within a microbial community. In response to our limited understanding of the relationships between specific genes and metabolic and ecological functions, more comprehensive community metagenomics techniques enable characterization of the total genomic content of microbial communities [16, 17]. These approaches have fundamentally changed microbial ecology research and our understanding of microbial community function in human and environmental health. However, while many bioinformatic tools and pipelines have been or are actively being developed to handle and interpret such data, the functional assignment of metagenomic sequence data remains a significant challenge [18].

Despite major advances in sequencing technologies, the analysis and interpretation of vast quantities of sequence data require non-trivial bioinformatics expertise and computational resources, posing a barrier to rapid analysis of samples in many smaller research or clinical settings. Because a central goal of human microbiome research is to develop strategies of personalized medicine tailored to an individual's microbial makeup [13], there is a pressing need for research tools that enable fast, simplified analysis of greater numbers of samples. One strategy to fulfill this need is the development of straightforward research tools that require simplified and streamlined bioinformatics analysis.

DNA microarrays are a faster, less expensive alternative to DNA sequencing. They are relatively simple to use, rely on established laboratory techniques and can be analyzed using statistical methods shown to be robust across many datasets [19]. Microarrays have been used in microbial ecology research for more than a decade [20, 21]. These have primarily consisted of phylogenetic microarrays that probe group-specific marker genes (e.g., 16S rRNA) used to classify taxa in a community or estimate species richness [22-25] much the same way that 16S rRNA amplicon sequencing is used to characterize community composition. In addition, some arrays also contain probes for genes of functional relevance to assess specific functional characteristics of a sample [26-28]. Numerous microarrays have been developed for both environmental (terrestrial and aquatic) and human-associated bacterial communities, and these have previously been reviewed [21, 29, 30]. Arrays differ in both the number of probes and taxonomic level assessed, the array platform used, accommodation of DNA or RNA samples, and the specific environment(s) for which they are applicable.

We developed a DNA microarray, termed the VChip, for exploratory analysis of community gene content and expression in the vaginal microbiome. The motivations for this project were an incomplete understanding of the ecological functions of vaginal microbiota and their role in women's health as well as a lack of high-throughput research tools to rapidly evaluate the composition of vaginal microbial metagenomes and metatranscriptomes. Using recent data on the species composition and relative abundance of vaginal microbiota in a large cohort of healthy women [6] and publicly available genome sequences of host-associated vaginal bacteria, we designed an extensive probe set (1.4 million 60-mer probes per sub-array of a three-plex glass slide array) targeting the genomes of 313 strains representing 184 bacterial species found in the vagina as well as 716 human immunity genes. We performed a series of validation experiments with the VChip to demonstrate its efficacy for both mock bacterial communities as well as DNA and cDNA prepared from clinical vaginal swabs from healthy adult women. We compared output from the VChip with 16S rRNA amplicon pyrosequencing data and demonstrated that it performs well in detecting DNA and cDNA from both well-represented and low-abundance taxa. Our findings support the utility of VChip as a discovery tool with a variety of potential applications, including identification of genes or taxa that contribute to community ecological function or health outcomes of the host, as well as rapidly screening a large number of samples for interesting patterns to select a manageable subset for more in depth investigations.

Methods

Probe set design and array production

Data from previous studies of the vaginal microbiome [6, 8] and bacterial genomes (draft and complete) available in the Genomes Online Database (http://www.genomesonline.org) guided the selection and design of probes for the DNA array. The NimbleGen array format was a three-plex 4.2 million custom designed array (1.4 million probes per sub-array). Development of the probe sets began with the coding genome sequences of 336 bacterial strains of 200 species associated with the vaginal microbiota, including a wide range of bacteria that are common and often highly abundant (e.g., Lactobacillus species) as well as numerous species that are only detected occasionally or in low relative abundance (this was later reduced to 313 strains/184 species due to maximum capacity of the microarray; see Results for explanation). Gene sequences less than 10 bp or greater than 9,999 bp were excluded, and the remaining sequences were clustered based on 80% identity and 80% coverage thresholds using the clustering program Cd-hit [31]. Multiple sequence alignments were performed in MUSCLE [32] to generate a representative consensus sequence for each cluster, replacing ambiguous or heterogeneous alignment sites with “N”. Consensus sequences were submitted to Roche NimbleGen (Madison, WI, USA) to design unique 60-mer probes for each cluster. Final array design and production were completed in collaboration between Roche NimbleGen and the Institute for Bioinformatics and Evolutionary Studies (IBEST) Genomics Resources Core at the University of Idaho.

Bacterial strains and human vaginal swabs

Nine mock communities with different combinations and proportions of genomic DNA from six bacterial strains in addition to human female genomic DNA (Promega, Madison, WI, USA) were created as shown in Table 1. The latter was included to evaluate the effect of human DNA burden on bacterial species detection and the level of cross-hybridization with probes targeting bacteria. The mock communities consisted of varying combinations of bacterial and human female genomic DNA, wherein bacterial DNA ranged from 0-100% of the total amount of DNA (2.5 µg total). The bacterial strains included three type strains: *Lactobacillus crispatus* ATCC 33820, *Atopobium vaginae* ATCC BAA-55, and *Gardnerella vaginalis* ATCC 14018. Additionally we included three bacterial isolates collected from vaginal swabs that were classified as *Finegoldia magna*, *Anaerococcus tetradius*, and *Anaerococcus hydrogenalis* based on >97% similarity in the V1-V5 region of 16S rRNA gene sequences in the NCBI GenBank database (http://www.ncbi.nlm.nih.giv/genbank/).

In addition to the mock communities, three clinical vaginal swabs from a 10-week longitudinal study of the vaginal microbiome (Gajer et al., unpublished) were included in the analysis. The study received Institutional Review Board approval from the University of Maryland School of Medicine. Sample VM-1 was from an individual (subject 1) whose vaginal microbiota was stably dominated by

Lactobacillus iners over the course of the study (Fig. 1A). Samples VM-2 and VM-3 were from two time points in a second individual (subject 2) collected at a time when her microbiota experienced an apparent disturbance that shifted the community composition during the eighth week of the study before returning to a *Lactobacillus* spp.-dominant state shortly thereafter (Fig. 1B). DNA and RNA were extracted from these samples for evaluation with the VChip.

Extraction of genomic DNA from vaginal swabs and bacterial cultures

Genomic DNA was extracted from vaginal swabs that had been stored in Amies transport media at -80°C. A validated procedure [6, 33, 34] was used that includes steps for enzymatic and physical lysis of bacterial cells followed by purification of genomic DNA using a QIAsymphony robotic platform and Qiagen CellFree500 kits (Qiagen, Venlo, Limburg, Netherlands) according to the manufacturer’s protocol. The same protocol was performed manually for extracting genomic DNA from bacterial pure cultures used to construct the mock communities.

V1-V2 16S rRNA Roche 454 pyrosequencing and Illumina RNA-Seq metatranscriptome sequencing of vaginal swabs

The three vaginal swabs tested on the microarray were characterized using Roche 454 pyrosequencing of 16S rRNA gene V1-V2 hypervariable regions as previously reported [8]. These data served as a comparison to the VChip microarray hybridization results to determine whether the species detected were reasonably consistent between methods.

Metatranscriptomic cDNA were prepared by extracting total RNA from vaginal swabs that had been stored in RNAlater and depleted of rRNA using a combination of Epicentre (Madison, WI, USA) Ribo-Zero rRNA removal kits for bacteria and human/mouse/rat kits. The remaining mRNA was reverse transcribed, and the resulting double stranded cDNA (~200 ng) was used to construct sequencing libraries for sequencing on an Illumina HiSeq 2000 instrument (Illumina, San Diego, CA, USA) at the University of Maryland Institute for Genome Sciences (IGS) using protocols recommended by the manufacturer and modified by the Genomic

Resource Center at IGS. The abundances of individual transcripts were determined based on the depth of read coverage.

Sample processing, microarray hybridization and analysis

Genomic DNA from mock communities and vaginal swabs and cDNA from vaginal swabs were processed in the Genomics Resources Core facility of the Institute for Bioinformatics and Evolutionary Studies (IBEST) at the University of Idaho following NimbleGen's protocols for comparative genomic hybridization (CGH) arrays (version 8.1, July 2011) [35]. Briefly, 0.5 µg of purified, unamplified, unfragmented genomic DNA or cDNA was labeled with high-efficiency Cy5 Random Nonamers, followed by hybridization and washing as described in the manual. Prepared samples were analyzed on a Roche NimbleGen MS200 scanner (Roche NimbleGen, Madison, WI, USA) along with standard quality controls.

Hybridization intensity signals were normalized using the Robust Multichip Average (RMA) method [19]. To perform comparisons between microarray hybridization results and expected mock community composition as well as data from 16S rRNA gene pyrosequencing and Illumina metatranscriptome sequencing, we converted each species' RMA-normalized hybridization intensity value to a percentage of the total signal across the entire array as follows: first, the 95th quantile of the hybridization values for the probes associated with a given species was calculated. Next, the median of the 95th quantile values was subtracted from the 95th quantile values, and negative values were set to 0. Finally, percentages of each species' signal were calculated from the un-logged values. The rationale for these steps is that in many cases, not all probe sets within a given species will produce hybridization signal. The 95th quantile represents probes with the highest signal within the group, while taking a median or mean results in values that are essentially indistinguishable from background signal. Following normalization, the minimum signal was 3.4, the mean was 4.2, and the 95th quantile was 4.6. This suggests that all values < 4.6 (approximately) are essentially equivalent to 0 and thus considered background signal. This procedure shifts the expression values to reflect this.

Qualitative and statistical analyses were performed and visualized using custom R scripts and packages available from Bioconductor (http://bioconductor.org [36]), including oligo [37], pdInfoBuilder [38], limma [39], qvalue[40, 41] and Biostrings [42].

Evaluation of VChip with mock community genomic DNA

In order to compare the mock community hybridization results from the VChip with expected species proportions, it was necessary to scale the proportion of each species' genomic DNA by its genome size. This was accomplished by dividing each species' genomic DNA proportion in a community by its genome size to estimate the expected proportion of genome copy equivalents (reported in parentheses in Table 1). Since exact genome sizes were not known for most of the strains used in the mock communities, they were estimated from available genome assemblies of other strains of the same species (up to three per species), excluding genomes that included plasmids or organelles and giving preference to contig- and chromosome-level over scaffold-level assemblies (Supplementary Table S1). The adjusted expected proportions of genome copy equivalents were compared to the VChip species-specific normalized hybridization signal using computed Pearson correlation coefficients.

Comparison of the VChip-derived community composition and V1-V2 16S rRNA Roche 454 pyrosequencing data from vaginal swabs

To compare the VChip hybridization of DNA from vaginal swabs to 16S rRNA-derived community composition data, we considered only the subset of species in the VChip probe sets that could be matched to the 16S rRNA sequence data by name. This was necessary because many operational taxonomic units (OTUs) determined by bioinformatic analysis of the Roche 454 16S rRNA gene pyrosequencing data were not directly comparable due to nomenclature differences in the databases used for annotation. Comparisons were then carried out at the species level by plotting the results and calculating the Pearson correlation coefficient between relative abundances of taxa detected by different techniques.

In silico probe assessment of Illumina RNA-Seq metatranscriptome reads

The 60 bp microarray probe sequences were mapped against the 100 bp Illumina RNA-Seq reads using Bowtie (v0.12.9, [43]) (parameters: “-v 3 --fullref --chunkmbs 512 --best --strata -m 20”). Alignments were then filtered to only those where the full length of each probe aligned to the read, allowing for two mismatches. The numbers of reads with mapped probes were summed and converted to a percentage of total reads with mapped probes. Finally, these were compared to the metatranscriptome cDNA array hybridization results for vaginal swab samples VM-1 and VM-2 by calculating Pearson correlation coefficients.

Assessment of changes in gene expression in subject 2

To demonstrate how the VChip could be used to compare samples and identify genes that were over- or under-expressed, we compared the hybridization signals from the cDNA hybridizations from samples VM-2 and VM-3. After normalization of the data following procedures referenced above [19], we calculated the log2-fold difference in expression values between the two samples. The number of gene clusters with >2 log2-fold difference (in magnitude) were determined for each species and averaged to evaluate whether overall change in gene expression within each species was up, down, or neutral in VM-3 relative to VM-2.

Results

VChip probe set design

The starting set of 336 bacterial strains (200 species) encompassed 812,653 coding sequences (CDS) of which 119,091 were duplicates (i.e., redundant). 240 CDS fell outside the range of 10-9,999 bp and these were excluded from further analysis. The remaining 693,322 CDS clustered into 473,709 clusters (364,808 singletons), and a consensus sequence for each cluster was generated as described above and submitted to Roche NimbleGen for probe design. Initially five probes per cluster were designed, but because the total number exceeded the capacity of the array (~2.4 million probes versus 1.4 million available per sub-array), we manually reduced the starting set of bacterial genomes to be represented down to 313 (184 species). After excluding the removed strains, 307,860 of the original 473,709 gene clusters were represented in the final array design in addition to 716 human immunity genes (3,580 probes). 74.4% of the bacterial gene clusters (n=229,131) had five probes, and all had at least two probes. Supplementary Table S2 includes bacterial gene cluster information and annotations, and Supplementary Table S3 contains similar information about the human immunity genes. Of the 313 strains represented on the array, 246 (78.6%) had strain-specific probe sets while the rest were represented by probe sets shared with other strains or species as listed in Supplementary Table S4. 165 of the 184 species on the array (89.7%) had species-specific probe sets (Supplementary Table S5). Supplementary Table S6 is the NimbleGen Design File (NDF) containing the probe sequences and detailed information about the design of the array. An explanation of the variables within this file can be found in the NimbleScan Software User’s Guide (version 2.5), pp. 93-94 [44].

VChip hybridization with mock community genomic DNA

We first tested mock communities composed of bacterial and human genomic DNA (Table 1) to measure the sensitivity of the VChip to detect genes from known species present in varying proportions relative to one another. Fig. 2 shows the proportions of normalized species-specific probe hybridization signal in the mock communities on the array compared to the expected proportions of genome equivalent copies. The VChip correctly and specifically detected both the bacterial species and human DNA present, and Pearson correlation coefficients ranged from 0.34 to 0.95 (mean 0.77, median 0.85). Mock communities with only a single species had the highest correlations (MC-8, 100% L. crispatus: r=0.95; MC-9, 100% human: r=0.92), while communities consisting of only bacteria in skewed proportions had the lowest correlations (MC-2: r=0.34; MC-3: r=0.60). Communities with both bacterial and human DNA also had relatively high correlation coefficients (MC-4: r=0.73; MC-5: 0.85; MC-6: r=0.85; MC-7: r=0.82), as did the mock community with balanced proportions of five bacterial species (MC-1: r=0.85; however, see below). These results demonstrate the VChip is capable of detecting genes or DNA fragments from both high and low-abundance bacteria in mixed samples with or without human DNA. Because the correlations of expected and observed species proportions on the VChip appear to be influenced somewhat by the distribution of species' abundance in the community, the VChip should not be used strictly as a quantitative tool; however, it may be useful for semi-quantitative (i.e., comparing relative differences across samples) or predictive purposes to be validated by further sequencing efforts.

Hybridization signals for probe sets assigned to species that were not present in the mock communities accounted for approximately 10% of the total signal overall, indicated by the "other" category in Fig. 2. This signal was collectively made up of very low relative proportions (typically <1%) of signal across many species and is essentially indistinguishable from background noise (the RMA signal-to-percent conversion filters out most low-level signal, but some residual background noise remains). The species detected using the VChip were concordant with our expectations in the mock communities but also indicated an unexpected presence of *L. crispatus* in one of our samples. *L. crispatus* probe sets on the array hybridized with mock community MC-1 showed strong signal for the presence of this species even though we had not knowingly added its DNA during sample preparation. Comparisons with other arrays hybridized with samples containing *L. crispatus* showed that the same set of probes had high levels of hybridization in both cases, suggesting contamination of this sample and not a failure of the array (see Supplementary Appendix S1).

Comparison of VChip-derived community composition and V1-V2 16S rRNA Roche 454 pyrosequencing data from vaginal swabs

As with the mock communities, the VChip performed well in correctly detecting both high- and low-abundance bacteria in the vaginal swab metagenomic DNA. We compared the VChip-derived bacterial community composition from the three vaginal swabs with species relative abundances based on V1-V2 16S rRNA Roche 454 pyrosequencing using computed Pearson correlations. These comparisons were performed for 42 bacterial species that had species name matches between the VChip probe dataset and the pyrosequencing dataset; collectively these accounted for at least 80% of the hybridization signal from the samples on the array. Of those 42 species, 15 had non-zero relative abundances in the 16S rRNA pyrosequencing dataset, and these comparisons are shown in Fig. 3. The Pearson correlation between percent hybridization signal on the VChip and relative abundance based on 16S rRNA sequencing was high when the community was highly skewed toward a single dominant species. For instance, sample VM-1 was composed of >99% *Lactobacillus iners*, and the correlation coefficient of relative abundances of species based on DNA hybridization and 16S rRNA relative abundances was 1.00. Samples VM-2 and VM-3, which had a more even distribution of species relative abundances, had lower correlation values of 0.53 and 0.84, respectively. Similar to what we observed with the mock communities, the VChip was sensitive to low-abundance species that were barely detected by amplicon sequencing. This is evidenced by the points in the upper left region of Figure 3; these are species more highly detected by VChip than by V1-V2 16S rRNA pyrosequencing. These discrepancies could be due in part to a number of factors including differences in the total number of species-specific probe sets targeting each species, probe hybridization efficiency (i.e., GC-content of genes and probes), gene copy number, or depth of sequencing in the 16S rRNA data. Although we would not recommend using VChip to quantify species relative abundance in a community the same way 16S rRNA sequencing is often used, we consider the high sensitivity of VChip to low-abundance taxa a strength of the microarray that could enable detection of important genes in species that might otherwise be overlooked.

We also performed qualitative comparisons of cDNA hybridizations on the VChip with 16S rRNA pyrosequencing and Illumina shotgun sequencing data to evaluate overall similarity in the species detected. One interesting outcome from this assessment was the unexpected abundance of transcripts detected from *Finegoldia magna* in VM-1. The relative abundance of this species was just 0.05% by 16S rRNA Roche 454 pyrosequencing, and *F. magna*-specific signal constituted less than 1% of the normalized signal in the metagenomic DNA hybridization. The cDNA hybridization from the same sample, on the other hand, constituted 17.5% of the normalized signal. Additionally, 3.41% of the Illumina RNA-Seq reads from this sample mapped to *F. magna*-specific VChip probe sets in the in silico analysis described below, providing further evidence to suggest *F. magna* was expressing genes. This possibility might have been overlooked based on 16S rRNA or metagenome sequencing due to the low relative abundance of this species. While observations like this could potentially spur further investigation, it would be necessary to perform this analysis with replicated samples to make statistically supported conclusions.

In silico analysis of Illumina RNA-Seq data and VChip from vaginal swabs

We performed *in silico* hybridization by mapping VChip probe sequences against Illumina RNA-Seq reads to determine whether the two approaches yielded comparable results. 2.26 million of the 100 bp RNA-Seq reads from sample VM-1 (2.77% of 81.80 million reads) and 1.78 million reads from sample VM-2 (2.23% of 79.44 million reads) were successfully mapped to VChip probe sequences (mapping was not performed for VM-3). There are several possible explanations for why a large proportion of the reads did not get mapped to the probe sequences. First, the percentage of the mapped reads is limited by the total number (1,443,693) and design of the probes. Each gene cluster represented on the array had maximally five 60-mer probes, so only up to 300 bp of any given gene cluster could potentially be mapped onto by the RNA-Seq reads. In most cases, this would leave large portions of genes ‘unprobed’, even if other segments of the genes were accounted for in the reads. Additionally, any eukaryotic reads aside from the 716 human immunity genes would not have been detected, nor would bacterial species or strains not present on the array. Finally, although a ribosomal RNA (e.g., 16S rRNA) depletion step was included prior to Illumina RNA-Seq sequencing, ~9% of the reads in each sample were identified as rRNA bioinformatically. Given these explanations, it is not surprising that a relatively low percentage of reads were mapped to VChip probes. Of the RNA-Seq reads that successfully mapped to VChip probes, the Pearson correlation coefficients for species-specific relative hybridization abundances between the in silico mapping and the cDNA hybridization on the VChip were 0.71 and 0.79 (Table 2). Although only a small fraction of the reads mapped to the probe sequences, the high degree of correlation with the actual cDNA hybridized on the array indicates good agreement in the detection of cDNA fragments using two very different technologies.

Relative gene expression differences between VM-2 and VM-3 in subject 2

The cDNA VChip-hybridizations of samples VM-2 and VM-3 were compared to observe differences in relative gene expression between the two time points, reported in Figure 4. A complete list of gene clusters with >2 log2-fold differences (in magnitude) between VM-2 and VM-3 are included in Supplementary Table S7. Many species, including several *Lactobacillus spp.*, showed a large number of species-specific genes with increased average relative expression in the latter time point, which is not surprising considering these species were increasing in relative abundance over the time frame observed (Fig. 1B). These included genes such as a GNAT family acetyltransferase, initator RepB protein and DNA methylase in *L. iners*, as well as a cell wall-associated hydrolase and endopeptidase O in *L. gasseri*, to name just a few examples. Interestingly, a large fraction of transcripts with the greatest number of differentially expressed genes with average positive change were attributed to less abundant taxa such as *Peptoniphilus lacrimalis*, *Dialister microaerophilus* and several others. A. vaginae had the greatest number of down-expressed genes on average in the latter time point (indicating it was much more active during the earlier time point). Several other species including *Prevotella amnii*, *Anaerococcus tetradius*, *Clostridiales genomosp*. and *Mobiluncus curtisii* had large numbers of differentially expressed genes, but the average change across all species-specific genes was close to zero, meaning some genes were over-expressed in the earlier time point and others in the latter. Additionally, 190 of the 716 human immunity genes were differentially expressed based on the >2 log2-fold threshold; of these, two were down (a myeloperoxidase, NCBI reference NM_000250; and NLRC4, NCBI reference NM_021209) while the rest were up in the latter time point.

While we cannot draw statistically sound conclusions from only a pair of samples, our results demonstrate it is feasible to detect differences in patterns of gene expression between samples using the VChip. If similar comparisons were conducted on a larger scale with both sample and technical replicates, such changes in expression patterns could provide important insights into the mechanisms driving shifts in community composition and function in response to a purported disturbance.

Discussion

Tools for rapid, streamlined analysis of vaginal microbial communities are needed to accelerate our understanding of the microbiome in relation to human health and devise more effective strategies for patient care. In response to this need, we developed the VChip, a DNA microarray that targets the vaginal microbiome more comprehensively than any previous microarray and is also the first to our knowledge to include human immunity genes in addition to bacterial genes. The VChip offers a faster and more accessible approach to screening vaginal microbial communities for interesting community metagenomic or gene expression patterns. Because it is based on a standard microarray platform, data can be analyzed in a straightforward manner using established methods, and the probe sets (provided in SupplementaryTable S6) can be tailored and reproduced on any oligonucleotide platform. The results of our validation experiments with both mock communities and vaginal swab samples support the utility of VChip as an exploratory tool for assessing gene content and expression of vaginal metagenomes and metatranscriptomes, respectively. We found it to be specific at the species level and also highly sensitive to both high- and low-abundance species in mixed communities. While the VChip did not accurately reflect species relative abundance in some of our samples, it reliably detected genes and transcripts from expected bacterial species and demonstrated the potential to reveal interesting patterns of gene expression.

Microarrays developed previously for applications in human microbiome research include arrays targeting microbes residing in the human gastrointestinal tract [23, 25, 45], oral cavity [46, 47], vaginal tract [48] and general human microbiota [24]. The vast majority are phylogenetic microarrays which are used to assess the species composition and relative abundance of microbes found in these various ecosystems based on a single gene or small set of genes. The array most similar to the VChip was developed by Dols et al. for detecting the presence of bacteria associated with bacterial vaginosis using 16S rRNA amplicons [48]. The VChip differs substantially from this and other human microbiota microarrays because it probes both species-specific and conserved genes spanning whole coding genome sequences rather than a limited set of marker genes (e.g., 16S rRNA) or a subset of characterized functional genes. Moreover, because the VChip also probes human immunity genes, it could be used to evaluate the local host immune response along with vaginal microbial community gene expression.

A major advantage of the VChip's more comprehensive design is that it can be used in an exploratory manner to identify potential interactions between the host and vaginal microbiota in health and disease. The process of gene selection for probe design of the VChip was agnostic toward metabolic or physiological function, so there are many genes represented on the array that have not yet been well characterized but could ostensibly be important for community ecology of the vaginal microbiota. Furthermore, species-specific probe sets are included for 165 bacterial species, and strain-specific probe sets for 246 strains, representing a wide spectrum of taxa found in the vagina at varying degrees of incidence and relative abundance. This enables detection of gene expression patterns even with so-called "rare" or low-abundance bacteria, as was the case with *F. magna* in the vaginal community of sample VM-1. Similarly, an investigator might want to determine which species contribute most to differences or changes in gene expression patterns across samples. Our comparison of two metatranscriptome samples from subject 2 (VM-2 and VM-3) revealed that several species exhibited large changes in transcript abundance from one time point to another, which could prompt more targeted analysis in future studies. However, the hybridization results need not be partitioned by species-specific probes at all; it is also feasible to compare the entire hybridization signals to represent the "total" community gene content or expression, with the caution that the "total" is limited to genes that are already on the array. This information could be leveraged to gain a better understanding of community function as well as generate hypotheses to test in additional experiments.

There are nonetheless some important caveats to using the VChip for certain applications. The first is that VChip is not recommended for quantifying the relative abundance of species in a community. VChip may be informative for semi-quantitative and predictive purposes, but high-throughput sequencing of 16S rRNA amplicons or quantitative PCR methods are better suited for characterizing species abundance. A second caveat is that the VChip is not intended to be a direct replacement for shotgun sequencing of metagenomes and metatranscriptomes if the goal is to characterize total genomic content or gene expression. As with all microarrays, the VChip can only detect genes that are highly similar to what is represented on the array (although fortunately, the VChip targets a majority of known bacterial species found in the vagina to date). A final caveat is that the three-plex 4.2 million probe glass slide custom array format is no longer manufactured by Roche NimbleGen (Madison, WI, USA). However, future iterations of the VChip could be reproduced on alternative microarray platforms, and we have provided the necessary probe design files and additional information (Supplementary Tables S2, S3 and S6) to enable other investigators to reconstruct the array or a portion of it, depending on the platform of choice and intended application.

In summary, we have demonstrated that the VChip produces similar overall patterns of species presence as 16S rRNA amplicon pyrosequencing and Illumina shotgun sequencing, and in fact it may be better suited at detecting genetic material from low-abundance bacteria that might be missed with shallow depth of sampling or sequencing. We have shown the VChip is suitable for exploratory analysis of vaginal microbiota and that it could be particularly useful as a screening tool to characterize and select samples of interest to study in greater detail with more comprehensive sequencing methods. Additionally, it can be used in the same way as traditional microarrays to evaluate differences in gene expression of vaginal microbial communities among samples. Alternatively, the VChip could be used as a diagnostic tool in clinical settings if certain gene expression patterns (either from the microbiota or host immune system) were shown to be associated with specific conditions of interest. We conclude the VChip has potential to become a versatile research tool that could be adapted for a variety of applications.

Figures

Figure 2.1. *Daily temporal dynamics of vaginal bacterial communities in two women over 10 weeks.*

The vaginal microbiota of subject 1 is shown in (A) and subject 2 in (B). The relative abundances of phylotypes in each community are depicted as interpolated bar plots (top panel). Beneath these are profiles of Nugent scores (range 0-10) and vaginal pH (range 4-7). Occurrence of menses (red dots), vaginal intercourse (blue inverted triangles) and vaginal symptoms (pink open circles) are indicated in the bottom panel, with samples selected for VChip analysis indicated directly below.

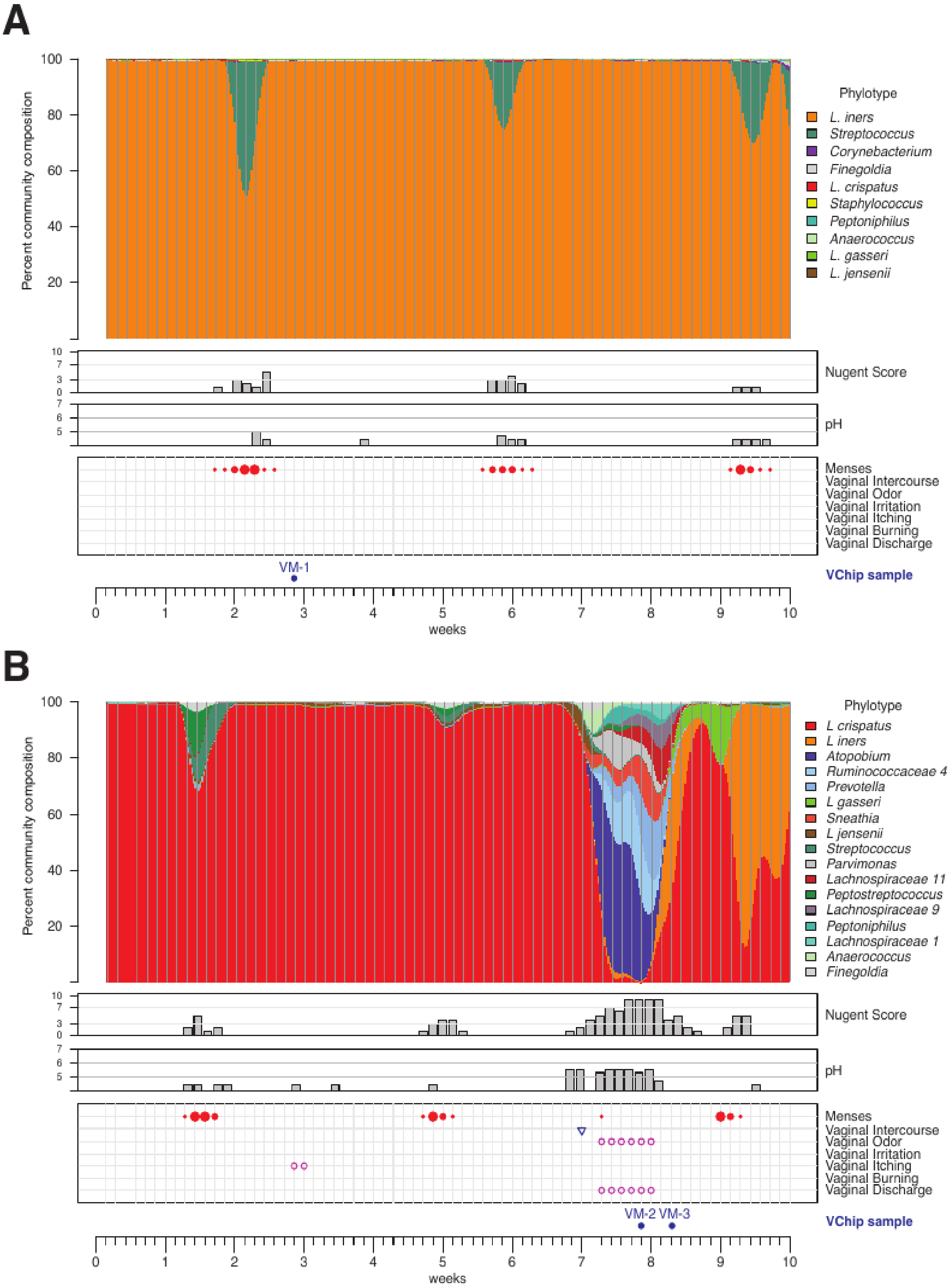

A
Percent community composition
100
80
60
40
20
Phylotype
L. iners
Streptococcus
Corynebacterium
Finegoldia
L. crispatus
Staphylococcus
Peptoniphilus
Anaerococcus
L. gasseri
L. jensenii
Nugent Score
10
7
3
0
pH
7
6
5
Menses
Vaginal Intercourse
Vaginal Odor
Vaginal Irritation
Vaginal Itching
Vaginal Burning
Vaginal Discharge
VM-1
VChip sample
0
1
2
3
4
5
6
7
8
9
10
weeks
B
Percent community composition
100
80
60
40
20
Phylotype
L crispatus
L iners
Atopobium
Ruminococcaceae 4
Prevotella
L gasseri
Sneathia
L jensenii
Streptococcus
Parvimonas
Lachnospiraceae 11
Peptostreptococcus
Lachnospiraceae 9
Peptoniphilus
Lachnospiraceae 1
Anaerococcus
Finegoldia
Nugent Score
10
7
3
0
pH
7
6
5
Menses
Vaginal Intercourse
Vaginal Odor
Vaginal Irritation
Vaginal Itching
Vaginal Burning
Vaginal Discharge
VM-2
VM-3
VChip sample
0
1
2
3
4
5
6
7
8
9
10
weeks

Figure 2.2. *Comparison of VChip-derived species composition of mock communities to expected proportions of species' genome equivalent copies.* Nine mock communities were constructed from genomic DNA as indicated in Table 1, and the proportions of DNA were converted to proportions of expected genome equivalent copies per species. The left bar in each plot indicates the expected proportions of genome equivalent copies per species in the mock community, and the right bar indicates the observed proportions of normalized hybridization signal attributed to each species on the VChip. The header of each subplot indicates the mock community label (MC-1 through MC-9) along with the Pearson correlation coefficient between observed and expected values for each sample. Species are indicated in the legend to the right of the plots. In addition to the six bacterial species and human, the 'other' category encompasses very low relative proportions (typically <1%) of signal across many species and is indistinguishable from levels of residual background noise.

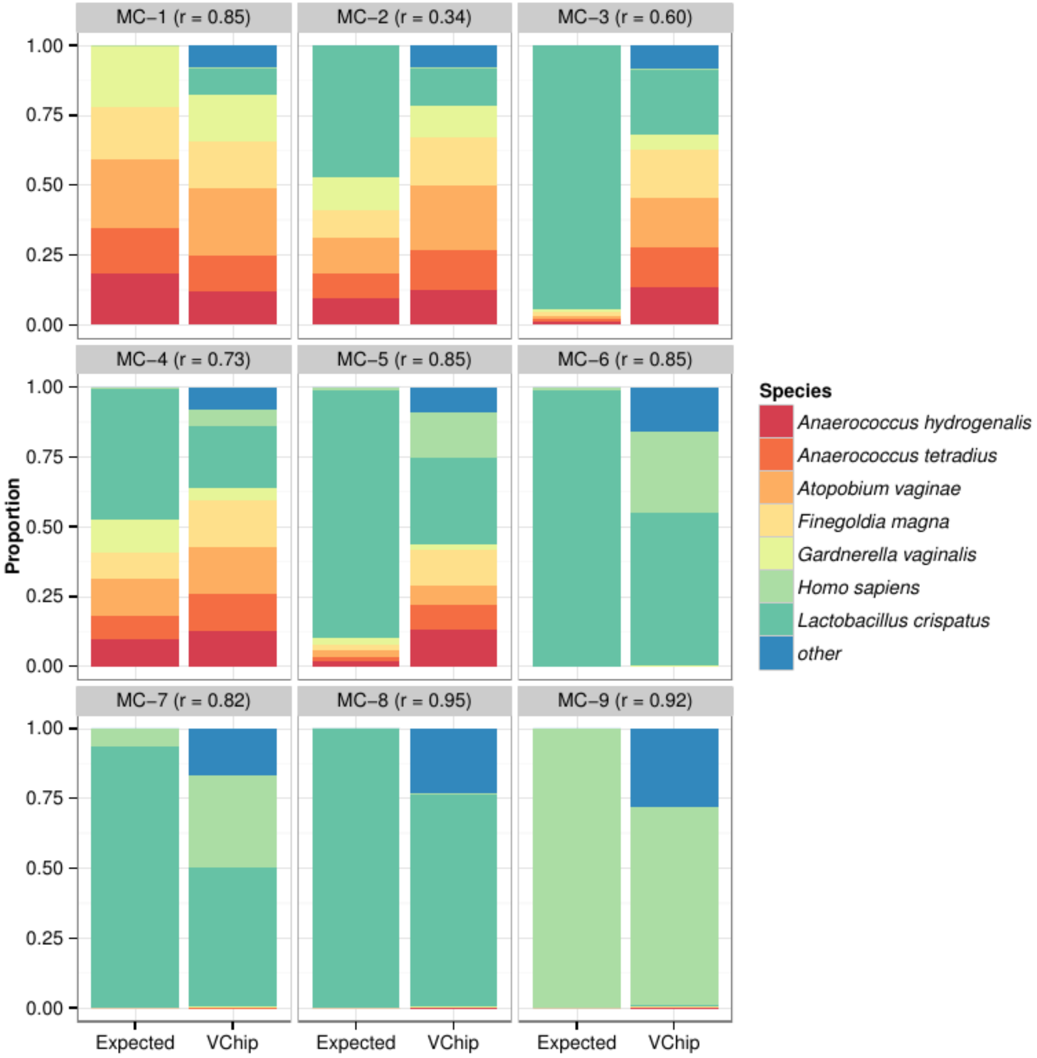


Figure 2.3. *Comparison of species relative abundance in vaginal swabs detected by VChip and 16S rRNA V1-V2 pyrosequencing.*
DNA hybridizations of three vaginal swab samples (sample VM-1, orange; VM-2, teal; VM-3, blue) were analyzed on the VChip. Hybridization signals were normalized and converted to relative abundances per species (y-axis) and compared to relative abundance data determined from V1-V2 16S rRNA pyrosequencing (x-axis). The data are plotted on a log10 scale to clearly separate out low relative abundance species. The top 15 most abundant of 42 species with exact name matches between the VChip taxa and pyrosequencing dataset are plotted as indicated in the legend to the right of the graph. Pearson correlation coefficients based on all 42 species are 1.00 for VM-1, 0.53 for VM-2 and 0.85 for VM-3.

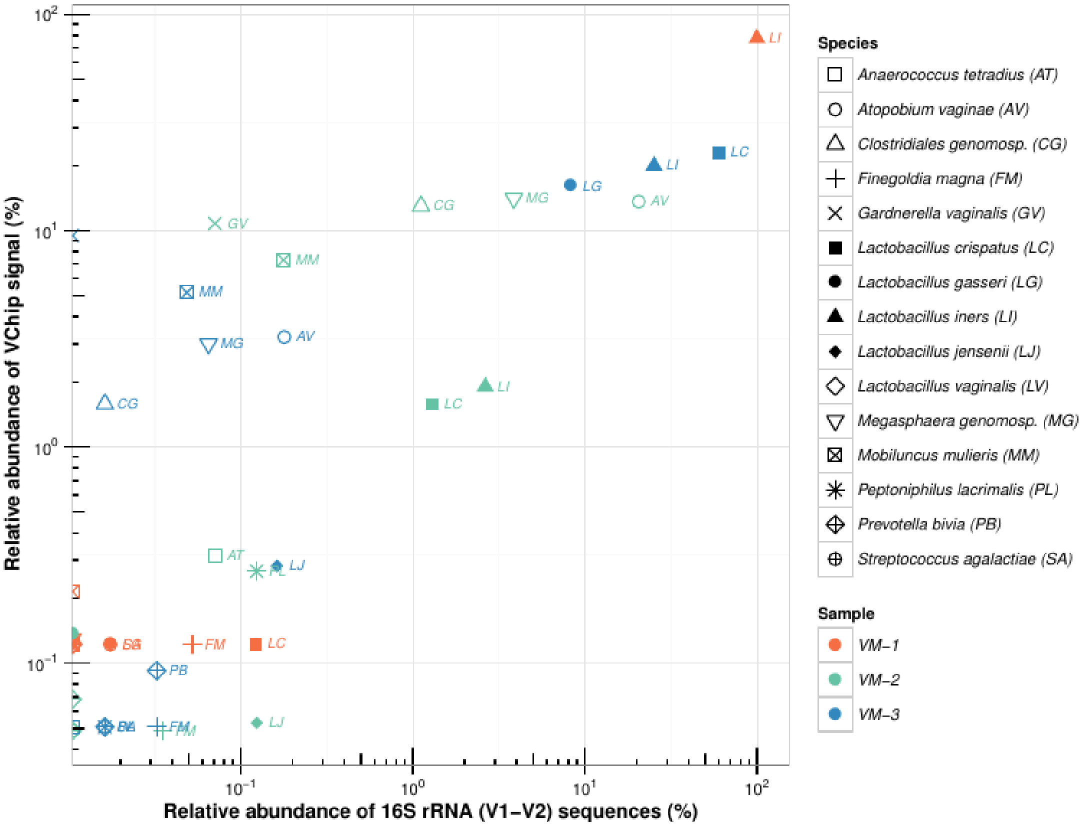
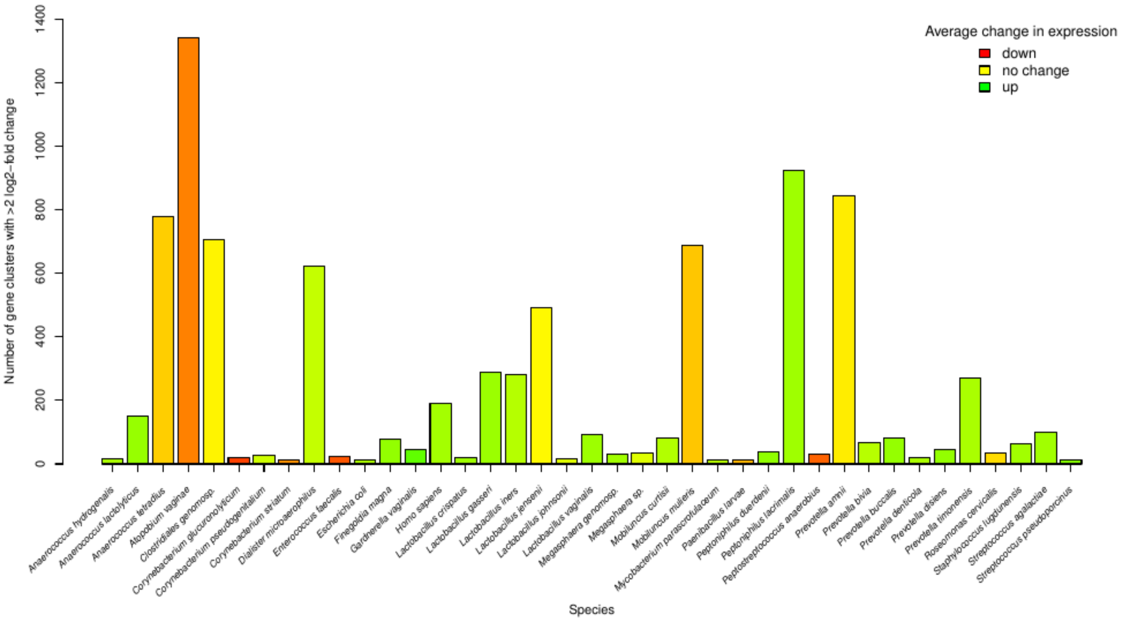


Figure 2.4. *Changes in gene expression in subject 2.*
The number of species-specific gene clusters with >2 log2-fold differences (in magnitude) in normalized cDNA hybridization signal between samples VM-2 and VM-3 are indicated on the y-axis. The coloring of each bar represents the average log2 fold-change difference in hybridization of species-specific probe sets in VM-3 relative to VM-2. The bars are colored on a continuous scale ranging from red (greatest average negative change) to yellow (zero average change) to green (greatest average positive change).



Tables

Table 2.1. *Composition of mock communities tested on the VChip microarray.* In order to compare VChip-derived community composition with expected proportions, proportions of genomic DNA were converted to proportions of expected genome copy equivalents based on individual species' genome sizes.

Mock community	**MC-1**	**MC-2**	**MC-3**	**MC-4**	**MC-5**	**MC-6**	**MC-7**	**MC-8**	**MC-9**
Species	Proportion of genomic DNA (proportion of expected genome copy equivalents[a])								
Anaerococcus hydrogenalis	0.200	0.100	0.010	0.010	0.001				
(vaginal swab isolate)	(0.183)	(0.096)	(0.010)	(0.096)	(0.018)	-	-	-	-
Anaerococcus tetradius	0.200	0.100	0.010	0.010	0.001				
(vaginal swab isolate)	(0.165)	(0.087)	(0.009)	(0.087)	(0.016)	-	-	-	-
Atopobium vaginae	0.200	0.100	0.010	0.010	0.001				
ATCC BAA-55	(0.246)	(0.130)	(0.014)	(0.129)	(0.024)	-	-	-	-
Finegoldia magna	0.200	0.100	0.010	0.010	0.001				
(vaginal swab isolate)	(0.184)	(0.097)	(0.010)	(0.096)	(0.018)	-	-	-	-
Gardnerella vaginalis	0.200	0.100	0.010	0.010	0.001				
ATCC 14018	(0.222)	(0.117)	(0.012)	(0.116)	(0.022)	-	-	-	-
Lactobacillus crispatus		0.500	0.950	0.050	0.050	0.050	0.010	1.000	
ATCC 33820	-	(0.473)	(0.945)	(0.470)	(0.889)	(0.987)	(0.937)	(1.000)	
Homo sapiens				0.900	0.945	0.950	0.990		1.000
(female genomic DNA)	-	-	-	(0.006)	(0.011)	(0.013)	(0.063)	-	(1.000)

Table 2.2. *Pearson correlation coefficients for VChip vs. in silico mapping of Illumina RNA-Seq reads against probe sequences.*

Genus (upper diag.) / Species (lower)	**VM-1 cDNA**	**VM-1 reads**	**VM-2 cDNA**	**VM-2 reads**
VM-1 cDNA	1.00	0.71	0.37	0.31
VM-1 reads	0.75	1.00	0.33	0.53
VM-2 cDNA	0.36	0.31	1.00	0.79
VM-2 reads	0.20	0.25	0.77	1.00

References

1. Larsen B, Monif GR (2001) Understanding the bacterial flora of the female genital tract. Clin Infect Dis 32:e69–77. doi: 10.1086/318710

2. Hickey RJ, Zhou X, Pierson JD, et al. (2012) Understanding vaginal microbiome complexity from an ecological perspective. Translational Research 160:267–282. doi: 10.1016/j.trsl.2012.02.008

3. Hyman RW, Fukushima M, Diamond L, et al. (2005) Microbes on the human vaginal epithelium. PNAS 102:7952–7957. doi: 10.1073/pnas.0503236102

4. Zhou X, Brown CJ, Abdo Z, et al. (2007) Differences in the composition of vaginal microbial communities found in healthy Caucasian and black women. ISME J 1:121–133. doi: 10.1038/ismej.2007.12

5. Zhou X, Hansmann MA, Davis CC, et al. (2010) The vaginal bacterial communities of Japanese women resemble those of women in other racial groups. FEMS Immunol Med Microbiol 58:169–181. doi: 10.1111/j.1574-695X.2009.00618.x

6. Ravel J, Gajer P, Abdo Z, et al. (2011) Vaginal microbiome of reproductive-age women. PNAS 108 Suppl 1:4680–4687. doi: 10.1073/pnas.1002611107

7. Fettweis JM, Serrano MG, Sheth NU, et al. (2012) Species-level classification of the vaginal microbiome. BMC Genomics 13:S17. doi: 10.1186/1471-2164-13-S8-S17

8. Gajer P, Brotman RM, Bai G, et al. (2012) Temporal dynamics of the human vaginal microbiota. Science Translational Medicine 4:132ra52. doi: 10.1126/scitranslmed.3003605

9. Ravel J, Brotman RM, Gajer P, et al. (2013) Daily temporal dynamics of vaginal microbiota before, during and after episodes of bacterial vaginosis. Microbiome 1:29. doi: 10.1186/2049-2618-1-19

10. Hugenholtz P, Goebel BM, Pace NR (1998) Impact of culture-independent studies on the emerging phylogenetic view of bacterial diversity. Journal of Bacteriology 180:4765–4774.

11. Dekio I (2005) Detection of potentially novel bacterial components of the human skin microbiota using culture-independent molecular profiling. J Med Microbiol 54:1231–1238. doi: 10.1099/jmm.0.46075-0

12. Bik EM, Eckburg PB, Gill SR, et al. (2006) Molecular analysis of the bacterial microbiota in the human stomach. PNAS 103:732–737. doi: 10.1073/pnas.0506655103

13. Turnbaugh PJ, Ley RE, Hamady M, et al. (2007) The Human Microbiome Project. Nature 449:804–810. doi: 10.1038/nature06244

14. Hill JE, Goh SH, Money DM, et al. (2005) Characterization of vaginal microflora of healthy, nonpregnant women by chaperonin-60 sequence-based methods. Am J Obstet Gynecol 193:682–692. doi: 10.1016/j.ajog.2005.02.094

15. Fredricks DN, Fiedler TL, Marrazzo JM (2005) Molecular identification of bacteria associated with bacterial vaginosis. N Engl J Med 353:1899–1911. doi: 10.1056/NEJMoa043802

16. Gilbert JA, Dupont CL (2011) Microbial metagenomics: beyond the genome. Annu Rev Marine Sci 3:347–371. doi: 10.1146/annurev-marine-120709-142811

17. Qin J, Li R, Raes J, et al. (2010) A human gut microbial gene catalogue established by metagenomic sequencing. Nature 464:59–65. doi: 10.1038/nature08821

18. De Filippo C, Ramazzotti M, Fontana P, Cavalieri D (2012) Bioinformatic approaches for functional annotation and pathway inference in metagenomics data. Briefings in Bioinformatics 13:696–710. doi: 10.1093/bib/bbs070

19. Irizarry RA, Hobbs B, Collin F, et al. (2003) Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics 4:249–264.

20. Wagner M, Smidt H, Loy A, Zhou J (2007) Unravelling microbial communities with DNA-microarrays: challenges and future directions. Microbial Ecology 53:498–506. doi: 10.1007/s00248-006-9197-7

21. Gentry TJ, Wickham GS, Schadt CW, et al. (2006) Microarray applications in microbial ecology research. Microbial Ecology 52:159–175. doi: 10.1007/s00248-006-9072-6

22. DeAngelis KM, Wu CH, Beller HR, et al (2011) PCR amplification-independent methods for detection of microbial communities by the high-density microarray PhyloChip. Appl Environ Microbiol 77:6313–6322. doi: 10.1128/AEM.05262-11

23. Tottey W, Denonfoux J, Jaziri F, et al. (2013) The Human Gut Chip “HuGChip,” an explorative phylogenetic microarray for determining gut microbiome diversity at family level. PLoS ONE 8:e62544. doi: 10.1371/journal.pone.0062544.s004

24. Ballarini A, Segata N, Huttenhower C, Jousson O (2013) Simultaneous quantification of multiple bacteria by the BactoChip microarray designed to target species-specific marker genes. PLoS ONE 8:e55764. doi: 10.1371/journal.pone.0055764.s003

25. Rajilić-Stojanović M, Heilig HGHJ, Molenaar D, et al. (2009) Development and application of the human intestinal tract chip, a phylogenetic microarray: analysis of universally conserved phylotypes in the abundant microbiota of young and elderly adults. Environ Microbiol 11:1736–1751. doi: 10.1111/j.1462-2920.2009.01900.x

26. He Z, Gentry TJ, Schadt CW, et al. (2007) GeoChip: a comprehensive microarray for investigating biogeochemical, ecological and environmental processes. ISME J 1:67–77. doi: 10.1038/ismej.2007.2

27. Zhou A, He Z, Qin Y, et al. (2013) StressChip as a high-throughput tool for assessing microbial community responses to environmental stresses. Environ Sci Technol 47:9841-9849. doi: 10.1021/es4018656

28. Lee YJ, Van Nostrand JD, Tu Q, et al. (2013) The PathoChip, a functional gene array for assessing pathogenic properties of diverse microbial communities. ISME J 7:1974–1984. doi: 10.1038/ismej.2013.88

29. Paliy O, Agans R (2012) Application of phylogenetic microarrays to interrogation of human microbiota. FEMS Microbiology Ecology 79:2-11. doi: 10.1111/j.1574-6941.2011.01222.x

30. Zhou J (2003) Microarrays for bacterial detection and microbial community analysis. Curr Opin Microbiol 6:288–294. doi: 10.1016/S1369-5274(03)00052-3

31. Li W, Godzik A (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22:1658–1659. doi: 10.1093/bioinformatics/btl158

32. Edgar RC (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res 32:1792–1797. doi: 10.1093/nar/gkh340

33. Forney LJ, Gajer P, Williams CJ, et al. (2010) Comparison of self-collected and physician-collected vaginal swabs for microbiome analysis. Journal of Clinical Microbiology 48:1741–1748. doi: 10.1128/JCM.01710-09

34. Yuan S, Cohen DB, Ravel J, et al. (2012) Evaluation of methods for the extraction and purification of DNA from the human microbiome. PLoS ONE 7:e33865. doi: 10.1371/journal.pone.0033865.t004

35. Roche NimbleGen (2011) NimbleGen Arrays User's Guide: CGH and CNV Arrays (version 8.1). 1–77.

36. R Core Team (2012) R: A Language and Environment for Statistical Computing. Vienna, Austria. http://www.R-project.org/

37. Carvalho BS, Irizarry RA (2010) A framework for oligonucleotide microarray preprocessing. Bioinformatics 26:2363–2367. doi: 10.1093/bioinformatics/btq431

38. Falcon S, Carvalho B with contributions from Carey V, Settles M, de Beuf K (2009) pdInfoBuilder: platform design information package builder. R package version 1.26.0.

39. Smyth GK (2005) Limma: linear models for microarray data. In: Gentleman R, Carey V, Dudoit S, et al. (eds) Bioinformatics and Computational Biology Solutions Using R and Bioconductor. Springer, New York, pp 397–420.

40. Storey JD (2002) A direct approach to false discovery rates. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64:479–498. doi: 10.1111/1467-9868.00346

41. Dabney A, Storey JD with assistance from Warnes GR (2004) qvalue: Q-value estimation for false discovery rate control. R package version 1.36.0.

42. Pages H, Aboyoun R, Gentleman R, DebRoy S (2013) Biostrings: string objects representing biological sequences and matching algorithms. R package version 2.30.1.

43. Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10:R25. doi: 10.1186/gb-2009-10-3-r25

44. Roche NimbleGen (2008). NimbleScan Software User’s Guide (version 2.5). 1–132.

45. Kang S, Denman SE, Morrison M, et al. (2010) Dysbiosis of fecal microbiota in Crohnʼs disease patients as revealed by a custom phylogenetic microarray. Inflammatory Bowel Diseases 16:2034–2042. doi: 10.1002/ibd.21319

46. Preza D, Olsen I, Willumsen T, et al. (2008) Microarray analysis of the microflora of root caries in elderly. Eur J Clin Microbiol Infect Dis 28:509–517. doi: 10.1007/s10096-008-0662-8

47. Crielaard W, Zaura E, Schuller AA, et al. (2011) Exploring the oral microbiota of children at various developmental stages of their dentition in the relation to their oral health. BMC Med Genomics 4:22. doi: 10.1186/1755-8794-4-22

48. Dols JAM, Smit PW, Kort R, et al. (2011) Microarray-based identification of clinically relevant vaginal bacteria in relation to bacterial vaginosis. Am J Obstet Gynecol 204:1.e1–1.e7. doi: 10.1016/j.ajog.2010.11.012

Chapter 3

ARC: Assembly by Reduced Complexity

Abstract

High throughput sequencing (HTS) technologies produce millions of short reads. Analysis of these reads is a difficult problem, especially in the context of non-model organisms where comparison of homologous sequences may be frustrated by the lack of a reference genome. Current read mapping based methods rely on the availability of a highly similar reference sequence while *de novo* assembly may be slow or intractable for large datasets. To partly overcome these problems I developed Assembly by Reduced Complexity (ARC), a software package for targeted assembly of homologous sequences. The algorithm consists of three steps. 1) Reads are mapped against a set of targets; 2) reads are then split into subsets based on the mapping results, and 3) assemblies are carried out for each target. This process is iterated using the newly assembled set of reads as mapping targets for the next iteration. ARC is implemented in Python and supports the Bowtie2 and BLAT mappers as well as the Roche/Newbler and Spades assemblers. We show that ARC works effectively with divergent references, functions well with short ancient DNA reads, and compares favorably to *de novo* assembly in CPU and memory requirements.

Introduction

High-throughput sequencing (HTS) techniques have become a standard method for producing genomic and transcriptomic knowledge about an organism [1]. Most currently available high-throughput sequencing platforms produce millions of short sequences referred to “reads” that range in length from 50 to 700 base pairs (bp) depending on the sequencing chemistry and platform. These short reads are typically produced at random from the much larger genome making them effectively meaningless without further analysis. Thus, the primary challenge in the analysis of HTS data is to organize and summarized the massive number of short sequences into a simpler form that provides insight into the underlying biology. Two analysis strategies, *de novo* sequence assembly, and sequence mapping have been widely adopted to achieve this end.

The objective of *de novo* assembly is to piece together short read sequences to form longer sequences known as contigs. Sequence assembly is a difficult problem, that is made more difficult by repeated elements in the genome, heterozygosity, short reads, and sequence read errors [2]. Additionally, assembly algorithms are computationally intensive for all but the smallest datasets, thus limiting their application [3]. Finally, *de novo* assembly of large datasets typically produces multiple contigs that can require significant additional organization and analysis. Despite many advances and a large selection of assembly software packages, fragmentation and missassembly are common problems and improving the quality of *de novo* sequence assemblies continues to be an area of active research [4].

Sequence mapping is often the first step carried out in resequencing projects where a good reference sequence exists. The objective of mapping is to properly align short reads against the much longer reference sequence, thereby permitting comparisons between a sequenced sample and the reference. This approach is much faster than sequence assembly and it has proven to be very effective at identifying variants at a large scale [5]. Unfortunately successful read mapping is entirely dependent on a reference sequence that is very similar at all loci to the reads being mapped. Differences between a sample and reference sequence such as structural variations (SVs), novel sequences, an incomplete or misassembled reference, or sequence divergence all result in unmapped or poorly mapped reads. Poor quality mapping can result in false variant calls [6] and in the context of RNA-seq experiments unmapped reads result in counting errors that make detection of differentially expressed genes more error prone [7]. In short, resequencing projects are done to identify differences between a sample and an established reference, however the regions that are most divergent are also the most difficult to map reads against. Because of this, mapping based approaches are inherently biased by the reference used and only provide reliable results when divergence is below the threshold at which reads can be mapped accurately.

The two approaches described above (mapping and *de novo* assembly) have primarily been developed and optimized for whole-genome analysis, however another class of problems exists in which specific regions of a genome or subsets of the sequenced DNA are analyzed. This type of analysis is appropriate in many instances, including sequence capture, viral genome assembly from environmental samples, RNA-seq, mitochondrial or cholorplast genome assembly, metagenomics, and many more. In cases like these, it is often necessary to develop custom pipelines to carry out analysis as in the work of Gilbert et al. [8].

In an attempt to address these issues we introduce a hybrid strategy, Assembly by Reduced Complexity (ARC) that combines the strengths of mapping and *de novo* assembly approaches while minimizing their weaknesses. This approach is designed for the myriad of situations in which the assembly of entire genomes or datasets is not the primary objective, but instead the assembly of several discreet, relatively small, targets such as mitochondrial genome sequences is required. ARC is an iterative algorithm that uses an initial set of reference sequences (targets) to seed *de novo* assemblies. Reads are first mapped against these targets, and then the mapped reads are pooled and assembled in parallel on a per-target basis to form contigs. These assembled contigs then serve as reference sequences for the next iteration (see Figure 1). This method breaks the assembly problem down into multiple, small problems, thereby addressing the poor scaling issue inherent in some *de novo* assembly methods caused by large numbers of reads. It also reduces the reference bias inherent in mapping by including a *de novo* assembly component.

Methods

The ARC algorithm proceeds through a number of stages described below.

Initialization

During the initialization stage a configuration file is processed and a number of checks are carried out to ensure that data and executables specified in the configuration file are available. If any of these checks fail, ARC will report an informative error message providing details about the problem and then exit. If all checks pass successfully the initialization process continues to the next stage in which internal data structures are created to store information about the experiment and pipeline progress. Working directories and index files are also created for each sample, and names that are file-system safe are assigned to each target sequence. Finally the job queue, workers, and job management system is started, and mapping jobs are added to the job queue for each sample. With setup complete, ARC begins the iterative part of the pipeline. This algorithm consists of four steps: mapping, splitting, assembling, and finishing. This work flow is diagrammed in Figure 1 and the ARC process of assembling reads is illustrated in Figure 2.

Mapping: reads are recruited by mapping against a set of reference targets

In the first step, ARC recruits reads by mapping them against a set of reference targets using one of the two supported mappers, BLAT [9] or Bowtie 2 [10] as specified in the configuration file. During all further iterations, the mapping reference consists of contigs assembled from previously recruited reads that must be highly similar to newly recruited reads.

BLAT is a fast, seed-and-extend sequence alignment tool that supports gapped alignments and has proven effective at recruiting reads even in cases where global sequence identity is as low as 70%. In the first iteration, BLAT is run using default parameters (minIdentity=90, minScore=30) but on all subsequent iterations mapping stringency is increased (minIdentity=98, minScore=40) to reduce recruitment of less similar reads. BLAT reports all alignments that meet the minimum score criteria, so it is possible to use the same read multiple times if it aligns successfully against more than one target. One serious drawback of using BLAT is that it does not support the fastq format. All current sequencing platforms produce quality information for reads and this information is typically encoded in fastq format, making this limitation of BLAT a significant problem for HTS data. To facilitate usage of ARC with fastq format data we include a code patch for BLAT that adds support for fastq. Instructions for applying this patch can be found in the supplementary material.

Bowtie 2 is another fast, gapped, read aligner that was specifically designed for mapping HTS reads [10]. Bowtie 2 is run in local alignment mode (--local option) which enables the recruitment of reads that partially map at the end of contigs and in low-homology regions. Additionally, the option to report up to five valid alignments (-k 5) is used by default. This setting can be modified based on the user's expectations by setting the bowtie2_k parameter in the ARC configuration file. Setting bowtie2_k=1 will cause Bowtie 2 to run in default mode where only the best alignment found is reported.

Splitting: reads are split into subsets based on mapping results

In the second step ARC splits reads into subsets using mapping results. The supported mappers, BLAT and Bowtie 2 generate PSL or SAM [11] formatted output files, respectively. The output file is processed by ARC and reads are then split by mapping target. This is accomplished by creating a set of output files corresponding to each target. Reads which mapped to that target are then written to their respective files, making it possible to process each set of reads independently from the others. Splitting requires fast random access to the read files, and this is facilitated by storing read offset values in a SQLite database as implemented in the Biopython SeqIO module [12]. Two special considerations are taken into account during splitting. First, since the Newbler assembler has no mechanism to indicated paired reads it is necessary to reformat the read identifier to ensure compatibility with Newbler paired-end detection. This is done by ensuring that the read identifier is made up of five fields separated by a colon, and ending in a sixth field indicating the pair number. Identifiers for single end reads are similarly reformatted, except that the sixth field which indicates pair number is left blank. Secondly, regardless of whether one or both ends of paired reads map to a target, both members of the pair are recruited as long as at least one of them was mapped. Recruiting paired reads in this way takes advantage of the information stored in paired reads, and allows for faster extension of targets.

Despite using a fast strategy for randomly accessing the read files splitting is limited by system input/output latency and to using a single CPU core per sample. To optimize CPU use on modern multi-core systems, ARC immediately adds an assembly job to the job queue as soon as all reads for a target have been split. This allows assemblies to proceed concurrently with the read splitting process.

Assembling: subsets of reads are assembled using either the Spades or Newbler assemblers

Because the read splitting process is carried out sequentially across mapping targets, an assembly job can be launched as soon as all reads for a target have been written. As soon as resources are available, the assembly job is started, allowing ARC to run the read splitting and assembly processes concurrently. Two assemblers are supported the Roche GS d*e novo* Assembler (also known as Newbler) [13], and SPAdes [14]. Assemblies within ARC are always run with a timeout in order to gracefully handle the rare cases where the assembler crashes, does not exit properly, or takes longer than expected to run. This allows ARC to continue running efficiently on large projects where a small number of targets might otherwise be problematic (i.e. due to recruiting reads from repetitive elements). The timeout value can be controlled using the assemblytimeout setting in the configuration file.

Newbler was originally designed to assemble reads generated by the 454 pyrosequencing platform [13] but recent versions have added support for Illumina PE reads. ARC supports two Newbler specific parameters that can sometimes improve performance. These can be set using urt=True which instructs Newbler to "use read tips" in assemblies, and rip=True, which instructs Newbler to place reads in only one contig. We have found that setting urt=True can reduce the number of ARC iterations necessary to assemble a target.

The SPAdes [14] assembler is also supported by ARC. SPAdes is an easy to use De Bruijn graph assembler that performed quite well in a recent evaluation of bacterial genome assemblers [15]. SPAdes tends to perform well in the ARC pipeline, but surprisingly is not as fast as Newbler for small read sets. This may partly be because SPAdes implements a number of steps in an attempt at improving the often fragmented De Bruijn graph assembly results. These steps include read error correction, multiple assemblies using different k-mer sizes, and merging of these assemblies. In ARC, SPAdes is run using the default parameters.

In some cases the available reference targets may be so divergent from the sequenced specimen that only a small number of reads can be recruited on the first iteration. If too few reads are recruited, the assemblers have very little data to work with, and, especially in the case of SPAdes, often fail to assemble contigs. In an attempt to address this situation, we provide a final pseudo-assembly option that skips assembly on the first iteration, and instead treats the recruited reads as contigs. These reads are then used as mapping targets in the second iteration. This option can be enabled by setting map_against_reads=True in the configuration file. Although we have had some success with this strategy, its success varies across datasets and targets. In some cases using reads as mapping targets results in recruiting large numbers of reads from repeat regions, causing the assembly to fail. For this reason we only recommend using this approach after testing ARC with normal settings.

Finishing: assembled contigs are written as a new set of mapping targets or to finished output

Once all assemblies are completed for a given sample, the final step in the ARC pipeline is initiated. During this step each target is evaluated and if stopping conditions are met then the contigs are written to the final output file and if not the contigs are written to a temporary file where they are used to recruit reads in the next iteration (see the following section *Folder structure: outputs and logging for details*). Stopping conditions are defined as followed: 1) Detection that an assembly was killed will result in no further attempts at assembling this target, instead any contigs produced on the previous iteration will be written to the output file. 2) If no additional reads have been recruited since the previous iteration then no further extension of the target is possible. 3) Occasionally a target will be flanked by repeated sequence in the genome that can cause a sudden spike in the number of recruited reads. The max_incorporation parameter controls sensitivity to this situation and by default it will be triggered if five times the previous number of reads are incorporated.

During output, contig identifiers are modified to reflect the sample, target, and contig number. Contigs are also masked for simple tandem repeats using an approach that relies on frequency of trinucleotides in a sliding window. The repeats are masked by setting them to lower case, or by modifying the repeat sequence to the letter 'N' depending on whether the BLAT or Bowtie 2 mapper is used. Additionally, for all contigs that are written to the final output file, all reads that were mapped in the final iteration are also written, however their description field is modified to reflect which target they belong to.

If, after all targets have been processed, any remain unfinished, the processes is reiterated using the newly assembled contigs as mapping targets.

Description of input files

Inputs to ARC consist of three types of files: a) a file containing reference sequences (referred to here as “targets”), b) files containing reads for each sample, and c) a configuration file.

a) The targets file contains each of the distinct sequences that are used as mapping references in the first iteration of ARC. These sequences must be in standard fasta format and should have informative, unique names. It is possible to use multiple sequences as a single target in cases where a number of homologous targets are available and it is not clear which of them is the most similar to the sequenced sample (e.g. in the case of ancient DNA extracted from unidentified bone material). This can be accomplished by naming the targets using ARC's internal naming scheme in which each contig is labeled using an identifier made of three parts separated by "_:_" (e.g. *P1_:_P2_:_P3)*. During read splitting, ARC will treat all target sequences which have an identical value in P2 as a single target.

b) Each biological sample can be represented by at most three read containing files; two paired end (PE) files, and one single end (SE) file. ARC will function with one SE file, a PE set of files, or all three files. If multiple sets of reads are available for a single biological sample (i.e. from different sequencing runs or technologies) they should be combined into at most three files. All reads for all samples must be in the same format (fasta or fastq) and this format should be indicated by the format parameter in the ARC configuration file. It is highly recommended that reads be cleaned to remove adapter sequences and low quality bases prior to analysis. Removing duplicate reads and trimming low-quality regions has also been observed to produce higher quality, less fragmented ARC assemblies, especially with capture data (data not shown).

c) The final input to ARC is the configuration file. This plain text file describes the data and sets various parameters that ARC will use during assembly, mapping, and output stages. By default the configuration file should be named ARC_config.txt, but any name can be used as long as the *-c filename* switch is passed to ARC. The configuration file is split into three types of entries, denoted by the first characters in the line. Lines starting with the characters "##" are treated as comments and ignored, lines starting with "#" are used to set parameters, and lines that don't begin with "#" indicate files belonging to samples. The one exception to this rule is the column header line, the first line which doesn't begin with "#", and contains column names. This line is ignored by ARC, but is expected in the configuration file. An example of an ARC configuration file is included in the "test_data" folder that comes with ARC and an example listing of this file is included in the Supplementary Material. A comprehensive list of configuration options are presented in Table S1 of the Supplementary Material.

Folder Structure Outputs and Logging

To minimize memory usage and interact with assembler and mapper programs, ARC relies heavily on temporary files. These files are organized into subdirectories under the path from which ARC is launched. During ARC processing a pair of folders is created for each sample. These folders have the prefixes “working_” and “finished_”. Temporary files used during ARC processing are stored in the “working_” folders while results and statistics are recorded to the “finished_” folders.

Although the “working_” folders contain temporary files and can be safely deleted after an ARC run, they contain information that can be useful in some cases, especially for debugging. In particular, the contigs assembled during each iteration are stored in a set of files with file names “*I00N_contigs.fasta*” (where “N” corresponds to the iteration). Also potentially useful are the “*t__0000N*” directories (where “N” corresponds to the numeric index of the target) that contain the final assembly log and files generated by the assembler. These files can be informative in determining why an assembly failed or for examining assembly statistics of a particular target in more depth. Additionally, these folders provide the option of manually re-running an assembly with a different set of parameters than those chosen for ARC.

The “finished_” folders contain the following files: *contigs.fasta, mapping_stats.tsv, PE1.fasta, PE2.fasta, SE.fastq*. The *contigs.fasta* file stores the final set of assembled contigs for each target. Contigs are named according to the three part naming scheme previously described, *sample_*:*_target_*:*_contig*, to facilitate easy comparisons between samples. The *mapping_stats.tsv* file is a tab-separated values file that stores information on the number of reads mapped to each target at each iteration. This file can be easily loaded into a spreadsheet, or statistical program such as R to generate plots or do other analysis. The final set of files, *PE1.fasta, PE2.fasta,* and *SE.fasta* contain all reads that were mapped on the final ARC iteration. If only pair-end or single-end files were provided then only reads of this type will be reported. These files will be formatted in the same way as the input files (fasta or fastq) and have modified description fields to indicate the sample and target to which they were mapped.

Datasets used for testing

ARC was tested using two datasets. The first dataset is made up of sequenced reads from two different exon capture experiments using samples collected from chipmunks (*Tamias sp.)*. This combined dataset consists represents 55 specimens, 3 of which were sequenced as part of [16] while the other 52 were sequenced as part of a separate study (Sarver et al. in prep). The second dataset consists of sequenced reads from a whole-genome shotgun sequencing experiment using ancient DNA extracted from a mammoth hair shaft sample.

The chipmunk dataset was used to investigate ARC's sensitivity to divergent references as well as its utility and performance with large datasets. For all 55 specimens, libraries were captured using an Agilent SureSelect custom 1M-feature microarray capture platform that contains 13,000 capture probes representing the mitochondrial genome and 9716 genes (designed by [16]). Libraries were then sequenced on the Illumina HiSeq 2000 platform (100bp paired end). The 55 chipmunks represent seven different species within the genus *Tamias* with representatives of *T. canipes*: 5, *T. cinereicollis*: 9, *T. dorsalis*: 12, *T. quadrivittatus*: 1, *T. rufus*: 5, and *T. umbrinus*: 10, collected and sequenced as part of (Sarver et al. in prep) and *T. striatus*: 3 collected and sequenced by [16]. Reads from the *T. striatus* samples were obtained from Dr. Jeffrey M. Good (personal communication).

Prior to ARC analysis, reads were preprocessed through a read cleaning pipeline consisting of the following steps. In the first step, PCR duplicates (duplicate reads resulting from multi-cycle PCR reactions carried out as part of library preparation) were removed using a custom Python script. Sequences were then cleaned to remove sequencing adapters and low quality bases using the software package Seqyclean (Zhbannikov et al. manuscript in prep, https://bitbucket.org/izhbannikov/seqyclean). Finally, because paired-end sequencing produces two reads sequenced from either end of a single template, it is often possible to overlap these reads to form a single long read representing the template in its entirety. This overlapping was carried out using the Flash software package [17]. Post-cleaning, the dataset consisted of 21.9 Gbp (giga base pairs) in

194,597,935 reads.

ARC analysis for the first dataset was carried out using two different sets of references. To determine how well ARC could perform with divergent references, the mitochondrial genome of each specimen was assembled against eleven different mitochondrial references (see Table 1). We also tested ARC's performance with a large number of targets by using a target set consisting of a manually assembled *Tamias canipes* mitochondrial sequence plus 11,976 exon sequences making up 7,627 genes. These sequences represent a subset of the 9716 genes that the capture probes were originally designed against.

A second dataset was used to test ARC's performance on short, poor quality reads that are typical of ancient DNA sequencing projects. Total DNA was extracted from ancient hair shafts and reads were sequenced on the Roche 454 platform by [8]. Although these reads represent shotgun sequencing of the nuclear and mitochondrial genomes, the authors report a high concentration of mitochondria in hair shaft samples resulting in high levels of mitochondrial DNA as compared to nuclear DNA. Sequenced reads for *Mammuthus primigenius* specimen M1 were obtained from the Short Read Archive using accession SRX001889 (http://www.ncbi.nlm.nih.gov/sra/?term=SRX001889) and cleaned with SeqyClean (Zhbannikov et al. manuscript in prep, https://bitbucket.org/izhbannikov/seqyclean) to remove sequencing adapters and low quality bases. Following cleaning, this dataset contains a total of 19 Mbp (Mega base pairs) in 221,688 reads with an average length of 86.2 bp. Although these reads were sequenced on the Roche 454 platform which typically produces much longer reads (500-700bp), 75% of cleaned reads were 101 bp or less in length making them extremely short for this platform. ARC analysis was carried out using three mitochondrial references, the published *Mammuthus primigenius* sequence from another specimen, M13, Asian elephant (*Elephas maximus*) the closest extant relative of the mammoth [18], and a divergent reference, mouse (*Mus musculus*) (accessions: EU153445, AJ428946, NC_005089 respectively).

Results

ARC is open source software implemented in the Python programming language with source code available for download from GitHub (https://github.com/ibest/ARC). ARC can be installed on most Linux servers, but will also work on many desktops or laptops provided that Linux and other requirements are installed. The installed size is only 3 Mb and system administrator access is not required making it easy to download and test. Configuration is done via a plain text file that can be distributed to make replication of results simple. Prerequisite software (Python 2.7.x, Biopython [12], BLAT [9] or Bowtie 2 [10] and Newbler [13] or SPAdes [14]) is easy to obtain, and may already be available on systems previously used for HTS analysis.

ARC was tested using the two datasets described in the Methods section. Tests were done to determine how well ARC performs when a divergent reference was used, whether it was effective in assembling sequences from short, poor quality reads produced from ancient DNA, and to measure its performance on a large dataset. The results of these tests are presented below.

ARC performs well even with divergent references

A divergent reference sequence can result in unmapped and poorly aligned reads when using mapping based approaches [3]. To test how robust ARC was to divergent reference sequences we assembled mitochondrial genomes using reads from sequence capture experiments performed on 55 chimpmunk specimens representing seven different species within the *Tamias* genus (*T. canipes, T. cinereicollis, T. dorsalis, T. quadrivittatus*, *T. rufus*, *T. umbrinus*, and *T. striatus*). Assembly was done using a set of mitochondrial references spanning mammalia with differences in percent identity ranging from 71% to 94.9% with respect to *Tamias cinereicollis* (see Table 2).

Reference bias in the ARC result was assessed by considering how similar the set of recruited reads was across targets (similar is defined here as the difference between recruited reads and median recruited reads being less than 100 reads). Detailed read counts are presented in Table S2 and summarized in Table 1.

For the majority of specimens (32 of 55) all targets recruited a similar number of reads. In 21 of the 55 cases, most targets recruited a similar number of reads, but one or more recruited a different number. In six of these cases, the Tasmanian devil reference caused incorporation of a large number of reads leading to assembly timeout, however in each of these cases the same core set of reads was incorporated even by the Tasmanian devil reference. Finally, in two specimens (S10 and S228) a large number of reads were recruited, causing ARC to terminate assemblies for all targets. Through further analysis of the contigs produced midway we found that these two samples contained an abundance of mitochondrial reads, with coverage depths >2000x in some cases.

To further characterize ARC's performance on a divergent set of references, we selected sample S152 (a *Tamias cinereicollis* specimen) for a more detailed analysis. As shown in Table 1, the number of ARC iterations required to complete the assembly of the mitochondrial genome for this specimen ranged from 3 to 16. This number is negatively correlated with the percent identity between the targets and the sequence for this specimen (Pearson correlation = -0.742). The relationship between target and read recruitment is further illustrated in Figure 3 which shows that the most similar target, the Gray-footed chipmunk (*T. canipes*), recruited almost the full set of reads (98.3%) in the first iteration and finished on the third iteration. At the other extreme, Platypus recruited a mere 6% of reads (2,305 of 38,388) on the first iteration, but after 15 iterations acquires the full set of reads.

ARC assembled a single contig for all 11 targets, however the contig lengths differed slightly between two groups of targets: Gray-footed chipmunk, Red squirrel, and House mouse targets produced identical 16,726 bp contigs, all others produced identical 16,730 bp contigs. A combination of pairwise alignments and dot-plots (data not shown) indicate that these differences are due to the way in which this circular sequence was linearized. The 16,726 bp contig has a 178 bp overlap between the beginning and end of the contig, while the 16,730 bp contig has a 182 bp overlap.

ARC assembles large contigs from short, poor quality reads produced from ancient DNA

Methods that permit investigators to extract DNA from samples that are as much as 50,000 years old and prepare libraries for HTS have been developed [8, 18, 19]. The DNA from these ancient samples tends to be partially degraded resulting in short, poor quality reads [19]. As illustrated in Figure 1, ARC relies on an iterative process to extend assemblies into gaps. These gaps are filled by recruiting reads with partial, overhanging alignments at the edge of a contig. To test whether ARC can be used effectively with short, single-end reads produced from ancient samples, we attempted to assemble the mammoth (Mammuthus primigenius) mitochondrial genome using reads sequenced by Gilbert et al. [8] from DNA collected from hair samples.

Sequenced reads were obtained for *Mammuthus primigenius* specimen M1 from the Short Read Archive (accession: SRX001889) and processed as explained in the Methods section. ARC analysis was done using three mitochondrial references, the published sequence from *Mammuthus primigenius* specimen M13, Asian elephant (*Elephas maximus*) the closest extant relative of the mammoth [18], and a more divergent reference, mouse (*Mus musculus*) (accessions: EU153445, AJ428946, NC_005089 respectively).

ARC results were assessed by alignment against the published *Mammuthus primigenius* M1 sequence (EU153444) that is 16,458bp in length. Results of this comparison are presented in Table 3. Percent coverage (> 99%) and identity (> 98%) was high for the mammoth and elephant references. The mouse reference resulted in a slightly smaller assembly (total length 15,781bp), however coverage (95.9%) and identity (99.4%) were still very good. Surprisingly, the mouse reference required 78 ARC iterations to build this final set of contigs, recruiting only 223 reads on the first iteration. Despite starting from such a small number of reads, by the 78th iteration a total of 4507 reads had been recruited, almost the same number as the other much less divergent references.

All assembled contigs could be aligned to the published reference sequence, however we noted that the assembled lengths (16,620 and 16,603 bp for mammoth and elephant respectively) were longer than the published sequence length of 16,458 bp. Also intriguing were two contigs from both the mammoth and elephant references that were forced to overlap when aligned against the publish M1 sequence. This overlapping region showed much lower percent identity in the alignment than the rest of the aligned contig. To investigate whether this was due to a poor quality assembly on the part of ARC, or an error in the published sequence, we aligned the ARC contigs produced from the mammoth reference against the published Asian elephant sequence (Figure 4). This alignment showed that a number of gaps existed in the ARC assembly as compared to the published contigs. Each of these gaps was associated with a homopolymer (consecutive identical bases, i.e. AAA), that are known to cause errors with pyrosequencing technology. More interesting was that the D-loop region of the published *Mammuthus primigenius* M1 sequence contains 10 'N' characters (indicated by an “N” annotation in the figure) followed by a 370bp gap when aligned against the Asian elephant reference. ARC assembles 220bp of this sequence, including sequence that crosses the unknown, “N” bases in the published sequence. These assembled bases align with high identity against the Asian elephant reference suggesting that they represent an accurate assembly of this locus and that the published M1 mitochondrial sequence may be missing sequence or be misassembled in this region.

ARC computational requirements for large datasets

To be useful for modern genomic experiments ARC must be able to process large datasets with multiple samples and thousands of targets. We benchmarked ARC's performance with the previously described chipmunk dataset that contains reads from 55 specimens representing sequence capture of 9,716 genes as well as the full mitochondrial genome. After stringent read cleaning to remove adapters, PCR duplicates, and overlapping paired end reads with short inserts, this dataset contained 21.9 Gbp in 194,597,935 reads. For comparison purposes we also did *de novo* assemblies of three libraries with Newbler v2.6 (Table 4). ARC required 77 hours 45 minutes to process this dataset, carrying out 1.3 million assemblies in total and using a maximum of 31.19 GB of memory. On average this equates to 1 hour 25 minutes per sample. Individual assemblies for the three specimens were variable, requiring between 6.71 GB and 17.54 GB of memory, with running times of between 31 minutes and 13 hours 27 minutes to complete. Although time and memory requirements are smaller for each individual assembly, the total requirement for 55 samples would most likely be much higher than the time required by ARC to process all samples. Additionally, the set of contigs resulting from each individual assembly have no annotation, requiring significant additional analysis before homologous sequences could be compared between samples. In contrast the results from ARC are annotated by target, making comparisons an easy next step (see Methods).

Since ARC breaks the assembly problem down into small pieces, we postulated that memory requirements would scale as a function of the number of CPUs used to do ARC assemblies rather than as a function of the number of total reads as is normally the case with sequence assembly [2]. To test this we performed nine ARC runs using between 10 and 50 CPU cores with the 55 specimen chipmunk dataset. Instead of the full set of targets we used a subset of 200 when running this experiment so that it could be completed in a reasonable amount of time. During each assembly we recorded maximum memory usage. The results showed a linear increase in memory usage as the number of cores is increased. A linear model was fit to this data resulting in an estimated slope of 0.07 GB per CPU core ($P < .005$, $R^2 = 0.96$). It is important to note that even though this dataset contained 21.9 Gbp of reads, analysis using a small number of CPU cores and a reduced dataset required less than 3 GB of RAM, making it possible to use ARC to analyze large datasets on a modern desktop computer.

Discussion

Here we have introduced ARC, a software package that facilitates targeted assembly of HTS data. This software is designed for use in situations where assembly of several discreet and relatively small targets is required and (potentially divergent) homologous reference sequences are available for seeding these assemblies. ARC fills the gap between fast, mapping based strategies that can fail to map reads properly at divergent loci, and *de novo* assembly strategies that can be slow, resource intensive, and require significant additional analysis after assembly is completed. ARC was evaluated in three ways: 1) determine whether ARC results were biased by divergence of the reference 2) effectiveness of ARC on short, low quality reads 3) characterize performance on a typical HTS dataset with thousands of targets.

Assemblies using a divergent set of references with chipmunk specimens, show that ARC does not require a close reference to produce high quality final contigs. Figure 2 illustrates that on the initial iteration, ARC was able to map only a tiny fraction of the mitochondrial reads to all but the closely related Gray-footed chipmunk reference, yet was able to recover the full set of reads after 15 iterations with the platypus reference. This small set of reads represents the total number of reads that would have been recruited in traditional mapping and illustrates how sensitive read mapping is to high levels of divergence. A similar pattern emerged when we used a mouse reference to seed assembly of a mammoth mitochondrial genome. A mere 223 reads mapped on the first iteration, but this was sufficient to seed assembly of an almost full-length sequence composed of 4507 reads. In all cases where assemblies were completed the resulting set of reads and contigs were identical or nearly so, providing strong evidence that ARC was able to assemble high quality, unbiased contigs using even very divergent references to seed initial read recruitment. This capability makes ARC a very useful tool when analyzing sequence data from non-model organisms or when the identity of a sample is in question.

Repetitive sequences and large numbers of reads increase memory usage and slow assembly [2]. Although ARC addresses this problem by breaking the full set of reads up into small subsets before assembly it can still encounter this problem with very high coverage libraries or when a target with repetitive sequences recruits a large numbers of reads. For example, when testing ARC's ability to handle diverse mitochondrial references, assembly could not be completed for two specimens, S10 and S228. In the case of the S10 specimen the depth of sequence coverage was ~1500x for the mitochondrial genome. This depth is not suited for the Newbler assembler which performs pairwise comparisons of every read and works best when coverage is between 20 and 100x. In addition to high coverage analysis of the intermediate ARC results for these specimens showed that a repetitive element was assembled from reads recruited at an early iteration in both specimens. This element then led to recruitment of many more reads and the assembly of multiple short contigs at later iterations. These repetitive reads resulted in very long assembly times and eventual timeouts. Although the iterative ARC process did not run to completion in these cases intermediate contigs were still reported. In the case of S228 a full length mitochondrial genome was reported for 7 of the 11 targets, however a number of short non-mitochondrial contigs were also reported. A similar situation occurred for the six specimens in which ARC could not finish analysis using the Tasmanian devil mitochondrial reference. Recruitment of a large number of additional reads for this reference may be due to a “GT” rich repeat that is not present in the other mitochondrial sequences.

Repetitive sequence is a well known problem in HTS sequence analysis [20]. ARC has a number of built in mechanisms to mitigate problems caused by these sequences, including a masking algorithm that inhibits recruitment of reads by simple tandem repeats, tracking of read recruitment patterns that skips assembly if an unexpectedly large number of reads is recruited between iterations, and an assembly timeout that terminates assemblies that run beyond a specified limit. In addition to these strategies there is also an option to down-sample recruited reads in cases of very high legitimate sequence depth. Down-sampling was not used in any of the tests described here but it may have improved results for samples such as S10 which had large numbers of reads. During testing and development we have observed improved behavior with each of these measures and implementing them has allowed ARC to run quickly and efficiently on large datasets while minimizing the impact of repeat elements. However it is clear that in rare cases recruitment of repeat elements can still cause problems for single targets or samples.

We tested ARC's ability to assemble contigs with short, low quality reads recovered from ancient mammoth DNA and found that it did surprisingly well. The mitochondrial genome assemblies appear to be as good or better than the assembly of these reads published by Gilbert et al. [8] despite using a divergent reference with ARC. Assembly of the M1 mammoth sequence by Gilbert et al. [8] was achieved through mapping against another mammoth mitochondrial sequence published by Krause et al. [21] that was generated with a laborious PCR based strategy. Because ancient DNA sequencing projects are often targeted at extinct organisms [19] rarely is there a high quality reference from the same species against which to map reads during analysis unless labor intensive methods are used. This makes ARC an excellent choice for this type of data where a target sequence from a related extant organism is likely to successfully seed assembly. Even in the case where no closely related organism exists, a more distant reference may still be appropriate as was demonstrated by the assembly of two large contigs representing ~96% of the mammoth mitochondrial genome using a mouse reference. Lastly, ARC is that it can be configured to use multiple reference sequences as a single target. In cases where specimens cannot be identified ARC can still be used by selecting a set of homologous targets from phylogenetically diverse taxa to seed assembly.

Analysis of HTS data can be computationally intensive and time and memory requirements can become a serious limitation, especially with larger datasets [22]. With ARC, we have attempted to reduce these requirements using a 'divide and conquer' approach that breaks large HTS datasets up into multiple small problems, each of which can be solved quickly and with reduced resources. This approach allows the user to control memory usage simply by changing the number of CPU cores available to ARC as shown in Figure 4. Less than 3 Gb of RAM was required when using 10 cores, despite processing a large dataset that would require many times this amount of memory to analyze using traditional assembly. Of course, using fewer CPUs comes with the cost of longer analysis time so ARC has been designed to utilize larger computational resources when they are available.

It is useful to think of the DNA sequence mapping problem as a trade-off between sensitivity and specificity [23]. To avoid mapping reads to multiple loci throughout the reference, mapping parameters must be tuned for high specificity.

However, when divergent loci exist within the reference sequence then high specificity limits the sensitivity of the mapper and reads are left unmapped. Assembly on the other hand can be seen as mapping reads against themselves thereby removing difficulties associated with divergent reference loci, but incurring the burden of all-by-all comparisons which is significant in large datasets. ARC circumvents these problems by removing divergence from the reference through an iterative mapping and assembly process. As the intermediate reference is improved more reads can be recruited without sacrificing specificity, allowing both specificity and sensitivity to remain high. At the same time, because only a small subset of reads is assembled, the all-by-all comparisons remain less burdensome. This process is carried out in an automated, easily configured manner, with standardized output that simplifies additional analysis, or integration into existing sequence analysis pipelines.

Figures

Figure 3.1. *ARC flowchart*

The ARC algorithm consists of an initialization stage, followed by four steps that are iterated until stopping conditions are met, at which point a final set of contigs and statistics are produced.

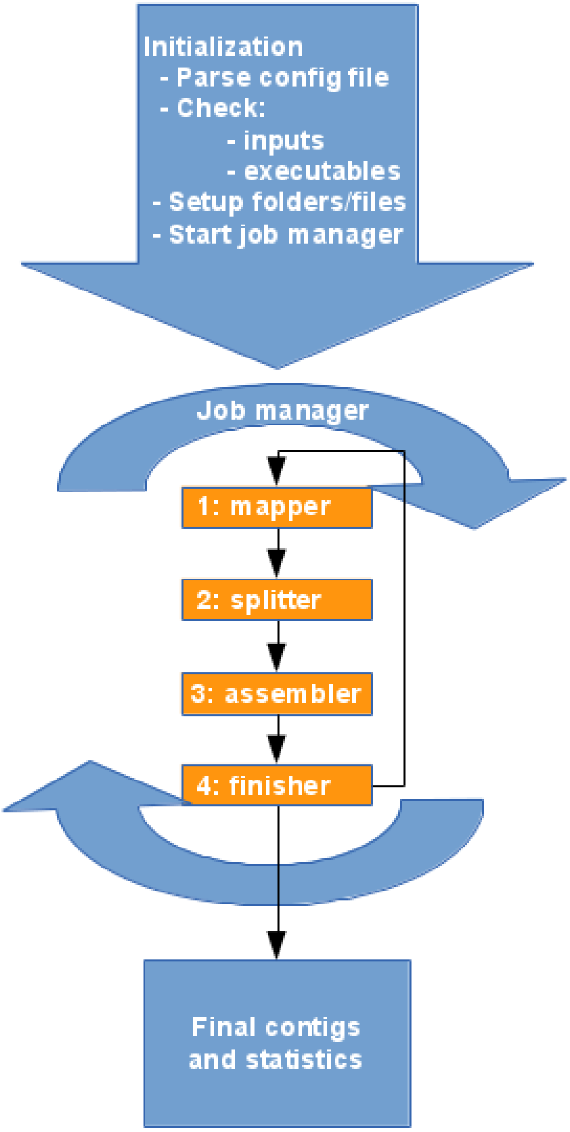


Figure 3.2. *Iterative assembly*
ARC is an iterative process for assembling homologous sequences. In iteration 1, a small number of reads and unmapped pairs are recruited to the more highly conserved regions of the divergent reference. These reads are assembled and the resulting contigs are used as mapping targets in the next iteration. This process is iterated until no more reads are recruited. Mapped reads are indicated in yellow, unmapped reads in orange. Paired reads are indicated with a connector. Both members of a pair are recruited if only one maps.

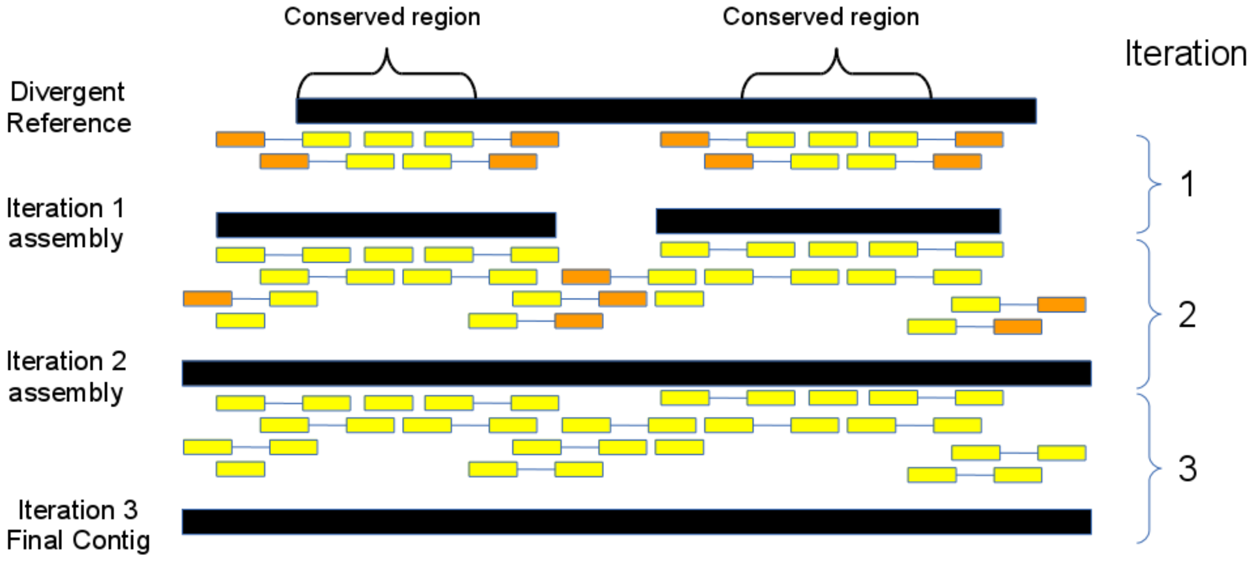


Figure 3.3. *Read recruitment*
Reads recruited at each iteration colored by target for specimen S152. The Gray-footed chipmunk target is the most similar and requires only 3 iterations to recruit the full set of reads. Other targets require more iterations, with Platypus requiring the most, however all arrive at the same set of reads.

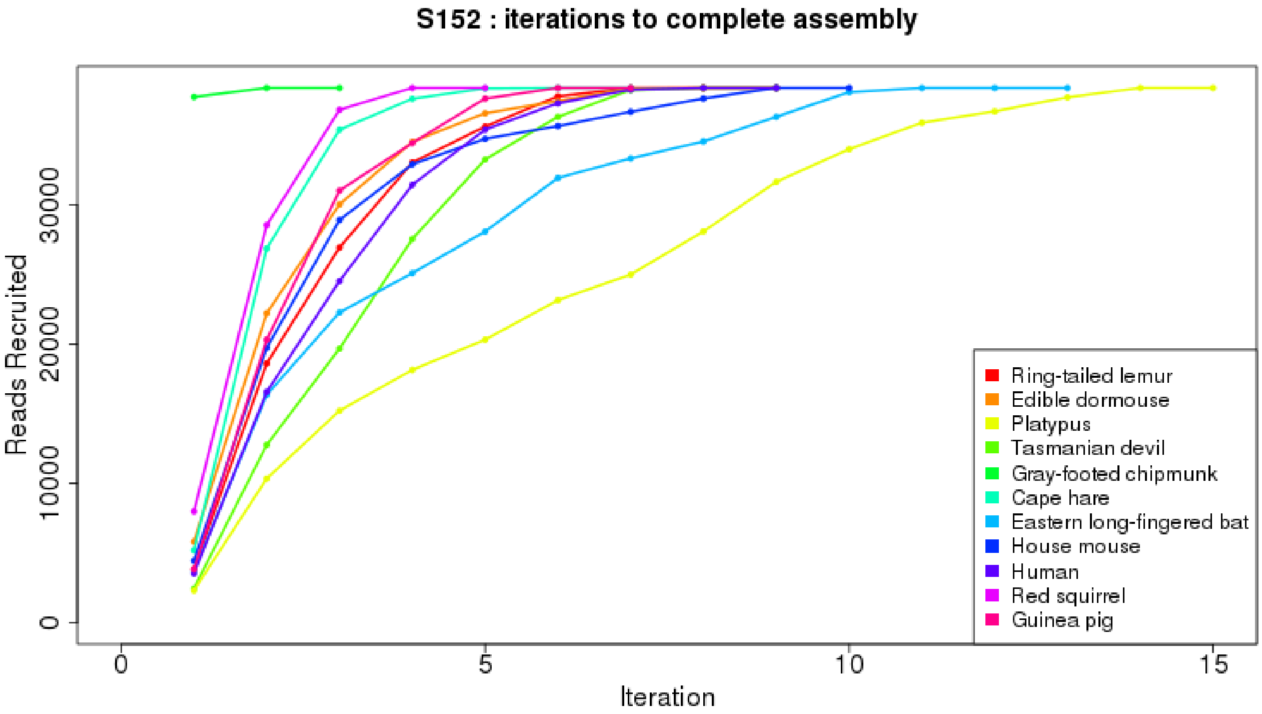


Figure 3.4. *Contig alignment*
Multiple alignment of ARC contigs assembled from Mammoth M1 reads aligned against the published M1 (EU153444) sequence and Asian elephant (AJ428946) mitochondrial genomes. The ARC contigs shown were assembled using the mammoth M13 reference, and are labeled “Mammoth ref”. All gaps indicated by red arrows are the result of homopolymers (a common problem in pyrosequencing data). An enlargement of the D-loop region showed that the published M1 sequence contains 10 unidentified bases (denoted by multiple “N”) followed by a sequence that aligns after a 370bp gap with respect to the Asian elephant reference. ARC assembles 220bp of this sequence (labeled with annotations “135bp novel ARC sequence” and “85bp novel ARC sequence”), providing evidence for a misassembly in the published EU153444 sequence.

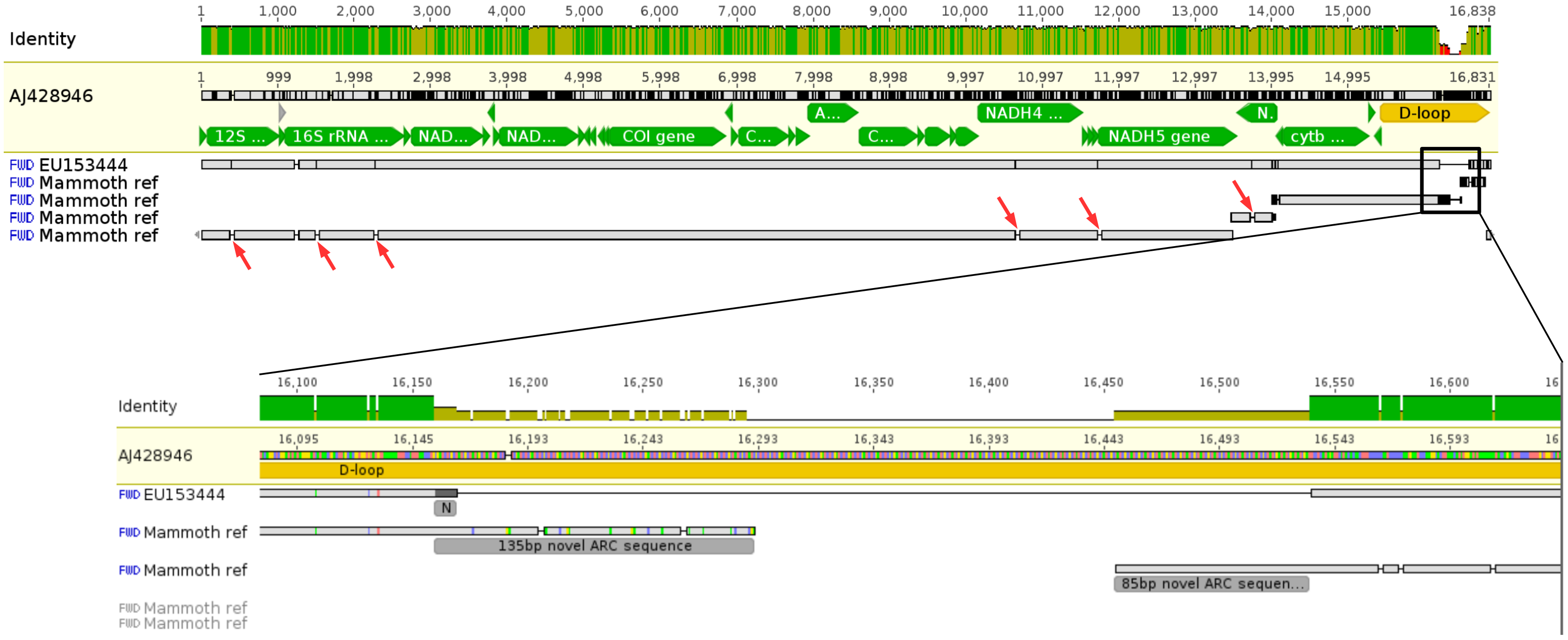
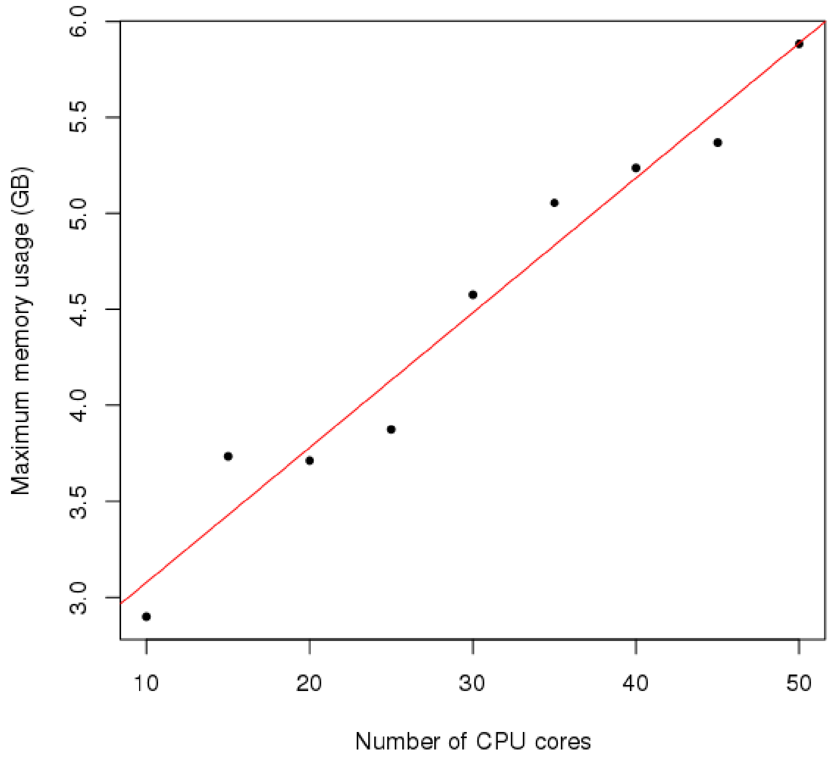


Figure 3.5. *Memory usage*
Memory usage scales linearly as a function of the number of CPU cores.



Tables

Table 3.1. *Assembly results*
Assembly of chimpmunk mitochondrial genomes using divergent references summarized by the number of targets that recruited similar sets of reads.

	Number of Specimens
All targets produce essentially the same results	32
Difference of > 100 reads in one target	15
Difference of > 100 reads in two targets	5
Difference of > 100 reads in four targets	1
No targets completed	2

Table 3.2. *Divergent references*
References used for assembly of chipmunk mitochondrial genomes. Percent identity is with respect to the Gray-Collared chipmunk (*Tamias cinereicollis*). Iterations columns indicate the number of ARC iterations required before assembly stopping conditions were met for this sample.

Reference	Species	Accession	Percent Identity	S152 Iterations
Tasmanian devil	*Sarcophilus harrisii*	NC_018788	72.10%	9
Ring-tailed lemur	*Lemur catta*	AJ421451	75.60%	9
Red squirrel	*Sciurus vulgaris*	NC_002369	80.00%	6
Platypus	*Ornithorhynchus anatinus*	NC_000891	71.20%	16
Human	*Homo sapiens*	HM156679	74.20%	8
House mouse	*Mus musculus*	NC_005089	75.50%	10
Guinea pig	*Cavia porcellus*	NC_000884	74.00%	9
Gray-footed chipmunk	*Tamias canipes*	(unpublished)	94.90%	3
Edible dormouse	*Glis glis*	NC_001892	76.60%	10
Cape hare	*Lepus capensis*	NC_015841	74.60%	7
Eastern long-fingered bat	*Myotis macrodactylus*	KF440685	73.40%	13

Table 3.3. *Ancient DNA assembly results*
ARC results for assembly of ancient mammoth DNA sequences. ARC produces a small number of contigs in all cases with good coverage and identity between the assembled contigs and published reference.

Reference	Contigs	Total Contig Length (bp)	Percent Coverage	Percent Identity	ARC Iterations	Reads
Mammuthus primigenius	4	16620	99.70%	98.10%	3	4633
Elephas maximus	4	16603	99.70%	98.20%	5	4631
Mus musculus	2	15781	95.90%	99.40%	78	4507

Table 3.4. *ARC performance*
ARC assembly of 55 specimens compared to individual *de novo* assemblies of three specimens (S151, S152, and S223). Maximum and average memory usage (RAM) is listed in gigabytes (GB). Total data processed is reported in millions of base pairs (Mbp).

	ARC	Newbler: S151	Newbler: S152	Newbler: S223
Total running time	77hr, 45min	31 min	1hr 13min	13hr 27min
Average Memory (GB)	22.78	5.847	8.337	16.36
Maximum Memory (GB)	31.19	6.71	9.967	17.54
Total assemblies performed	1,300,076	Not Applicable	Not Applicable	Not Applicable
Average assemblies per second	7.03	Not Applicable	Not Applicable	Not Applicable
Mbp unassembled reads	21913	243	367	629

References

1. Schbath S, Martin V, Zytnicki M, et al. (2012) Mapping reads on a genomic sequence: an algorithmic overview and a practical comparative analysis. J Comput Biol 19:796–813. doi: 10.1089/cmb.2012.0022

2. Li Z, Chen Y, Mu D, et al. (2012) Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph. Brief Funct Genomics 11:25–37. doi: 10.1093/bfgp/elr035

3. Li H (2012) Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. Bioinformatics 28:1838–44. doi: 10.1093/bioinformatics/bts280

4. Bradnam KR, Fass JN, Alexandrov A, et al. (2013) Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. Gigascience 2:10. doi: 10.1186/2047-217X-2-10

5. Abecasis GR, Altshuler D, Auton A, et al. (2010) A map of human genome variation from population-scale sequencing. Nature 467:1061–73. doi: 10.1038/nature09534

6. Li H (2011) Improving SNP discovery by base alignment quality. Bioinformatics 27:1157–8. doi: 10.1093/bioinformatics/btr076

7. Pyrkosz AB, Cheng H, Brown CT (2013) RNA-Seq mapping errors when using incomplete reference transcriptomes of vertebrates. arXiv Prepr arXiv13032411 1–17.

8. Gilbert MTP, Tomsho LP, Rendulic S, et al. (2007) Whole-genome shotgun sequencing of mitochondria from ancient hair shafts. Science 317:1927–30. doi: 10.1126/science.1146971

9. Kent WJ (2002) BLAT--the BLAST-like alignment tool. Genome Res 12:656–64. doi: 10.1101/gr.229202. Article published online before March 2002

10. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9:357–9. doi: 10.1038/nmeth.1923

11. Li H, Handsaker B, Wysoker A, et al. (2009) The Sequence Alignment / Map format and SAMtools. 25:2078–2079. doi: 10.1093/bioinformatics/btp352

12. Cock PJ a, Antao T, Chang JT, et al. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics 25:1422–3. doi: 10.1093/bioinformatics/btp163

13. Margulies M, Egholm M, Altman WE, et al. (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376–80. doi: 10.1038/nature03959

14. Bankevich A, Nurk S, Antipov D, et al. (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol 19:455–77. doi: 10.1089/cmb.2012.0021

15. Magoc T, Pabinger S, Canzar S, et al. (2013) GAGE-B: an evaluation of genome assemblers for bacterial organisms. Bioinformatics 29:1718–25. doi: 10.1093/bioinformatics/btt273

16. Bi K, Vanderpool D, Singhal S, et al. (2012) Transcriptome-based exon capture enables highly cost-effective comparative genomic data collection at moderate evolutionary scales. BMC Genomics 13:403. doi: 10.1186/1471-2164-13-403

17. Magoc T, Salzberg SL (2011) FLASH: fast length adjustment of short reads to improve genome assemblies. Bioinformatics 27:2957–63. doi: 10.1093/bioinformatics/btr507

18. Gilbert MTP, Drautz DI, Lesk AM, et al. (2008) Intraspecific phylogenetic analysis of Siberian woolly mammoths using complete mitochondrial genomes. Proc Natl Acad Sci U S A 105:8327–32. doi: 10.1073/pnas.0802315105

19. Knapp M, Hofreiter M (2010) Next generation sequencing of ancient DNA: requirements, strategies and perspectives. Genes (Basel) 1:227–243. doi: 10.3390/genes1020227

20. Miller JR, Koren S, Sutton G (2010) Assembly algorithms for next-generation sequencing data. Genomics 95:315–27. doi: 10.1016/j.ygeno.2010.03.001

21. Krause J, Dear PH, Pollack JL, et al. (2006) Multiplex amplification of the mammoth mitochondrial genome and the evolution of *Elephantidae*. Nature 439:724–7. doi: 10.1038/nature04432

22. Zhang W, Chen J, Yang Y, et al. (2011) A practical comparison of de novo genome assembly software tools for next-generation sequencing technologies. PLoS One 6:e17915. doi: 10.1371/journal.pone.0017915

23. Fonseca NA, Rung J, Brazma A, Marioni JC (2012) Tools for mapping high-throughput sequencing data. Bioinformatics 28:3169–77. doi: 10.1093/bioinformatics/bts605

Supplementary Material

Table S3.1. *ARC parameters*
A comprehensive list of parameters that can be used to control ARC behavior. An * indicates parameters that are required.

Parameter	Description
reference*	A fasta file contain one or more reference sequences.
numcycles	Maximum number of mapping and assembly cycles ARC will carry out Default: 1
max_incorporation	Control for repeat elements. If total reads recruited in the current cycle is greater than max_incorporation X reads recruited in previous cycle, assembly will not be carried out. Default: 5
bowtie2_k	Controls the max number of matches Bowtie 2 will report for each read. Default 5
format*	Format for files containing reads, can be fasta or fastq.
mapper*	Mapper to use during read recruitment, can be bowtie2 or blat.
assembler*	Assembler to use during assembly stage, can be newbler or spades
urt	Newbler parameter “use read tips” may reduce the number of ARC iterations by instructing Newbler to extend contigs using single reads at the edges of contigs. Note that ARC will not use 'urt' on the final iteration to ensure higher quality contigs. Default False
verbose	Output extensive logging details about ARC operation including all calls to external programs Default False
assemblytimeout	Amount of time (in minutes) ARC will wait for an assembly to finish before killing the assembly process. Adjusting this value can make assemblies of large targets possible, or reduce the impact of repeats on large ARC runs. Default 10.
cdna	Newbler parameter that enables experimental RNAseq assembly and read incorporation reporting. Newbler will be run in transcriptome assembly mode on the final ARC iteration. Default: False

continued on next page..

Parameter	Description
rip	Newbler parameter that instructs Newbler to only place reads in a single contig. In some cases Newbler will split a read placing parts of it in more than one contig. Default: False
subsample	Subsample read depth to a percentage of the orginal number of mapped reads. In cases where sequencing depth is great (>100x) it is often beneficial to only assemble a random subset of the mapped reads. For example, subsample=0.4 would cause ARC to retain 40% of mapped reads for assembly. Default: 1
maskrepeats	Causes ARC to mask simple tandem repeats in contigs before mapping. This results in recruitment of fewer reads contain repeats. Default: True
nprocs	Number of processors ARC should use. ARC can effectively make use of at least 64 cores when processing large jobs. Default: 1
fastmap	BLAT mapper parameter, runs BLAT in fastMap mode that requires high identity and doesn't allow insertions or deletions.

Listing S3.1. *Downloading and installing ARC*

- First, download the source:
  - *git clone https://github.com/ibest/ARC.git*
- Choose an installation option:
  - Option 1: Run without installing
    - Simply run ARC directly using: *./ARC/bin/ARC*
  - Option 2: Use a Python Virtual Environment
    - *mkdir pyenvironments*
    - *cd pyenvironments*
    - *virtualenv arc*
    - *source arc/bin/activate*
    - *cd /path/to/ARC/source*
    - *python setup.py install*
  - Option 3: install to the system path:
    - *python setup.py install*

Listing S3.2. *Patching BLAT to support FASTQ*

- This is process is modified from the instructions for a normal BLAT install. An additional step is added to add FASTQ support.
  - *wget http://users.soe.ucsc.edu/~kent/src/blatSrc.zip*
  - *unzip blatSrc.zip*
  - *patch -p0 </path/to/ARC/contig/blat+fastq_support.patch*

  - *cd blatSrc*
  - *export MACHTYPE=x86_64*
  - *mkdir ~/bin*
  - *mkdir ~/bin/x86_64*
  - *make*
- Executables for blat will now be located in the ~/bin/x86_64 folder.

Listing S3.3. *ARC configuration file*

```
## Comments are indicated by double pound signs, e.g. ##
## Parameters are indicated by a single pound sign, e.g. #
## parameters use a Name=value format
## Sample information is listed without a pound sign
# reference=targets.fa
# format=fastq
# mapper=bowtie2
# assembler=newbler
# nprocs=7
Sample_ID	FileName	FileType
Sample1 ./reads/Sample1_R1.fastq	PE1
Sample1 ./reads/Sample1_R2.fastq	PE2
Sample1 ./reads/Sample1_SE.fastq	SE
Sample2 ./reads/Sample2_R1.fastq	PE1
Sample2 ./reads/Sample2_R2.fastq	PE2
Sample3 ./reads/Sample3_SE.fastq	SE
```

Table S3.2. *Read recruitment*

Reads recruited per target for 55 chipmunk samples. Values in red represent assemblies that were killed due to incorporation of repetitive reads. Values in blue indicate targets that recruited a different number of reads from the rest of the targets (based on 100 or more reads differences from the median number of recruited reads). The “Common to All” column (in bold) displays the number of reads common to all targets, and “Mbp Sequence Data” is the total megabases of sequence data for each specimen after cleaning.

Sample	E. L. F. bat	Cape hare	Edible dormouse	G. f. chipmunk	Guinea pig	House mouse	Human	Platypus	Red squirrel	Ring-tailed lemur	Tasmanian devil	Common	Mbp Data
S10	**268,251**	**312,652**	**299,857**	**258,005**	**264,670**	**231,992**	**229,953**	**268,459**	**305,340**	**286,007**	**111,576**	**0**	971
S11	190,401	190,403	190,428	190,401	190,401	190,401	190,417	190,420	190,428	190,401	**223,363**	**190,382**	1,145
S12	238,138	238,121	238,138	238,156	238,146	238,143	238,132	238,130	238,129	238,129	**252,644**	**238,058**	1,143
S151	19,300	19,300	19,313	19,298	19,298	19,300	19,300	19,300	19,300	19,317	19,300	**19,293**	243
S152	38,388	38,388	38,388	38,389	38,388	38,389	38,388	38,388	38,389	38,388	38,388	**38,383**	367
S154	75,335	48,307	75,335	75,381	75,335	75,335	75,349	75,335	75,335	40,515	75,335	**35,223**	324
S155	109,443	109,443	109,443	109,460	109,443	109,440	109,443	109,443	109,439	70,089	46,312	**35,673**	635
S156	75,069	46,831	75,064	75,095	75,065	75,069	75,069	75,069	75,069	39,545	75,069	**33,623**	271
S158	21,921	21,921	21,921	21,926	21,921	21,921	21,921	21,921	21,921	21,921	21,921	**21,921**	346
S159	1,366	1,387	1,369	1,618	1,578	1,367	1,392	1,359	1,557	1,366	1,263	**1,046**	148
S160	2,448	2,448	2,448	2,448	2,448	2,448	2,448	2,448	2,448	2,448	2,321	**2,321**	68
S176	21,653	21,653	21,653	21,653	21,653	21,653	21,653	21,653	21,653	21,653	21,653	**21,653**	222
S185	8,637	8,637	8,637	8,637	8,637	8,637	8,637	8,637	8,637	8,637	8,637	**8,637**	108
S188	14,810	14,810	14,810	14,810	14,810	14,810	14,810	14,810	14,810	14,810	14,810	**14,810**	223
S201	14,888	14,886	14,888	14,889	14,888	14,888	14,888	14,888	14,888	14,888	14,966	**14,885**	614
S217	16,735	16,735	16,735	16,735	16,735	16,735	16,735	16,735	16,735	16,735	16,779	**16,735**	402
S218	6,399	6,399	6,399	6,399	6,399	6,399	6,399	6,399	6,399	6,399	6,440	**6,399**	266
S220	15,371	15,371	15,371	15,371	15,371	15,371	15,371	15,371	15,371	15,371	15,371	**15,371**	373
S222	30,352	30,344	30,352	30,368	30,344	30,352	30,352	30,344	30,352	30,352	30,352	**30,337**	559
S223	12,910	12,910	12,917	12,912	12,856	12,856	12,856	12,856	12,910	12,856	12,856	**12,845**	629
S225	9,136	9,136	9,136	9,163	9,136	9,136	9,136	9,136	9,162	9,162	9,188	**9,136**	543
S226	36,676	36,676	36,676	36,741	36,676	36,676	36,694	36,676	36,691	36,676	**61,790**	**36,672**	663
S228	**46,457**	**54,191**	**49,404**	**46,821**	**46,457**	**49,671**	**49,648**	**46,457**	**46,469**	**46,457**	**61,729**	**0**	316
S230	16,109	16,109	16,109	16,145	16,109	16,109	16,109	16,109	16,126	16,126	23,042	**16,109**	431
S231	54,832	54,832	54,832	54,893	54,832	54,832	54,832	54,832	54,832	54,832	22,595	**22,538**	487
S236	5,743	5,743	5,743	5,743	5,743	5,743	5,743	5,743	5,743	5,743	5,743	**5,743**	452
S237	84,987	84,987	84,987	84,993	84,987	84,987	84,987	84,987	84,987	84,987	34,103	**34,094**	587
S238	2,498	2,498	2,498	2,498	2,498	2,498	2,498	2,498	2,498	2,498	2,498	**2,498**	187
S250	9,959	9,959	9,959	9,979	9,959	9,959	9,959	9,959	9,959	9,959	9,959	**9,959**	510
S251	11,153	11,153	11,153	11,153	11,153	11,153	11,153	11,153	11,153	11,153	11,291	**11,153**	391
S256	8,096	8,096	8,096	8,096	8,096	8,096	8,096	8,096	8,096	8,096	8,137	**8,096**	370
S267	12,359	12,359	12,362	12,362	12,359	12,359	12,359	12,359	12,359	12,345	12,558	**12,336**	879
S324	11,100	11,100	11,100	11,100	11,100	11,100	11,100	11,100	11,100	11,100	11,100	**11,100**	100
S330	22,577	22,577	22,577	22,582	22,577	22,577	22,577	22,577	22,577	22,577	22,577	**22,577**	541
S335	17,712	17,712	17,712	17,712	17,712	17,712	17,712	17,712	17,712	17,712	17,712	**17,712**	182
S563	10,160	10,160	10,160	10,167	10,160	10,160	10,160	10,160	10,160	10,160	10,198	**10,160**	222
S572	20,922	20,922	20,922	20,922	20,922	20,922	20,922	20,922	20,922	20,922	20,922	**20,922**	104
S573	13,614	13,614	13,614	13,614	13,614	13,614	13,614	13,614	13,614	13,614	13,614	**13,614**	200
S574	17,632	17,632	17,632	17,632	17,632	17,632	17,632	17,632	17,632	17,632	17,632	**17,632**	246
S582	16,166	16,166	16,167	16,166	16,166	16,166	16,166	16,166	16,166	16,166	16,166	**16,166**	244
S587	22,785	22,785	22,785	22,789	22,785	22,785	22,785	22,785	22,785	22,785	22,785	**22,784**	384
S60	9,714	9,714	9,714	9,714	9,714	9,714	9,714	9,714	9,714	9,714	9,714	**9,714**	149
S600	37,871	37,871	37,871	37,872	37,872	37,872	37,872	37,872	37,872	37,871	15,898	**15,766**	391
S605	4,142	4,180	4,177	4,142	4,140	4,140	4,140	4,140	4,140	4,140	4,480	**4,137**	713
S613	17,008	17,008	17,008	17,008	17,008	17,008	17,008	17,008	17,008	17,008	**58,009**	**17,007**	388
S70	17,445	17,445	17,445	17,445	17,445	17,445	17,445	17,445	17,445	17,445	17,445	**17,445**	161
S700	5,082	5,082	5,082	5,082	5,082	5,082	5,082	5,082	5,082	5,082	**49,113**	**5,082**	609
S704	18,253	18,253	18,253	18,253	18,253	18,253	18,253	18,253	18,253	18,253	**21,254**	**18,253**	181
S705	4,815	4,815	4,815	4,809	4,815	4,809	4,809	4,815	4,809	4,809	**19,318**	**4,803**	292
S711	5,025	5,025	5,025	5,025	5,025	5,025	5,025	5,025	5,025	5,025	5,025	**5,025**	321
S713	2,043	2,043	2,078	2,078	2,078	1,908	2,078	2,078	2,078	2,043	1,961	**1,734**	511
S721	2,217	2,116	2,197	2,549	2,269	2,134	2,176	427	2,197	2,134	2,188	**326**	465
S781	31,547	31,547	31,547	31,547	31,547	31,547	31,547	31,547	31,547	31,547	31,588	**31,547**	279
S85	18,866	18,866	18,866	18,866	18,866	18,866	18,866	18,866	18,866	18,866	18,866	**18,866**	180

Chapter 4

StopGap, an approach for improving genome assembly

Abstract

High throughput sequencing (HTS) has brought about a revolution in the way that much of biological research is done. Despite wide application of these technologies, a number of problems during the data analysis stage remain unsolved. One of these is that the short sequences produced by HTS do not provide sufficient information to resolve repeated regions common in most genomes, resulting in assemblies which are broken into multiple fragments. The optimal assembly would perfectly represent the genome producing fully contiguous chromosomes with all repeat regions represented and properly placed in the assembly. Here we explored a strategy for filling gaps between assembled contigs to produce less fragmented assemblies without relying on a reference genome sequence. Our approach (StopGap) identifies gap-spanning Pacific Biosciences continuous long reads that are used to guide assembly of high quality Illumina or 454 reads with the Assembly by Reduced Complexity (ARC) pipeline. We evaluated the effectiveness of two assembly merging algorithms, CISA and Mix, to incorporate the contigs produced by ARC into assemblies produced with the Newbler and SPAdes assemblers. CISA was able to produce a more contiguous assembly and at the same time incorporate a number of long repeat sequences Mix was less successful, producing an assembly which was much longer than expected, duplicating a pair of plasmids, and failing to incorporate many repeats.

Introduction

High throughput sequencing (HTS) technologies have brought about a revolution in molecular biology [1]. Despite the popularity of these technologies, a number of outstanding problems remain to be solved for the analysis of HTS data. One of these is that short reads are difficult to assemble into a high quality “finished” representation of the original sequence, often due to repetitive sequence [2]. Instead sequence assembly algorithms produce a number of short sequences, each a fragment of the full genome with unknown order, orientation, and gap size between fragments. These fragmented assemblies can make it difficult to identify and annotate open reading frames, leading to errors in comparison of gene content between isolates [2]. Comparative structural analysis of genomes using fragmented assemblies is also difficult [3], particularly since large-scale changes such as inversions, duplications, insertions and deletions often occur in the context of repeat sequences [4, 5]. Although the problem of fragmented genome assemblies is particularly serious for eukaryotic genomes, even relatively small and simple bacterial genomes are often difficult or expensive to assemble completely. Given sufficient sequencing depth and high quality reads, the typical cause for assembly fragmentation is repeated sequences in the genome [2, 6]. If these repetitive regions are longer than the length of sequenced reads then there is no way to correctly incorporate them into the assembly using information from the reads alone; a problem that is illustrated in Figure 1. Rather than risk incorporating the repeats incorrectly, assemblers typically break the assembly into a set of sequences called contigs. Some of these contigs represent repetitive sequences which occur multiple times within the genome while others represent non-repetitive sequence.

A number of strategies have been developed to address the issue of fragmented assemblies. These strategies include long-insert sequencing using specialized library preparation protocols (e.g. Illumina “mate-pair”: http://www.illumina.com/technology/ mate_pair_sequencing_assay.ilmn and Roche/454 “paired-end”: http://454.com/ applications/whole-genome-sequencing/#paired-end-sequencing) to bridge repeats and sequentially organize contigs (often referred to as “scaffolding the genome” or “resolving repeats”) [7]. Alternatively, a genome sequence can be “finished” by closing gaps using a combination of PCR and Sanger sequencing (e.g. Sheppard et al., 2013), or by nonsequencing based approaches such as scaffolding contigs using optical mapping of restriction site patterns [9] (e.g. through OpGen MapIt Services: http://www.opgen.com). In addition to these, a large variety of bioinformatic approaches have been developed including tools such as ABACAS, which orders and orients contigs by aligning them against a closely related reference genome [10], and GapFiller, which recruits paired reads that fall within a gap of estimated size and attempts to assemble a sequence that fills this gap without violating the gap-size constraint. Another bioinfomatic strategy is to use multiple assemblers optimized for different data types, or a single assembler with multiple different settings to assemble a dataset. The resulting assemblies are then combined to form a final set of contigs. A number of software packages have been developed for this task including Mix [11], MAIA [12], CISA [13], Graph Accordance Assembly (GAA) [14], minimus2 [15] and GAM-NGS [16] among others. Two of these (minimus2 and GAA) were evaluated by Magoc et al. [17] who report improved N50 size with GAA for some combinations of assemblies. To our knowledge, little additional work has been done to compare these contig integration strategies, and it is currently unclear whether they properly handle contigs representing repeated sequences.

The emergence of single molecule real time sequencing, which is currently only available from Pacific Biosciences (PacBio), has presented a number of new opportunities for genome assembly. Reads from this technology come in two varieties, both of which were analyzed by Ono *et al.* [18] while developing the PacBio read simulation program PBSIM. The first variety, continuous long reads (CLR), have lengths as high as 22 Kbp but have high error rates. When compared to error free sequence, percent identity ranges from 76.19 to 83.81, substitution rates from 0.67 to 1.75 percent, insertion rates from 8.40 to 10.80 percent, and deletion rates from 1.60 to 4.63 percent for CLR reads. The second variety, circular consensus sequences (CCS), are shorter with maximum lengths of 2,605 bp, however percent identity is much better at 97.43 to 98.23, with lower substitution

(0.09 to 0.19), insertion (0.70 to 0.86), and deletion (1.04 to 2.34) rates. Despite their high error rate, PacBio CLR reads have attracted significant attention because of their potential to resolve repeats. For example, the software package PBjelly [19] attempts to close gaps between scaffolded contigs. This strategy recruits PacBio reads that map at the ends of scaffolded contigs and then assembles these reads in an attempt to reduce error. *English et al. [19]* report that this strategy results in filled gaps that have 91.7% mean similarity to Sanger validation sequences suggesting that this attempt at error correction is only partially successful.

Although each of these approaches is developed for closing gaps in genomes where repeat sequences are commonly encountered, none take copy number into account when joining contigs. Additionally, with a few rare exceptions such as Sheppard *et al.* [8] who used read coverage relative to the chromosome to estimate copy number of nine native plasmids in *Bacillus thuringiensis*, few authors of draft genome assemblies include information about the number of copies of repetitive contigs, or even identify contigs as being repeats.

In this study we explore an alternative strategy for incorporating information from PacBio CLR reads into an existing assembly. We hypothesized that by using the repeat spanning PacBio sequences to recruit short, high quality Illumina or Roche/454 reads, we may be able to provide enough information for a sequence assembler to produce high-quality, repeat spanning contigs that could then be used for gap closure. To test this hypothesis we implemented StopGap, an extension to the Assembly by Reduced Complexity (ARC) pipeline that was originally designed to facilitate targeted assembly of homologous sequences. Instead of using homologous sequences from a related species as targets, we use low quality PacBio reads that span gaps between contigs. We then tested two recently published assembly merging algorithms, Mix [11] and CISA [13] to see if they could make use of the resulting ARC contigs to produce more contiguous assemblies both in the presence and absence of known repeat contigs.

Methods

Datasets

Datasets for this analysis were produced as part of a study that examined how broad host range plasmids that encode multiple antibiotic resistance genes evolve to become stably maintained in novel bacterial hosts. The host bacterium, *Pseudomonas moraviensis* strain R28-s was isolated from the municipal wastewater treatment plant of Moscow, Idaho, USA as a transconjugant after it acquired plasmid pB10::*rfp [20].* DNA from this bacterium was extracted and sequenced using three sequencing platforms, Illumina MiSeq, Roche 454 and PacBio RS II and protocols recommended by the manufacturer. A draft quality assembly of the host genome has been deposited in Genbank (accession: AYMZ00000000) and a genome announcement was recently published [21].

Illumina reads were sequenced in the Genomics Resources Core of the Institute for Bioinformatics and Evolutionary Studies (IBEST GRC) at the University of Idaho using an Illumina MiSeq sequencer and 300 bp paired end kit. This sample was multiplexed with a number of others and produced 459 Mbp of sequence data in 1,528,717 paired reads. Prior to analysis the reads were preprocessed through a read cleaning pipeline consisting of the following steps. In the first step, duplicate read pairs (possibly resulting from multi-cycle PCR reactions carried out as part of library preparation) were removed using a custom Python script. Sequences were then cleaned to remove sequencing adapters and low quality bases using the software package Seqyclean (Zhbannikov et al. manuscript in prep, https://bitbucket.org/izhbannikov/seqyclean). Finally, because paired-end sequencing produces two reads from both ends of a single template, it is often possible to overlap these reads and form a single long read representing the template in its entirety. This overlapping was carried out using the Flash software package [22]. The majority (83 percent) of paired reads could be overlapped in this dataset producing combined, single reads with an average length of 201 bp. After cleaning, a total of 273 Mbp (43x expected coverage) in 1,168,085 single end, and 184,608 paired end reads was retained for further analysis.

454 reads were also sequenced at the IBEST GRC using the Roche 454 Titanium chemistry. Reads were cleaned to remove sequencing adapters and low-quality bases using the sequence cleaning pipeline Seqyclean (Zhbannikov et al. manuscript in prep, https://bitbucket.org/izhbannikov/seqyclean). After cleaning, 92 Mbp (14.7x expected coverage) in 233,851 reads that mean length 395 bp were retained for further analysis.

PacBio reads were sequenced at the Mount Sinai School of Medicine DNA Core Lab using a single RS II SMRT cell that produced 336 Mbp of sequenced data in 155,126 reads corresponding to approximately 53x coverage. Reads were cleaned by the sequencing center using PacBio processing protocols. After cleaning, the mean read length was 2,168 bp, the maxium read length was 21,990 bp, and 7657 reads were greater than 5 Kbp in length.

Genome Assembly

De novo shotgun assembly of Illumina and 454 reads was done using SPAdes v3.0 [23] with the *–careful* option, and Newbler v2.8 [24] using default parameters. SPAdes is a De Bruijn graph based assembler designed for bacterial genomes that utilizes a number of complex strategies for improving assemblies including a sequencing error correcting stage, automatic use of multiple k-mer lengths during assembly, incorporation of linkage information in paired reads, and support for highly variable coverage. SPAdes has performed well in two recent comparisons of genome assemblers [17, 25]. Newbler was originally designed to assemble reads generated by the 454 pyrosequencing platform [24] but recent versions have added support for Illumina paired-end reads.

PacBio Read Selection and ARC processing

PacBio CLR reads have high error rates which makes them unsuitable for most assemblers. We attempted to resolve this problem by using PacBio reads to recruit high quality Illumina and 454 reads, which could then be assembled, creating a contig representative of the original PacBio read. To do this, we implemented an ARC extension called StopGap as a pre-processing step to incorporate information from the long, error prone PacBio reads into the final assemblies. StopGap both identifies potentially repetitive contigs within an assembly, and assembles contigs which can resolve these repeats in the following sequence of steps:

1) *Estimating Contig Copy Number and Identifying Repetitive Contigs*

Bowtie2 is used to align short reads against contigs using default parameters, the resulting SAM file is then processed with Samtools [26] to calculate per-position mapping depth. The global mean mapping depth across all contigs is calculated allowing for an estimate of copy number to be derived by dividing average mapping depth for each contig by the global mean mapping depth. Contigs with a ratio greater than 1.2 (indicating 1.2 copies of said contig relative to all other contigs) are classified as being potentially repetitive. Contigs representing putative repeats are annotated with the estimated copy number and written to a *repeat_contigs.fasta* file while all other contigs are written to *norepeat_contigs.fasta*. A report file is also generated at this step indicating contig identifier, length, average mapping depth, and copy number for all contigs. In some cases multi-copy contigs may also represent circular plasmids. StopGap makes no attempt to discriminate between these and genomic repeats, and it is left to the user to identify and handle these plasmid contigs properly.

2) *PacBio Read Selection*

Putative gap-spanning PacBio reads are recruited using both single-copy, and repetitive contigs identified in the previous step. The ends of these contigs are trimmed back 100 bp in order to remove potential misassemblies, and then 500 bp sequences are cut from both ends. These “contig ends” are aligned against PacBio reads using the alignment tool BLAT [27] with *minScore=300* and *minIdentity=75* parameters. All PacBio reads that align with two or more contig end sequences are recruited as potentially bridging gaps, and used in the next step.

3) *Assembly by Reduced Complexity*

The Assembly by Reduced Complexity (ARC) pipeline (manuscript in preparation: https://github.com/ibest/ARC) was designed to facilitate assembly of discreet, homologous sequences given a divergent set of references. PacBio reads selected in the previous step were used as target sequences for ARC assembly. ARC was run with the following settings, *mapper=blat, assembler=newbler,* and *numcycles=1*.

The Blat sequence aligner supports a number of parameters for controlling alignment scoring and reporting. We found that increasing the *minScore* parameter to 90 (instead of the default setting of 30) reduced the number of reads recruited but resulted in improved ARC assemblies (data not shown). Support for this more stringent setting was added to ARC and can be enabled by setting *pacbio=True* in the configuration file. Newbler was also tested in both mapping based, and *de novo* assembly modes. In mapping mode, Newbler first aligns reads to a reference and then does local-assembly of the reads using a multiple alignment step. This step allows Newbler to resolve differences between reads and recover insertions and deletions with respect to the reference (454 Sequencing System Software Manual, v2.8). Support for using Newbler mapping mode in ARC was added and can be enabled by setting *NewblerMap=True*. We found that using Newbler in mapping mode also improved the final ARC contigs even when the more stringent Blat parameters described above were not used. For the results reported in this study we used the following additional settings for ARC: *NewblerMap=True, pacbio=False*.

4) *Screening ARC Assemblies for Gap-Spanning Contigs*

The contigs produced by the ARC pipeline were aligned against the contig end sequences extracted in step 2, again with BLAT, but with a more stringent *minIdentity=95* parameter. ARC contigs that aligned to two or more contig end sequences were retained for further analysis. This set of contigs is referred to as “bridge contigs” in the remainder of this manuscript.

Merging Assemblies

Two assembly merging algorithms, Mix [11] and CISA [13] were tested. We wished to compare the results of these algorithms both with and without ARC contigs, and also to explore how repeat contigs would be handled by these algorithms. To this end we merged four different combinations of the data using both programs: 1) all contigs produced by Newbler and SPAdes *de novo* assemblies, 2) non-repetitive contigs produced by Newbler and SPAdes *de novo* assemblies, 3) bridge contigs plus all contigs produced by Newbler and SPAdes *de novo* assemblies, 4) bridge contigs plus non-repetitive contigs produced by Newbler and SPAdes *de novo* assemblies.

CISA was run as per instructions in the manual (http://sb.nhri.org.tw/CISA) using all four combinations listed above. Configuration files were created and contigs were merged into a single file (*python Merge.py Merge.config*). The main CISA pipeline was then run to produce the final, combined assembly (*python CISA.py CISA.config*).

Mix was also run for all four combinations listed above. Mix is slightly more complicated to run than CISA, but does not require a configuration file. The first step combines contigs into a single fasta file, with a prefix included for the names of each contig to distinguish which set it belongs to (*preprocessing.py -o contigs.fa ../AllNewblerContigs.fasta ../AllSPAdesContigs.fasta*). The second step (*nucmer --maxmatch -c 30 -l 30 -banded -prefix=alignments contigs.fa contigs.fa*) aligns contigs against themselves using Nucmer, a many verses many DNA alignment algorithm that is part of the MUMmer package [28]. Next, alignment information is processed and summarized with show-coords, which is also part of the the Mumer package (*show-coords -rcl alignments.delta > alignments.coords).* Finally, the Mix algorithm is run to combine contigs based on alignment information produced in the previous steps (*Mix.py -g -a alignments.coords -c contigs.fa -o ./ -C 300 -A 200*).

Results

Assemblies and Repeat Contigs

In this study we explored a strategy for incorporating information from PacBio CLR reads into an existing assembly with the objective of closing gaps between contigs and better incorporating repeat elements into the final assembly. To evaluate this strategy, we used a combination of Illumina, 454, and PacBio CLR reads sequenced from the bacterial isolate *Pseudomonas moraviensis* strain R28-s.

After preprocessing the Illumina and 454 reads, the first step in this analysis was to generate d*e novo a*ssemblies of these reads using the Newbler v2.8 and SPAdes v.3.0.0 assemblers. Contigs produced in this step will be referred to here as “*de novo* contigs”. Statistics for the resulting assemblies are presented in supplementary table S1 and summarized in Table 1. Both assemblers produced relatively good assemblies for a genome of this size with some contigs greater than 700 Kbp in length and similar total lengths (Newbler: 6,286,279 bp, SPAdes: 6,297,403 bp). However, the SPAdes assembly was much less fragmented overall with higher mean contig length (217,200 vs 161,200 bp) and an N50 value which is more than twice as high as that produced by Newbler (853,530 vs 338,867 bp). The N50 value represents a “balance point” at which half of the total assembled sequence length is accounted for in contigs as big or bigger than the N50 value. A large N50 value is therefore indicative of a less fragmented assembly.

Following assembly, Illumina and 454 reads were mapped against the assembled contigs to calculate mapping coverage and infer copy number. Results of this analysis are reported in Table 2, and in supplementary table S1. Nine contigs greater than 500 bp in length were identified as being putative repeats in both assemblies, and an additional 3 repeat contigs less than 500 bp were reported for SPAdes. Two of the repeat contigs are plasmids known to exist in this isolate. Both were fully assembled by both assemblers. The plasmid assemblies have similar lengths (81,744 vs 81,973 bp and 13,120 vs 13,247 bp) and pairwise alignments showed that in both cases, the assembled plasmids were 100% identical to each other at all aligned sites. The slightly longer SPAdes contigs have a small overlap at each end of the contig, while the Newbler assembly has no such overlap. These plasmids are circular, explaining this small difference and overlap, however it is interesting to note that both SPAdes and Newbler linearized these circular sequence at approximately the same position. The rest of the repeat contigs presented in Table 2 have less obvious analogs between the two assemblies. However, by using Blat to align the sets of contigs against each other, we have been able to resolve the relationships for all identified repeats. The naming schemes used by Newbler (contig000NN) and SPAdes (NODE_NN) are retained here to match the repeat contigs listed in Table 2. The relationships between the repeat contigs in both assemblies are summarized here:

- Newbler contig00026 is part of three SPAdes contigs, overlapping NODE_15 on one end, containing NODE_25 completely, and overlapping NODE_7 on the other. Further analysis of mapping depth across this contig showed that it is consistently high (data not shown), suggesting that the full-length 32 Kbp sequence is present twice within the genome.

- Newbler contigs contig00031 and contig00032, are represented in SPAdes NODE_16 overlapping the SPAdes contig at either end. Based on mapping depth and the Newbler results, it appears that contig00031 and contig00032 may be connected to each other in some cases, but occur individually in others.
- Newbler contig00035 is partially represented in four SPAdes contigs, NODE_15, NODE_17, NODE_27, and NODE19. NODE_15 and NODE_17 both have a 377 bp overlap with one end of contig00035, NODE_27 is fully contained within this contig, and NODE_19 has a 2,790 bp overlap at the other end.
- Newbler contig00036 and NODE_20 are perfect matches.
- Newbler contig00038 is contained entirely within NODE_3. The SPAdes NODE_3 contig is 890 Kbp long, and the putative Newbler repeat contig00038 (2,263 bp long) is present at one end of this much longer contig.
- Newbler contig00008 is contained twice within NODE_1, with 3,434 bp of sequence between the two copies
- SPAdes NODE_18 was identified as a putative repeat, and fully contains Newbler contig00034 which was not. Both have low average coverage as compared to the other putative repeats and may have been mis-identified as repeats.
- SPAdes NODE_24 is present in two Newbler contigs, contig0030 and contig0037, overlapping 590 bp at the end of each Newbler contig.

Repeat elements were clearly problematic for these assemblers. There is one repeat that was similarly represented in both assemblies (Newbler contig00036 and SPAdes NODE_20). The remainder of the repeats were assembled in different ways by the two assemblers. In some cases a repeat is incorporated into a longer, non-repetitive contig by one of the two assemblers while at other times the repeat is broken into multiple smaller contigs in one of the two assemblies. Based on this comparison the following set of repeat contigs was chosen to characterize the behavior of Mix and CISA when repeat elements are present in the contigs: contig00023*, contig00026*, contig00028, contig00031, contig00032, contig00035, contig00036, contig00038, and NODE_24.

Putative Gap-filling PacBio Reads and ARC Results

Contigs assembled by Newbler were used as a starting point for recruiting PacBio reads that could aid in closing gaps. These reads were recruited by aligning the ends of Newbler contigs against the PacBio reads using Blat and then filtering the results for PacBio reads that had two or more high scoring hits. Using this strategy, 980 putative gap-bridging PacBio reads were identified. Of these, 808 aligned to two contig ends, 130 aligned to three contig ends, 33 aligned to 4, and 9 aligned to five or more. ARC was run using these PacBio reads as targets. ARC assembly produced a total of 3,319 sequences. These sequences were filtered, again by alignment against the contig ends used to screen PacBio reads. Of the 3,319 sequences produced by ARC, 653 were identified as potentially bridging gaps, and were retained for further analysis.

A similar process was carried out using the contigs assembled by SPAdes. Following the first round of filtering, 830 putative gap-bridging PacBio reads were identified. 489 aligned to two contig ends, 105 aligned to three contig ends, 103 aligned to four contig ends, and 133 aligned to five or more contig ends. Using ARC with this set of 830 PacBio reads as targets produced 2,537 contigs, and of these 646 were identified as potentially bridging gaps and retained.

All subsequent analysis steps involving ARC contigs used both sets of potentially gap-bridging contigs (653 based on the Newbler assembly, and 646 based on the SPAdes assembly). This combined set of contigs are referred to as “bridge contigs” here.

Merging Assemblies

Bridge and *de novo* contigs were combined with two recently published tools, Mix [11], and CISA [13]. The resulting combined assemblies are summarized in Table 3. Merging was done with four different combinations of the *de novo* contigs and bridge contigs. These four combinations include the following:

1. Non-repetitive contigs produced by Newbler and SPAdes de novo assemblies, **C** and **M** in Table 3.
2. All contigs (including repeats) produced by Newbler and SPAdes *de novo* assemblies, **C-R** and **M-R** in Table 3.
3. Bridge contigs plus non-repetitive contigs produced by Newbler and SPAdes *de novo* assemblies, **C-B** and **M-B** in Table 3.
4. Bridge contigs plus all contigs produced by Newbler and SPAdes *de novo* assemblies, **C-R+B** and **M-R+B** in Table 3.

We used Mix and CISA with these four combinations because we wished to compare the results of these algorithms both with and without the bridge contigs produced by ARC and to explore how repeat contigs would be handled by these algorithms.

Mix and CISA performed similarly when combining assemblies without using bridge contigs. When combining only non-repeat contigs (**C** and **M**), CISA was more aggressive than Mix, producing 10 contigs while Mix produced 14. The total length of Mix contigs was also higher by 31,203 bp, suggesting that CISA identified reasonably large overlaps. Using the full set of 39 Newbler contigs and 29 SPAdes contigs produced by *de novo* assembly (**C-R** and **M-R**), both CISA and Mix produced 17 contigs with the same maximum size and N50. The only difference was the total number of bases which differed by 1,077 bp. Both of these results represent a significant decrease in the total number of contigs, indicating a more contiguous assembly.

Differences between CISA and Mix became more pronounced when the bridge contigs were included. CISA again produced 10 contigs with and without repeat contigs (**C-B** and **C-R+B**), maximum contig length and N50 values both increased. In contrast Mix was able to combine fewer contigs after the addition of bridge contigs (**M** vs **M-B** and **M-R** vs **M-R+B**). The final number of contigs increased, but more troublesome was that the total length of the assembly grew significantly, reaching values well above 7 Mbp.

To characterize the behavior of Mix and CISA with respect to repeat sequences we counted the number of times a set of repeat contigs was incorporated into the combined assemblies (see Table 2 and associated text). We expected that CISA and Mix would be able to utilize bridge contigs to incorporate these repeats into the assembly and close gaps. Counts for repeated contigs are presented in Table 4. As expected, the two plasmids (contig00028* and contig00023*) appeared only once in each CISA output where repeat contigs were available. On the other hand, Mix duplicated the plasmids when both repeat contigs and bridge contigs were available, creating a single sequence for each plasmid which twice contained the plasmid. The authors of CISA and Mix do not mention support for circular sequences in their respective manuscripts, however CISA clearly performs better here. Neither CISA or Mix was able to improve incorporation of repeats contig00026, contig00035, and contig00038 into the assemblies. In all cases, either one, or no copies were represented in the final assemblies.

Incorporation of the remaining repeat contigs, conti00036, NODE_24, contig00031, and contig00032 differed substantially between CISA and Mix. For all but contig00032, only a single copy of each repeat was represented in the Mix results, even when bridge contigs were available. In contrast, when bridge contigs were available CISA incorporated these repeats into the final assemblies in quantities that were comparable to the estimated copy number. With an estimated copy number of 2.55, Contig00036 was incorporated twice in both cases where bridge contigs were available. NODE_24 was already present twice in the *de novo* Newbler assembly, but was only represented once in all of the Mix results, while it was present twice in all CISA results with the exception of the CISA assembly where no repeat contigs or bridge contigs were provided. Contig00031 and Contig00032 showed especially large differences. Contig00031 has an estimated copy number of 5.36 and is present 4 times in CISA assemblies where bridge contigs were available. Mix appears to have been unable to make use of the bridge contigs, with one copy of contig00031 represented when repeats were available, and no copies without these contigs. Mix does slightly better with contig00032, incorporating it twice in the final assemblies, but CISA was able to incorporate it three times.

Both algorithms were able to reduce fragmentation of the assembly as compared to the results from Newbler and SPAdes alone. However, CISA clearly outperformed Mix in these comparisons by producing more contiguous assemblies with fewer contigs, incorporating more repeats and increasing N50, handling circular plasmids appropriately, and not bloating the assembly well beyond the estimated genome size of approximately 6.3 Mbp.

Discussion

In this study we explored a strategy for improving *de novo* assemblies of short reads by incorporating information from PacBio CLR reads to resolve repeats and produce more contiguous assemblies. Because PacBio reads are of very poor quality they were not used directly for assembly, but were instead used as mapping targets in the ARC pipeline to recruit short, high quality Illumina and 454 reads that were then assembled on a per-target basis. Rather than attempting to use all of the PacBio reads available, we first screened the set of reads for those that were more likely to bridge gaps in the assemblies. Using this strategy, ARC was able to assemble bridge contigs for 71.76% of the PacBio reads.

We then tested whether two recently released software packages, CISA and Mix, could use these bridge contigs to reduce fragmentation of the assembly and better represent the number of repeats estimated to exist in the genome based on mapping depth. Both of these packages were able to produce less fragmented versions of the genome when combining *de novo* assemblies from SPAdes and Newbler and performed similarly with this data alone. Fragmentation of the assembly was also reduced by both Mix and CISA after combining the *de novo* and bridge contigs. However, only CISA was able to incorporate repeats in numbers approaching our expectations based on mapping depth, while at the same time raising N50 and maintaining an expected total genome size. On the other hand, Mix generally failed to incorporate repeats, despite producing assemblies that were much larger than expected, the largest being 7.5 Mbp instead of the expected 6.3 Mbp. Additionally Mix duplicated a pair of circular plasmids, representing each twice in the final output.

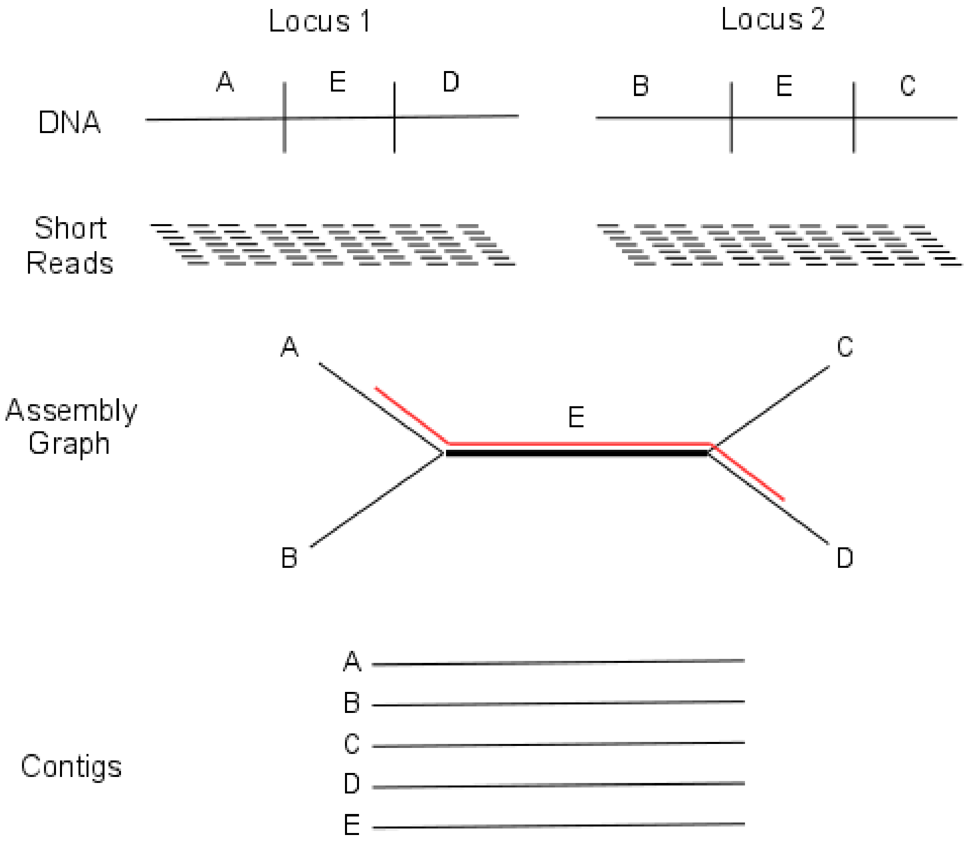
Analysis of mapping depth for contigs produced by Newbler and SPAdes showed that this genome contains at least 8 repeats larger than 500 bp with the longest being 32 Kbp in length. However identification of repeats proved to be more difficult than expected, for example the large 32 Kbp repeat was incorporated into one of the non-repetitive contigs in the SPAdes assembly, while Newbler produced a separate contig for this sequence. Mapping depth was consistently higher than expected across the 32 Kbp region suggesting that this was in fact a repeat but that SPAdes incorporated it anyway. A similar situation occurred for NODE_16, which was split into Contig00031 and Contig00032 by Newbler. Mapping coverage for NODE_16 dips at the junction between these two contigs suggesting that it is in fact a pair of repeats that sometimes occur together. Mapping these two contigs against the CISA results also support this interpretation with four copies of Contig00031 and three copies of Contig00032 integrated into the assembly. Despite the complex nature of these repeats, identifying repeats and estimating their copy number is a useful exercise when attempting to finish a genome. For example, in this genome the longest repeat was 32 Kbp, while the longest PacBio read was much shorter at 21.9 Kbp making it impossible to solve this repeat with the available PacBio data. Other repeats were shorter, 5.3 Kbp or less in length, however with only 7657 PacBio reads longer than 5 Kbp, it is probable that some instances of these repeats were not fully captured within a single PacBio read.

Optimally, the final assembly for this bacterium would consist of three contigs, one for each plasmid plus a third 6.3 Mbp contig representing the entire genome and incorporating repeats the appropriate number of times. Neither Mix nor CISA was able to produce this result when combining contigs produced by *de novo* assembly of the reads using the Newbler and SPAdes assemblers alone. However, addition of bridge contigs assembled with ARC allowed CISA to get much closer, reducing the genome to just ten contigs, two of which were plasmids. In addition, the N50 for these CISA assemblies was improved, and the number of repeats more closely agreed with our expectations based on mapping depth. These improvements show that the StopGap strategy is an effective method for taking advantage of PacBio CLR data for improving genome assemblies.

Figures

Figure 4.1. Repeat Elements

A simple example of a repeat element. The repeat element E exists at two different loci within the DNA. Short sequenced reads are only sufficient to cover the junctions between E and its immediate neighbors. Assembly of these reads collapses E into a single sequence, resulting in an graph-structure which cannot be resolved because it is impossible to determine whether A-E-D and B-E-C or A-E-C and B-E-D are correct. This results in 5 contigs (A, B, C, D, E), each broken at the junction with E. Note that twice as many reads would map to E because it actually exists twice in the genome. This puzzle may be resolved with the addition of a long read (in red) which bridges the repeat, providing a sure path between A and D.



Tables

Table 4.1. Summary Statistics for Assembly

Summary statistics for Newbler and SPAdes assemblies.

	Newbler	SPAdes
Total Contigs > 500 bp	39	24
N50	338,867	853,530
Mean Length (bp)	161,200	217,200
Total Length (bp)	6,286,279	6,297,403
Repeats > 500 bp	9	9
Largest repeat (bp)	32,393	5,312

Table 4.2. Putative Repeat Contigs

Putative repeat contigs assembled by Newber v2.8 and SPAdes v3.0 on the same set of reads. Plasmids are indicated with *. Average mapping depth is reported and estimated copy number is calculated as the ratio of contig mapping depth to global mapping depth.

Newbler Repeats				SPAdes Repeats			
	Contig Length	Mapping Depth	Estimated Copy number		Contig Length	Mapping Depth	Estimated Copy number
contig00023*	81,744	99	1.71	NODE_11*	81,973	99	1.74
contig00026	32,393	114	1.97	NODE_14*	13,247	306	5.37
contig00028*	13,120	307	5.29	NODE_15	5,312	115	2.02
contig00031	3,394	311	5.36	NODE_16	5,192	309	5.42
contig00035	3,314	112	1.93	NODE_18	3,775	69	1.21
contig00036	2,346	148	2.55	NODE_19	2,891	111	1.95
contig00038	2,263	99	1.71	NODE_20	2,346	148	2.60
contig00032	1,771	301	5.19	NODE_24	682	78	1.37
contig00008	1,079	95	1.64	NODE_25	514	123	2.16
				NODE_26	500	74	1.30
				NODE_27	452	106	1.86

Table 4.3. Assembly Combination Results
Results of combining assemblies using CISA or Mix.

	Method	Repeat Contigs Included	Bridge Contigs	Max Contig Length	N50 Contig Length	Number of Contigs	Total Length of Contigs
	Newbler Assembly			769,200	338,867	39	6,286,279
	SPAdes Assembly			1,041,000	853,530	29	6,297,403
C	CISA	N	N	1,695,000	890,291	10	6,154,209
C-R	CISA	Y	N	1,695,000	890,556	17	6,297,323
C-B	CISA	N	Y	1,698,000	1,043,751	10	6,232,656
C-R+B	CISA	Y	Y	1,698,000	1,057,801	10	6,319,052
M	Mix	N	N	1,695,000	890,556	14	6,185,412
M-R	Mix	Y	N	1,695,000	890,556	17	6,296,326
M-B	Mix	N	Y	1,697,000	893,756	17	7,188,203
M-R+B	Mix	Y	Y	1,697,000	857,314	20	7,541,383

Table 4.4. Repeat Incorporation
Repeat incorporation with CISA and Mix assembly combiner programs. Plasmids are indicated with an asterisk (*).

Repeat	Estimated Copy Number	Length	**C**	**C-R**	**C-B**	**C-R+B**	**M**	**M-R**	**M-B**	**M-R+B**
contig00028*	5.29	13,120	0	1	0	1	0	1	0	2
contig00023*	1.71	81,744	0	1	0	1	0	1	0	2
contig00026	1.97	32,393	0	1	1	1	1	1	1	1
contig00035	1.93	3,314	0	1	1	1	0	1	0	1
contig00036	2.55	2,346	0	1	2	2	0	1	1	1
contig00038	1.71	2,263	1	1	1	1	1	1	1	1
NODE_24	1.37	682	1	2	2	2	1	1	1	1
contig00031	5.36	3,394	0	1	4	4	0	1	0	1
contig00032	5.19	1,771	0	1	3	3	0	1	2	2

References

1. Paszkiewicz K, Studholme DJ (2010) De novo assembly of short sequence reads. Brief Bioinform. doi: 10.1093/bib/bbq020

2. Klassen JL, Currie CR (2012) Gene fragmentation in bacterial draft genomes: extent, consequences and mitigation. BMC Genomics 13:14. doi: 10.1186/1471-2164-13-14

3. Koren S, Harhay GP, Smith TP, et al. (2013) Reducing assembly complexity of microbial genomes with single-molecule sequencing. Genome Biol 14:R101. doi: 10.1186/gb-2013-14-9-r101

4. Pedró L, Baños RC, Aznar S, et al. (2011) Antibiotics shaping bacterial genome: deletion of an IS91 flanked virulence determinant upon exposure to subinhibitory antibiotic concentrations. PLoS One 6:e27606. doi: 10.1371/journal.pone.0027606

5. Nalbantoglu U, Sayood K, Dempsey MP, et al. (2010) Large direct repeats flank genomic rearrangements between a new clinical isolate of *Francisella tularensis* subsp. tularensis A1 and Schu S4. PLoS One 5:e9007. doi: 10.1371/journal.pone.0009007

6. Phillippy AM, Schatz MC, Pop M (2008) Genome assembly forensics: finding the elusive mis-assembly. Genome Biol 9:R55. doi: 10.1186/gb-2008-9-3-r55

7. Wetzel J, Kingsford C, Pop M (2011) Assessing the benefits of using mate-pairs to resolve repeats in de novo short-read prokaryotic assemblies. BMC Bioinformatics 12:95. doi: 10.1186/1471-2105-12-95

8. Sheppard AE, Poehlein A, Rosenstiel P, et al. (2013) Complete Genome Sequence of *Bacillus thuringiensis* Strain 407 Cry-. Genome Announc 1:5–6. doi: 10.1128/genomeA.00158-12

9. Schwartz DC, Li X, Hernandez LI, et al. (1993) Ordered restriction maps of *Saccharomyces cerevisiae* chromosomes constructed by optical mapping. Science 262:110–4.

10. Assefa S, Keane TM, Otto TD, et al. (2009) ABACAS: algorithm-based automatic contiguation of assembled sequences. Bioinformatics 25:1968–9. doi: 10.1093/bioinformatics/btp347

11. Soueidan H, Maurier F, Groppi A, et al. (2013) Finishing bacterial genome assemblies with Mix. BMC Bioinformatics 14:S16. doi: 10.1186/1471-2105-14-S15-S16

12. Nijkamp J, Winterbach W, van den Broek M, et al. (2010) Integrating genome assemblies with MAIA. Bioinformatics 26:i433–9. doi: 10.1093/bioinformatics/btq366

13. Lin S-H, Liao Y-C (2013) CISA: contig integrator for sequence assembly of bacterial genomes. PLoS One 8:e60843. doi: 10.1371/journal.pone.0060843

14. Yao G, Ye L, Gao H, et al. (2012) Graph accordance of next-generation sequence assemblies. Bioinformatics 28:13–6. doi: 10.1093/bioinformatics/btr588

15. Sommer DD, Delcher AL, Salzberg SL, Pop M (2007) Minimus: a fast, lightweight genome assembler. BMC Bioinformatics 8:64. doi: 10.1186/1471-2105-8-64

16. Vicedomini R, Vezzi F, Scalabrin S, et al. (2013) GAM-NGS: genomic assemblies merger for next generation sequencing. BMC Bioinformatics 14 Suppl 7:S6. doi: 10.1186/1471-2105-14-S7-S6

17. Magoc T, Pabinger S, Canzar S, et al. (2013) GAGE-B: an evaluation of genome assemblers for bacterial organisms. Bioinformatics 29:1718–25. doi: 10.1093/bioinformatics/btt273

18. Ono Y, Asai K, Hamada M (2013) PBSIM: PacBio reads simulator--toward accurate genome assembly. Bioinformatics 29:119–21. doi: 10.1093/bioinformatics/bts649

19. English AC, Richards S, Han Y, et al. (2012) Mind the gap: upgrading genomes with Pacific Biosciences RS long-read sequencing technology. PLoS One 7:e47768. doi: 10.1371/journal.pone.0047768

20. De Gelder L, Vandecasteele FPJ, Brown CJ, et al. (2005) Plasmid donor affects host range of promiscuous IncP-1beta plasmid pB10 in an activated-sludge microbial community. Appl Environ Microbiol 71:5309–17. doi: 10.1128/AEM.71.9.5309-5317.2005

21. Hunter SS, Yano H, Loftie-Eaton W, et al. (2014) Draft genome sequence of *Pseudomonas moraviensis* R28-S. Genome Announc. doi: 10.1128/genomeA.00035-14

22. Magoc T, Salzberg SL (2011) FLASH: fast length adjustment of short reads to improve genome assemblies. Bioinformatics 27:2957–63. doi: 10.1093/bioinformatics/btr507

23. Bankevich A, Nurk S, Antipov D, et al. (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol 19:455–77. doi: 10.1089/cmb.2012.0021

24. Margulies M, Egholm M, Altman WE, et al. (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376–80. doi: 10.1038/nature03959

25. Gurevich A, Saveliev V, Vyahhi N, Tesler G (2013) QUAST: quality assessment tool for genome assemblies. Bioinformatics 29:1072–5. doi: 10.1093/bioinformatics/btt086

26. Li H, Handsaker B, Wysoker A, et al. (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25:2078–9. doi: 10.1093/bioinformatics/btp352

27. Kent WJ (2002) BLAT--the BLAST-like alignment tool. Genome Res 12:656–64. doi: 10.1101/gr.229202. Article published online before March 2002

28. Delcher AL, Phillippy A, Carlton J, Salzberg SL (2002) Fast algorithms for large-scale genome alignment and comparison. Nucleic Acids Res 30:2478–83.

Supplementary Material

Table S4.1. Detailed Assembly Results

Results of assembly using Newber v2.8 and SPAdes v3.0 on the same set of Illumina and 454 reads. Repeat contigs are indicated in bold and plasmids are indicated with *. Average mapping depth is reported, and estimated copy number is calculated as the ratio of contig mapping depth to global mapping depth.

	Newbler		
Contig ID	Contig Length	Mapping Depth	Estimated Copy number
contig00001	306,531	54	0.93
contig00002	101,300	55	0.95
contig00003	769,154	57	0.98
contig00004	30,718	53	0.91
contig00005	709,669	51	0.88
contig00006	727,235	59	1.02
contig00007	2,358	53	0.91
contig00008	**1,079**	**95**	**1.64**
contig00009	272,049	55	0.95
contig00010	413,022	55	0.95
contig00011	407,275	54	0.93
contig00012	273,168	57	0.98
contig00013	129,282	58	1.00
contig00014	338,867	58	1.00
contig00015	272,350	58	1.00
contig00016	265,376	59	1.02
contig00017	231,864	57	0.98
contig00018	189,634	57	0.98
contig00019	155,198	56	0.97
contig00020	147,285	49	0.84
contig00021	124,689	58	1.00
contig00022	109,535	51	0.88
contig00023*	**81,744**	**99**	**1.71**
contig00024	81,652	55	0.95
contig00025	36,591	55	0.95
contig00026	**32,393**	**114**	**1.97**
contig00027	16,183	65	1.12
contig00028*	**13,120**	**307**	**5.29**
contig00029	12,292	53	0.91
contig00030	11,777	56	0.97
contig00031	**3,394**	**311**	**5.36**
contig00032	**1,771**	**301**	**5.19**
contig00033	3,525	57	0.98
contig00034	3,459	65	1.12
contig00035	**3,314**	**112**	**1.93**
contig00036	**2,346**	**148**	**2.55**
contig00037	2,293	50	0.86
contig00038	**2,263**	**99**	**1.71**
contig00039	511	2	0.03

	SPAdes		
Contig ID	Contig Length	Mapping Depth	Estimated Copy number
NODE_01	1,040,851	56	0.98
NODE_02	977,721	54	0.95
NODE_03	890,556	51	0.89
NODE_04	853,530	59	1.04
NODE_05	668,171	58	1.02
NODE_06	594,296	58	1.02
NODE_07	422,131	61	1.07
NODE_08	360,200	58	1.02
NODE_09	155,459	56	0.98
NODE_10	122,459	58	1.02
NODE_11*	**81,973**	**99**	**1.74**
NODE_12	62,782	58	1.02
NODE_13	23,817	54	0.95
NODE_14*	**13,247**	**306**	**5.37**
NODE_15	**5,312**	**115**	**2.02**
NODE_16	**5,192**	**309**	**5.42**
NODE_17	4,800	58	1.02
NODE_18	**3,775**	**69**	**1.21**
NODE_19	**2,891**	**111**	**1.95**
NODE_20	**2,346**	**148**	**2.60**
NODE_21	2,118	51	0.89
NODE_22	793	2	0.04
NODE_24	**682**	**78**	**1.37**
NODE_25	**514**	**123**	**2.16**
NODE_26	**500**	**74**	**1.30**
NODE_27	**452**	**106**	**1.86**
NODE_27	382	25	0.44
NODE_28	315	18	0.32
NODE_29	128	8	0.14

Summary Statistics		
	Newbler	SPAdes
Total Contigs > 500 bp	39	24
N50	338,867	853,530
Mean Length (bp)	161,200	217,200
Total Length (bp)	6,286,279	6,297,403
Repeats > 500 bp	9	9
Largest repeat (bp)	32,393	5,312

Chapter 5

Draft genome sequence of *Pseudomonas moraviensis* R28-S

Samuel S. Hunter[a,b], Hirokazu Yano[a,b], Wesley Loftie-Eaton[a,b], Julie Hughes[a,b], Leen De Gelder[a,b], Pieter Stragier[c], Paul De Vos[c], Matthew L. Settles[a,b], Eva M. Top[a,b]

[a]Department of Biological Sciences, University of Idaho, Moscow, Idaho, USA
[b]Institute for Bioinformatics and Evolutionary Studies (IBEST), University of Idaho, Moscow, Idaho, USA
[c]Laboratory of Microbiology, Ghent University, Ghent, Belgium

Publication

Published January 2014 in the journal Genome Announcements. This is an open-access article distributed under the terms of the *Creative Commons Attribution 3.0 Unported license* (http://creativecommons.org/licenses/by/3.0/) and requires no permission to reproduce here. This article can be access with the following doi: 10.1128/genomeA.00035-14.

Abstract

We report the draft genome sequence of *Pseudomonas moraviensis* R28-S, isolated from the municipal wastewater treatment plant of Moscow, ID. The strain carries a native mercury resistance plasmid, poorly maintains introduced IncP-1 antibiotic resistance plasmids, and has been useful for studying the evolution of plasmid host range and stability.

Genome Announcement

Pseudomonas moraviensis R28 is a member of the *Gammaproteobacteria* and was originally reported as *Pseudomonas koreensis* R28. The strain was isolated from activated sludge of the municipal wastewater treatment plant in Moscow, ID, as a transconjugant after a plate mating of a sludge sample with a donor plasmid pB10::rfp. The transconjugants were selected on defined aerobic basal (DAB) medium supplemented with succinate, acetate, and citrate and the antibiotics tetracycline (10 mg/liter) and streptomycin (50 mg/liter) [1]. In the laboratory, it has been a useful strain for studying the stability and evolution of broad-host-range multidrug resistance plasmids [1–3]. The first isolate of the species *P. moraviensis* was collected from oil-polluted soil in the Czech Republic and was shown to hydrolyze diverse carbohydrates and utilize an impressive array of substrates [4]. Strain R28-S, a streptomycin-resistant mutant of R28, was identified as a member of this species via an in-house four-gene-based (atpA, glnA, rpoB, and rpoD) multilocus sequence analysis (MLSA) scheme, which for each gene undoubtedly showed the highest match with the type strain *P. moraviensis* LMG 24280.

The genome of *P. moraviensis* R28-S was sequenced using a whole-genome shotgun approach, with paired 150-bp reads generated on the MiSeq (Illumina) and 454 (Roche) sequencing platforms. The sequencing adapters and low-quality bases were trimmed using a custom script, and the reads were assembled using Newbler version 2.6. A total of 36 contigs >500 bp were produced. Of these, the largest is 815,593 bp and the N50 contig size is 462,409 bp. The assembled contigs were ordered and oriented using a whole-genome map produced by OpGen optical mapping MapIt services. The optical mapping results were corroborated by aligning R28-S contigs against the closely related and so-called *Pseudomonas fluorescens* Pf0-1 (accession no. NC_007492) genome using r2cat [5]. Small contigs that could not be scaffolded with the optical map were placed using these alignments. Additional gaps were then closed using the program GapFiller [6], and the paired Illumina reads resulted in a final assembly consisting of 12 contigs with a total length of 6,226,470 bp (including estimated gap sizes).

Included in the set of contigs was a native 81,846-bp plasmid, pR28. The replication initiator gene (repA) and origin of replication gene (oriV) of pR28 bear 89 and 84% nucleotide identities, respectively, to that of the IncP-9θ plasmid pSVS15 isolated from *Pseudomonas putida* [7]. Although many IncP-9 plasmids are self-transferable, pR28 does not encode a full conjugative system. Its genome contains multiple transposons, one of which encodes resistance to mercury. Based on its read coverage, its copy number is estimated at 2/cell (1.9× for each chromosome copy).

Nucleotide sequence accession numbers.

This whole-genome shotgun project has been deposited at GenBank under the accession no. AYMZ00000000. The version described in this paper is version AYMZ00000000.1. Strain R28-S is available from the LMG culture collection as LMG 28150 (http://bccm.belspo.be/about/lmg.php).

References

1. De Gelder L, Vandecasteele FPJ, Brown CJ, et al. (2005) Plasmid donor affects host range of promiscuous IncP-1beta plasmid pB10 in an activated-sludge microbial community. Appl Environ Microbiol 71:5309–17. doi: 10.1128/AEM.71.9.5309-5317.2005

2. De Gelder L, Ponciano JM, Joyce P, Top EM (2007) Stability of a promiscuous plasmid in different hosts: no guarantee for a long-term relationship. Microbiology 153:452–63. doi: 10.1099/mic.0.2006/001784-0

3. Sota M, Yano H, Hughes JM, et al. (2010) Shifts in the host range of a promiscuous plasmid through parallel evolution of its replication initiation protein. ISME J 4:1568–80. doi: 10.1038/ismej.2010.72

4. Tvrzová L, Schumann P, Spröer C, et al. (2006) Pseudomonas moraviensis sp. nov. and Pseudomonas vranovensis sp. nov., soil bacteria isolated on nitroaromatic compounds, and emended description of Pseudomonas asplenii. Int J Syst Evol Microbiol 56:2657–63. doi: 10.1099/ijs.0.63988-0

5. Husemann P, Stoye J (2010) r2cat: synteny plots and comparative assembly. Bioinformatics 26:570–1. doi: 10.1093/bioinformatics/btp690

6. Boetzer M, Pirovano W (2012) Toward almost closed genomes with GapFiller. Genome Biol 13:R56. doi: 10.1186/gb-2012-13-6-r56

7. Sevastsyanovich YR, Krasowiak R, Bingle LEH, et al. (2008) Diversity of IncP-9 plasmids of Pseudomonas. Microbiology 154:2929–41. doi: 10.1099/mic.0.2008/017939-0

Chapter 6

Conclusions and Future Directions

In this dissertation, I have presented three tools that are useful for the collection or analysis of data from nucleic acid sequences. The VChip, a microarray platform for studying DNA or RNA from the vaginal microbiota, ARC, a tool for reference seeded assembly of homologous sequences, and StopGap, an approach for closing gaps in assemblies using PacBio reads.

Insights

The genesis of ARC was based on two important insights. The first is that homologous sequences from different species often have highly conserved regions which remain largely unchanged even when the rest of the sequence is much more divergent. An initial “sloppy” mapping step can take advantage of these regions to recruit reads. Following up this initial recruitment stage with assembly and more stringent mapping in subsequent iterations allows ARC to address the trade-off between sensitivity and specificity which is inherent in traditional read mapping, and incorporate reads from more divergent loci into the final assembly. The second insight is that it is often not necessary or efficient to process an entire HTS dataset at once. This makes divide and conquer strategies an obvious choice for processing this type of dataset. In essence, this is what ARC is, an approach for simplifying the difficult problem of sequence assembly by splitting a large set of reads into multiple small subsets, each of which is much easier to assemble than the large set would have been.

High Throughput Sequencing

HTS technologies have developed rapidly since the commercial introduction of pyrosequencing in 2004. From this point until at least 2012, expense has dropped rapidly, with the cost in base-pairs per dollar cut in half every 5 months [1]. This precipitous drop in prices, driven by rapid increases in throughput, has allowed for an equally impressive amount of sequence data to be collected. Currently 2,295 trillion (2.3 quadrillion) bases have been deposited in the Short Read Archive, a National Institutes of Health database that stores raw sequencing data generated by high throughput sequencing (http://www.ncbi.nlm.nih.gov/Traces/sra/). This drive for ever higher throughput continues, and Illumina has claimed victory in the quest for a “$1000 human genome” in their January 2014 announcement of the HiSeq X Ten sequencing cluster. Already the Roche 454 platform has succumbed to the rapid pace of sequencer technology development. Despite having longer overall reads, the high cost of a sequencing run and low relative yield make it a poor choice compared to options from Illumina and Ion Torrent. Although throughput is often considered the most important consideration in sequencing technology innovation, read length is a limiting factor in the ability to assemble sequence data and detect large structural variants such as inversions and other rearrangements within a genome.

In fact, short read length is the root cause of most of the complexity involved in HTS data analysis. Short reads necessitate high coverage to ensure that all (or most) regions of the genome are properly represented. The requirement for high coverage leads to large datasets that are difficult to analyze. This is partially due to the sheer size of the data sets, but also because of the complex algorithms required to resolve or remove errors within reads, detect variants, and address the challenges of assembly and mapping. The idealized sequencer would produce a single, error free, long read for each chromosome or other genetic element within a sample, requiring minimal further processing beyond sequencing. While it is impossible to say whether the technology necessary to produce such reads will ever exist, some emerging “3rd generation” technologies are making major improvements on read length. Single Molecule Real Time sequencing (SMRT) from Pacific Biosciences has slowly but steadily improved both length and throughput, and actual data from Oxford Nanopore's long awaited implementation of nanopore DNA sequencing technology was presented at the February 2014 Advances in Genome Biology & Technology (AGBT) meeting. Few details are available for the Oxford Nanopore data but the cost for PacBio data is still too high for wide-spread adoption by the human genome resequencing community. However, the landscape of *de novo* bacterial genome assembly is already starting to change with the latest advancements from PacBio.

This leaves software packages such as ARC, StopGap, and the hundreds of other tools for analyzing high throughput sequencing data in a strange position. These tools were designed with the currently dominant sequencing platform (Illumina) in mind, and have great utility for the large number of unanalyzed datasets that were produced in the rush to become part of the HTS revolution. However, future improvements in throughput and read length for 3rd generation sequencing platforms could rapidly render many of these tools obsolete.

The VChip

The VChip contains probes for gene sequences from 313 bacterial strains representing 184 bacterial species as well as 716 selected human genes. The microbial communities living on and in the human body have emerged as a major new frontier in research and medicine, particularly since investments from the NIH in the Human Microbiome Project have allowed for larger scale data collection and characterization efforts than had previously been possible. The interplay between these organisms and the human host appear to directly impact health, leading to illness when disturbed as in the case of bacterial vaginosis. The VChip represents an embodiment of this interplay, containing probes for both bacterial genes and a panel of human genes involved in a variety of functions that may be involved in maintaining, or responding to changes in the vaginal bacterial community. Collecting information about simultaneous changes in gene expression in the bacteria and human host may lead to a better understanding of how these system work together, provide insights into why communities change over time, and identify the drivers of disease. Microarrays also have a well established utility for measuring effects of treatment. A platform such as the VChip may provide useful insights that provide a basis for the development of methods to manipulate the microbiota (or host) to re-establish a healthy state following a disturbance.

The VChip represents a use of what might arguably be the pinnacle of microarray technology: custom, programmable, UV light directed synthesis of 60 bp oligonucleotides at the incredibly high density of 1.4 million spots in triplicate on a standard glass microscope slide. While microarrays have become a standard tool in many biological laboratories, their use is in decline due to the emergence and rapid improvements in HTS. In response to the increased popularity of sequencing based methods, Roche NimbleGen has discontinued its microarray operations, making it impossible to purchase the VChip in its current form. Other options for printing the set of probes developed for the VChip are available however, including the Custom Gene Expression Microarrays from Agilent Technologies (www.genomics.agilent.com) and MyGeneChip^{TM} custom arrays by Affymetrix (www.affymetrix.com).

References

(The following references cover the Introduction and Conclusions and Future Directions sections only.)

1. Crick F (1970) Central Dogma of Molecular Biology. Nature 227:561–563.

2. Jou WM, Haegeman G, Ysebaert M, Fiers W (1972) Nucleotide Sequence of the Gene Coding for the Bacteriophage MS2 Coat Protein. Nature 237:82–88. doi: 10.1038/237082a0

3. Sanger F, Nicklen S, Coulson AR (1977) DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci U S A 74:5463–7.

4. Springer M (2006) Applied Biosystems : Celebrating 25 Years of Advancing Science. Am. Lab. News

5. Augenlicht LH, Wahrman MZ, Halsey H, et al. (1987) Expression of cloned sequences in biopsies of human colonic tissue and in colonic carcinoma cells induced to differentiate in vitro. Cancer Res 47:6017–21.

6. Schena M, Shalon D, Davis RW, Brown PO (1995) Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. Science (80-) 270:467–470. doi: 10.1126/science.270.5235.467

7. Fodor SPA, Read JL, Pirrung MC, et al. (1991) Light-directed, spatially addressable parallel chemical synthesis. Science 251:767–73.

8. Nuwaysir EF, Huang W, Albert TJ, et al. (2002) Gene expression analysis using oligonucleotide arrays produced by maskless photolithography. Genome Res 12:1749–55. doi: 10.1101/gr.362402

9. Margulies M, Egholm M, Altman WE, et al. (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376–80. doi: 10.1038/nature03959

10. Breitling R (2006) Biological microarray interpretation: the rules of engagement. Biochim Biophys Acta 1759:319–27. doi: 10.1016/j.bbaexp.2006.06.003

11. Bradnam KR, Fass JN, Alexandrov A, et al. (2013) Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. Gigascience 2:10. doi: 10.1186/2047-217X-2-10

12. Magoc T, Pabinger S, Canzar S, et al. (2013) GAGE-B: an evaluation of genome assemblers for bacterial organisms. Bioinformatics 29:1718–25. doi: 10.1093/bioinformatics/btt273

13. Kim SY, Speed TP (2013) Comparing somatic mutation-callers: beyond Venn diagrams. BMC Bioinformatics 14:189. doi: 10.1186/1471-2105-14-189

14. Li H (2012) Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. Bioinformatics 28:1838–44. doi: 10.1093/bioinformatics/bts280

15. Ravel J, Gajer P, Abdo Z, et al. (2011) Vaginal microbiome of reproductive-age women. Proc Natl Acad Sci U S A 108 Suppl :4680–7. doi: 10.1073/pnas.1002611107

16. Gajer P, Brotman RM, Bai G, et al. (2012) Temporal dynamics of the human vaginal microbiota. Sci Transl Med 4:132ra52. doi: 10.1126/scitranslmed.3003605

17. Mamanova L, Coffey AJ, Scott CE, et al. (2010) Target-enrichment strategies for next-generation sequencing. Nat Methods 7:111–8. doi: 10.1038/nmeth.1419

18. Phillippy AM, Schatz MC, Pop M (2008) Genome assembly forensics: finding the elusive mis-assembly. Genome Biol 9:R55. doi: 10.1186/gb-2008-9-3-r55

19. Klassen JL, Currie CR (2012) Gene fragmentation in bacterial draft genomes: extent, consequences and mitigation. BMC Genomics 13:14. doi: 10.1186/1471-2164-13-14

20. Stein LD (2010) The case for cloud computing in genome informatics. Genome Biol 11:207. doi: 10.1186/gb-2010-11-5-207