UPPSALA
UNIVERSITET

MSc BIOINF 11 001

Whole genome sequencing and assembly of an avian genome, the European crow *Corvus corone* spec.

A reduced representation library based approach

Nagarjun Vijay

Degree project in bioinformatics, 2011
Examensarbete i bioinformatik 45 hp till masterexamen, 2011
Biology Education Centre and Dept. of Evolutionary biology , EBC, Uppsala University
Supervisor: Dr Jochen Wolf

1 Abstract

Whole genome sequencing using novel next-generation technologies has over the last few years become a viable cost-effective alternative to first generation sequencing methodologies. It holds great promise to obtain large scale sequence information for evolutionary/ecological models where previously only scarce genetic resources were available. However, assemblies generated using just next-generation technologies are highly fragmented and have many gaps which puts a premium on improving genome assembly approaches.

Using in-silico simulations we compared if reduced representation library (RRL) based genome assembly provides more contiguous and accurate assemblies than the whole genome shotgun (WGS) approach in an avian model organism. The RRL approach partitions the genome based on the size of fragments obtained after digesting genomic DNA with restriction enzymes. To obtain overlapping reads throughout the genome, DNA is digested with two different restriction enzymes separately to obtain four differently sized partitions for each enzyme. Each of these partitions is separately sequenced with paired end reads which are assembled individually. These individually assembled partitions are 'meta-assembled' based on overlaps between the two sets of assemblies. In contrast, the WGS approach involves random fragmentation of genomic DNA followed by sequencing and assembling paired end reads from these random fragments.

The most contiguous assemblies were obtained with the RRL strategy at 25X coverage with 100 BP paired end reads with 400 BP insert size. The RRL assembly was more contiguous than the WGS assembly. We use this approach as a pilot study for determining a sequencing strategy for the crow genome and further looked into its feasibility and effectiveness by constructing reduced representation libraries in the laboratory.

Whole genome sequencing and assembly of an avian genome, the European crow *Corvus corone* spec.

A reduced representation library based approach

- Popular science summary

Nagarjun Vijay

The complete hereditary information of an organism which mostly decides the various characters of an organism are stored in its genome in the form of different combinations of nucleotide bases in the DNA. To understand the various characters of an organism, genes and functional elements responsible for these features it is useful to sequence its genome. Sequencing involves finding the order in which the nucleotide bases are organized. Various sequencing methods can determine the order of the bases in a stretch of 100 to a maximum of 1000 bases. Hence, for organisms with large genomes it is not possible to sequence the entire genome directly.

The maximum number of consecutive bases that can be sequenced is limited to a maximum of 1.5 KB. To overcome this limitation 'shotgun' genome sequencing has been utilised. A sidewalk will eventually be completely covered by randomly falling raindrops. Similarly, the entire genome can be covered by randomly sequencing smaller fragments of the genome. This is done by first breaking down DNA into a number of random fragments of length suitable for sequencing. These fragments are then sequenced individually. Enough fragments are sequenced to have covered the genome multiple times. Sequenced pieces of the genome are put together into a single continuous sequence of DNA using a computer program called “Genome assembler”. The genome assembler looks for overlapping regions between the sequenced fragments and makes use of this information to place the different fragments with respect to each other. This method of genome assembly is known as ‘shotgun’ genome sequencing.

In this project we tested the benefits of an alternative method for genome assembly called reduced representation library approach. In this approach the genome is first partitioned into smaller reproducible fractions which are then individually subjected to shotgun genome sequencing. These individually assembled genomes are then put together to obtain a complete genome assembly. Partitioning of the genome into smaller fractions is expected to reduce the number of incorrect assemblies and allow for faster assembling of the data. Thereby using in-silico simulation on the zebra finch genome the reduced representation library approach has been found to provide assemblies which are more contiguous than those obtained from shotgun genome sequencing.

Degree project in bioinformatics, 2011
Examensarbete i bioinformatik 45 hp till masterexamen, 2011
Biology Education Centre and Dept. of Evolutionary biology, EBC, Uppsala University
Supervisor: Dr Jochen Wolf

“Hyperboloids of wondrous Light

Rolling for aye through Space and Time

Harbour those Waves which somehow Might

Play out God's holy pantomime”

-A Turning

Contents

2 Introduction

By the turn of the century, DNA sequencing had scaled up remarkably and the publishing of the human genome in 2001 heralded the post-genomic era (Venter et al., 2001). Being able to read genomes has helped us understand the functional aspects of genes and regulatory networks (Anna Sharman et al., 2001). Sequencing of multiple closely related organisms has allowed for the comparative analyses of genomes which is very useful in evolutionary studies. While most of the manually maintained genomes available now have been produced with the di-deoxy chain termination method, several novel methods of DNA sequencing have been developed in subsequent decades (Michael L. M et al., 2010). Improvements in sequencing technology have made it possible to sequence larger genomes at lower costs. This now opens up the possibility to investigate organisms where only scarce genetic data is available.

While technically now it is possible to produce multiple fold coverage of a genome using the latest sequencing technology, still many algorithmic and computational challenges in obtaining a fully assembled high quality draft remain (Sante et al., 2001, Miller et al., 2010). We will here first review the most common WGS approach and then describe a variant (RRL) that will be explored in more detail. The aim of this study was to compare the two with respect to contiguity, accuracy, recovery and include some experimental tests on library preparation in the laboratory.

2.1 Whole genome shotgun assembly

Sequencing or reading the entire genome is limited by the maximum length of 'continuous' DNA that can be sequenced. Smaller genomes could be sequenced as smaller parts and put together into a genetic map based on the overlap shared by the sequenced regions. For example, the "Epstein-Barr Virus Genome" was compiled together using previously published features that had been annotated (Baer et al., 1984). However, sequencing larger genomes is more challenging. The first larger genome where this could be accomplished was the one of the bacteria Haemophilus influenzae.

The genome of the bacterium Haemophilus influenzae was sequenced and assembled by random sequencing followed by assembly to obtain the first complete genome sequence of a free living organism (Fleischmann et al., 1995). This strategy of sequencing whole genomes by randomly fragmenting DNA into smaller fragments which are sequenced separately is known as shotgun sequencing. In the whole genome shotgun assembly strategy (Figure 1), genomic DNA is randomly broken into numerous smaller pieces. These smaller pieces are sequenced to obtain reads. Large numbers of such reads are generated so as to obtain overlapping reads. Based on the overlap shared between the different reads, computer programs known as 'assemblers' build these reads into a continuous sequence known as a 'contig'.

Finding the correct overlap between millions of short sequence reads is a difficult algorithmic task that has so far not found an analytical solution. Heuristic methods which

utilise experience based methods for speeding up solving of problems are continuously being developed (Daniel et al., 2004, Scheibye-Alsing et al., 2009, Miller et al., 2010) to improve the quality of the assemblies. The problem is exacerbated by errors in the process of reading the genomic sequence as a result of limitations in the sequencing technology, polymorphism in genomic DNA of diploid organisms; repeat regions which cannot be distinguished or placed in the correct location make the process of genome sequencing and assembly more complicated in practice.

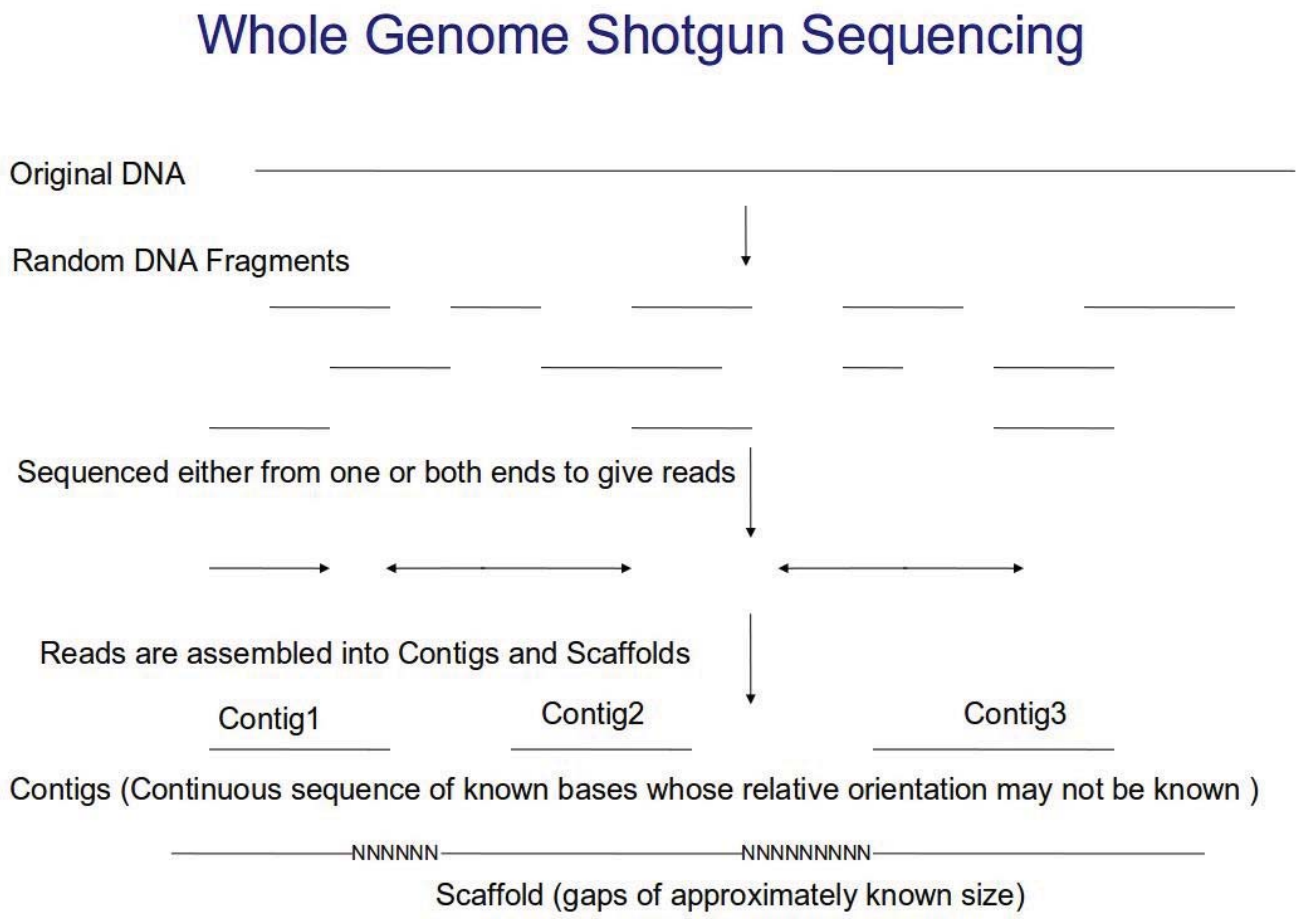

Figure 1: Flowchart of shotgun Genome sequencing

A way to improve the assembly across repeat regions which constitute a particular challenge (Daniel R et al., 2010) and to join contigs with gaps in between is the use of fragments sequenced from both ends with gaps of a known size in between. These special types of reads are generated by randomly fragmenting DNA, size-selecting and sequencing it from both ends and are usually referred to as ‘mate pairs' or jump libraries (Pop et al., 2004, Sante et al., 2010). 'Mate pairs' provide additional information about the relative position of the pair of reads which can be very useful information in resolving repeat regions. Different sizes of DNA fragments (generally 2, 5, 10, 50, 100 and 150 kb) are selected and sequenced from both ends. These 'mate pairs' of different sizes are used by the 'assembler' to position the reads based on estimates of the distance between the read pair. Using the information available in the reads and 'mate pairs', the computer program will organize 'contigs' into 'scaffolds' based on the relative position of the contigs based on connections between mate pairs. The gaps present in the scaffolds are filled by a process known as genome finishing. Gaps smaller than 10kb are covered by PCR amplification and sequencing of the region. Gaps bigger than 10kb are generally filled using other strategies such as BAC clone sequencing. BAC clones target a specific reduced region of the genome that can be assembled without having to resolve reads from other regions of the genome.

2.1.1 Traditional sanger sequencing

The chain termination method (Sanger et al., 1975) of DNA sequencing has been widely used due to its reliability and relative ease of use. In the chain termination method single stranded DNA sample is elongated in a DNA replication reaction using a DNA primer, DNA polymerase, deoxynucleotidephosphates (dNTPs) and modified deoxynucleotidephosphates

(dNTPs) which lack a 3' OH group (required for establishing phosphodiester bonds between adjacent nucleotides). DNA elongation is interrupted when a modified dNTP is incorporated. Each of the many hundred different reactions is interrupted at a different base along the DNA sequence. Radioactively or fluorescently labelled modified dNTPs can be visualized after denaturing into single stranded DNA followed by size separation. Various dNTP's A, T, G and C is labelled with a different colored dye to be able to identify bases present at different positions along the sequence.

Reads (continuous sequence of DNA) of about 1kb can be sequenced with the traditional Sanger sequencing methodology. Data obtained from this method is of poor quality at both ends of the sequenced read. Hence, quality values are generated for the sequenced reads to aid the assembly process. Depending on the software used for post processing of reads based on these quality scores, error rates of 0.001 to more than 1% have been reported (Hoff 2009).

2.1.2 NGS: Next-generation sequencing

Subsequent to sequencing of the Human genome, the need for a cost effective and fast method of sequencing has led to the development of various new sequencing technologies. These newer methods are collectively known as next-generation sequencing (NGS) (Michael L. M 2010) and provide a cost effective alternative to traditional Sanger sequencing. NGS generates shorter reads of 25 base pair up to 400 base pairs in length (Paul Flicerk et al., 2009) depending on the sequencing technology used.

Sequencing Technology	Cost per Mb of DNA sequence	Cost per human sized Genome	Cost estimated on below Date	Reference
First generation platforms (Sanger and other capillary based methods)	$397.09	$7,147,571	October-2007	Wetterstrand KA (Reference 41)
Second-generation platforms (454, SOLID, Illumina, etc)	$102.13	$3,063,820	January-2008	Wetterstrand KA (Reference 41)
Second-generation platforms (454, SOLID, Illumina, etc)	$0.23	$20,963	January-2011	Wetterstrand KA (Reference 41)

Table 1: Cost comparison of DNA sequencing

Due to the significant cost benefits (table 1) associated with NGS technologies, they have become a popular tool for not only genome re-sequencing but also de novo genome assembly. Its popularity will arguably even increase, since sequencing costs have been decreasing drastically since the introduction of the second generation sequencing methods and are expected to continue to do so for a while.

2.1.3 Genome assembly and its challenges

Advances in sequencing technology have been reflected in the methods for assembling genomes. Before the advent of NGS technologies assembly programs relied on "overlap-layout-consensus" methods for genome assembly. Long read lengths and a error rate of 1% made it possible to assemble genomes based on overlap graphs (Denisov et al., 2008).

Shorter read lengths, exponential increase in the number of reads and higher error rates are not suitable for these algorithms. Hence, various assembly programs (Daniel R et al., 2010) based on de Bruijn Graphs have been created. A de Bruijn graphs is a type of directed graph used to represent overlaps between sequences of symbols.

These approaches treat the data as words of k number of nucleotide bases or k-mers instead of reads. Higher redundancy generated by NGS methods is handled by this method as reads are broken into 'k-mers' that make up each read. The de Bruijn graph is constructed by representing a series of overlapping k-mers as a node in the graph. After construction of the graph and 'hashing' the reads based on k-mer's that make up the read, the graph is simplified without losing data using string graph based methods. A string graph is an intersection graph of curves in the plane, where each curve is called a string. Representing the intersection of the different k-mer's allows the use of string graph simplification methods. Simplification of the graph expedites the subsequent error removal, repeat resolution and scaffolding steps. Improvements in error handling and repeat resolution algorithms have made it possible to assemble whole genomes using just short-read sequencing (Ruiqiang Li et al., 2010). The overlap consensus based approach, on the other hand, relies entirely on finding overlaps between reads for performing the assembly. Presence of redundant information makes the overlap based method cumbersome as memory requirements are very high.

Despite of the difficulties associated with short read data, sequencing of Giga base sized genomes using short-read sequencing has been demonstrated with considerable success (Ruiqiang Li et al., 2010, Schatz et al., 2010).

The higher error rate in the short-read sequencing along with the polymorphism present in the genome imposes a limit on the maximum contiguity that can be assembled. It has been shown that the contiguity measured in terms of N50 (the length X of the sequence which has 50% of all bases in sequences longer than X) of the assembled genome reaches a maximum value at a coverage of around 50X and actually decreases at higher coverage with 1% sequencing error and 1% SNPs in the raw data used for assembly (Daniel 2009). Depending on the dataset used the maximum contiguity is obtained between coverage of 40 and 60 X.

Traditionally repeat regions have been resolved by scaffolding using mate pairs of various insert size. Different insert size libraries have been used with varying degrees of utility. Contiguity, cost benefits, accuracy, genome representation obtained by the different methods has shown that each method has its own advantages and disadvantages (Liang Ye et al., 2011). Repeat content and GC content of genomes being sequenced also have a considerable impact on library preparation, sequencing and assembly methods.

Recently, high quality genome assemblies of mammalian genomes were done from massively parallel sequence data (Sante et al., 2010) with a quality very similar to that

obtained from Sanger sequencing. Using specific types of libraries and new assembly programs (ALLPATHS-LG) high quality assemblies have been obtained.

2.2 Partitioning the genome to reduce the problem of assembly

2.2.1 BAC clones

One way to reduce the problem of assembly is to reduce the assembly target size. Genomic libraries have been divided into smaller parts through the use of Bacterial artificial chromosomes (BAC's) or Yeast artificial chromosomes (YACs). BACs are constructed by partially digesting and fragmenting genomic DNA to obtain large chunks of the genome that are then preserved in bacterial colonies. Traditionally BAC libraries have been constructed based on the "divide and conquer" strategy to obtain high quality genome assemblies using Sanger sequencing. Partitioning the genome into smaller fragments that can be assembled separately simplifies the assembly problem and provides more contiguous assemblies. It has been found that both long and short range mis-assemblies can be reduced in BAC based approaches due to sequencing information being restricted to the genomic region covered by the BAC clone (Marra et al., 1998). The genome can be split into BAC clones and different parts can be sequenced by different groups simultaneously as was done while sequencing the human genome.

BAC libraries have been pooled and sequenced using NGS technologies to obtain assemblies that bring together BAC and NGS methodologies. It has also been shown (Niina et al., 2011) that pooled BAC strategy for whole genome sequencing has many benefits such as lower

cost and better assembly quality compared to whole genome shotgun sequencing. However, BAC libraries are expensive to construct and maintain. BAC libraries suffer from some form of variable cloning bias which leads to over representation of some regions that are cloned better at the expense of other regions that are not cloned. Since, the BAC and YAC strategies involve random fragmentation; the same libraries cannot be reproduced. Contamination from various sources such as the cloning vector used to produce the clone can also be a major problem while using bacteria or yeasts.

2.2.2 Reduced representation library based genome assembly

Construction of a series of reduced representation libraries by size fractionation has been seen as another possible way for partitioning the genome. The genome is fragmented by digesting with a restriction enzyme to obtain a reproducible set of fragments. These fragments are separated based on size into distinct libraries based on size to obtain reproducible genomic partitions.

Whole genome assembly using reduced representation libraries and short reads has been demonstrated to be effective (Young et al., 2010) with the *Drosophila* genome and in comparison with a reference genome sequenced by traditional Sanger sequencing. Better quality assemblies were obtained using restriction enzymes to create reduced representation than using whole genome library assemblies.

Restriction enzyme based fragmentation of genomes produces reproducible fragments. Hence, reduced representation libraries have been used for obtaining SNP maps (Altshuler et al., 2000). It has been suggested (Young et al., 2010) that reduced representation libraries are easier and cheaper to produce than BAC or YAC clones.

Reduced representation libraries can be produced by digesting genomic DNA with a restriction enzyme followed by electrophoretic size separation of genomic DNA. DNA from different size range are cut out from the gel and purified. Each library then consists of fragments of a particular size range. Individual libraries are sequenced separately or sequenced together after tagging them with a unique sequence marker by a process known as bar coding. Assembling these libraries separately has benefits similar to those obtained by using BAC libraries.

It is expected that reduced representation libraries will reduce mis-assemblies without introducing problems associated with the BAC approach. Apart from being expensive to construct and maintain, BAC libraries are prone to contamination and variable cloning bias which leads to over representation of some regions at the expense of other regions. Using single end sequencing for Drosophila genome, Young et al., have shown that reads generated by reduced representation approach gives a better assembly when each of the libraries is assembled separately and put together hierarchically rather than assembling all the reads together. However, reads generated by reduced representation approach will lack overlapping reads at restriction enzyme cut sites which could lead to poor performance of

the whole genome assembly method. Hence, we simulate reads for reduced representation approach and whole genome approach separately.

2.3 Sequencing the crow genome

In this project we tried to ascertain the feasibility and utility of applying such a reduced representation library based approach to genome assembly in a large avian genome of one hooded crow (*Corvus cornix*) individual. The motivation for genome sequencing of the crow comes from its importance in evolutionary biology.

Carrion and hooded crows are a well studied example of incipient speciation, in which two sub-species rarely interbreed or only with little success. However, the underlying genetic mechanism of the hybrid zone is yet to be studied in detail. Although these species are morphologically and taxonomically different, previous studies (Wolf et al., 2010) have shown that molecular differentiation between the species is not pronounced and is in stark contrast to morphological differences in plumage coloration. Hence, a genome wide study could serve as a starting point for obtaining a better understanding of the putatively few genes responsible for the morphological differences. We plan to use the crow genome assembly as a backbone for re-sequencing of several populations in a population genomic framework.

We compared the performance of the WGS assembly strategy with the RRL assembly strategy for sequencing the crow genome using in silico genome sequencing based on

simulations. We also looked at effectiveness and ease of constructing a reduced representation library using restriction enzymes in the laboratory.

3 Materials and methods

3.1 In silico genome sequencing

Using in silico simulations we compared the WGS and RRL method of genome assembly. The WGS method (Figure 3) assembles short reads obtained randomly from the entire genome, while the RRL method (Figure 2) first partitions the genome into smaller fractions which are treated as 'mini-genomes'. Just as in the WGS method, short reads are randomly drawn from these 'mini-genomes' and assembled into contigs. In a second meta-assembly step ‘mini-genomes' are assembled into a complete genome.

For comparing the WGS and RRL methods we make use of the zebra finch genome for performing in silico simulations to simulate the kind of results that would be obtained while using the crow genome. The zebra finch genome is the closest available genome sequence. The latest version of the zebra finch draft genome assembly (WUGSC 3.2.4/taeGut1) was obtained from the UCSC website (http://genome.ucsc.edu/) for performing all simulations.

In the crow, samples are obtained from wild populations as crows as it’s difficult to breed them in captivity and highly inbred individuals cannot be produced due to a suite of practical reasons. Hence, intra-individual polymorphisms need to be incorporated during the assembly process. NGS sequencing also has a high error rate (Dohm et al., 2008). It has been shown that increase in polymorphism and/or sequencing error produces more fragmented assemblies (Daniel. 2009) when using the WGS method of genome assembly.

Based on previous estimates of sequencing error and polymorphism (Wong et Al., 2004) all reads were simulated using dwgsim version: 0.1.2 (part of DNA Analysis Package) with 0.1% sequencing error (base error rate) and 0.1% polymorphism (rate of mutations). Required coverage was obtained using the wrapper script "readsim.pl" (refer supplementary material for source code).

3.1.1 Restriction enzyme and fragment size selection

Reduced representation libraries are constructed so as to partition the genome into smaller sized reproducible fractions that are easier to handle in the assembly process. For each enzyme, it would be optimal to have equally sized partitions of the genome to ensure uniform distribution of reads. When the genome is yet to be sequenced identifying enzymes which partition the genome into equally sized partitions can only be achieved by using a closely related genome. We choose to use the chicken and zebra finch genomes which are closely related to the crow genome, to ensure that the fragmentation pattern is robust despite the multiple gaps in the final draft genomes.

During library preparation in the laboratory, fragments <1kb will be lost. Moreover, with very large fragments (>20kb) it would make it difficult to partition the genome into 4 equally sized libraries. Therefore, an enzyme needed to be selected such that less than 5% of the genome was present in fragments <1kb or >20kb.

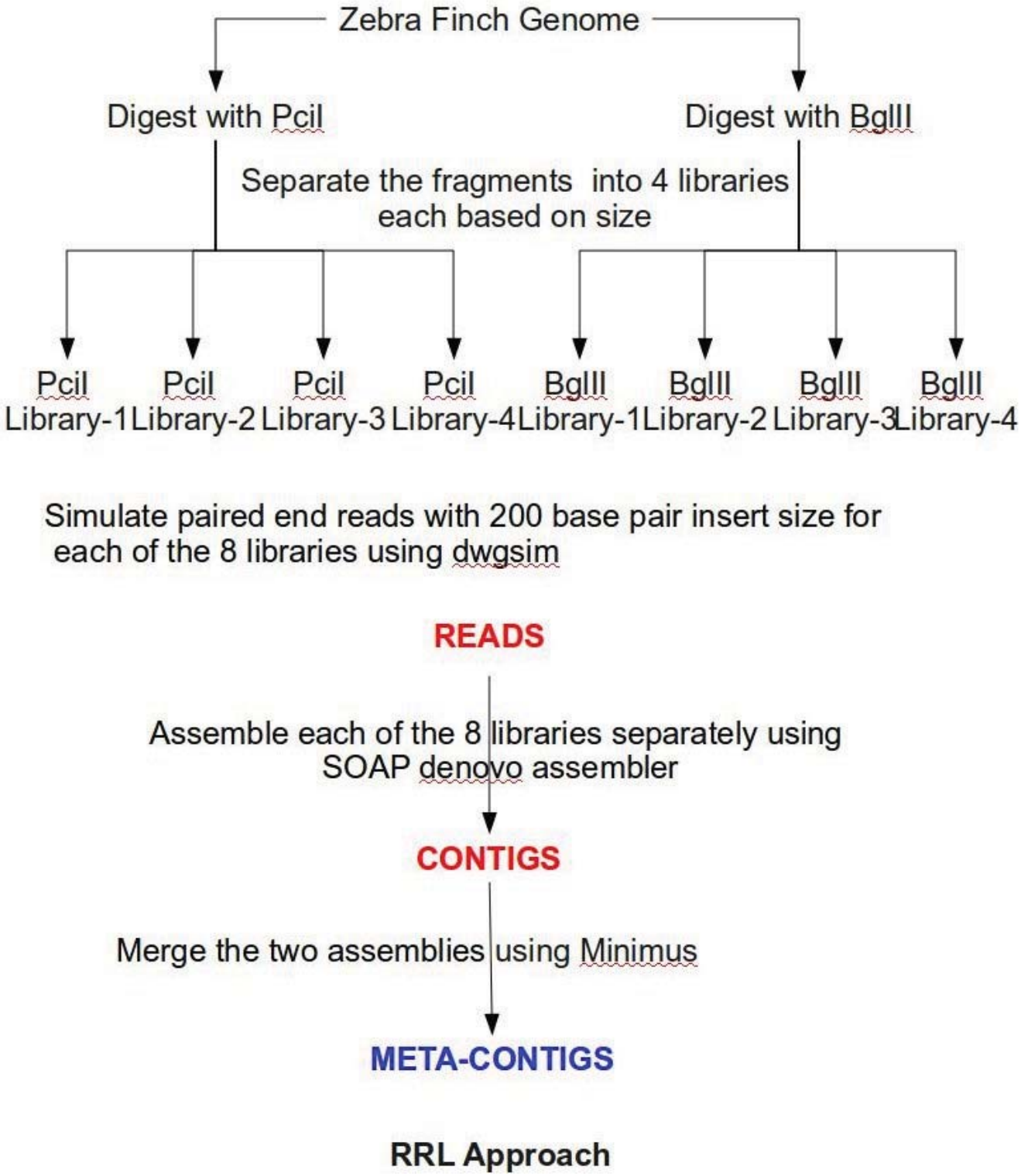

Figure 2: Reduced Representation Library (RRL) approach to genome assembly (Approach taken in this thesis project)

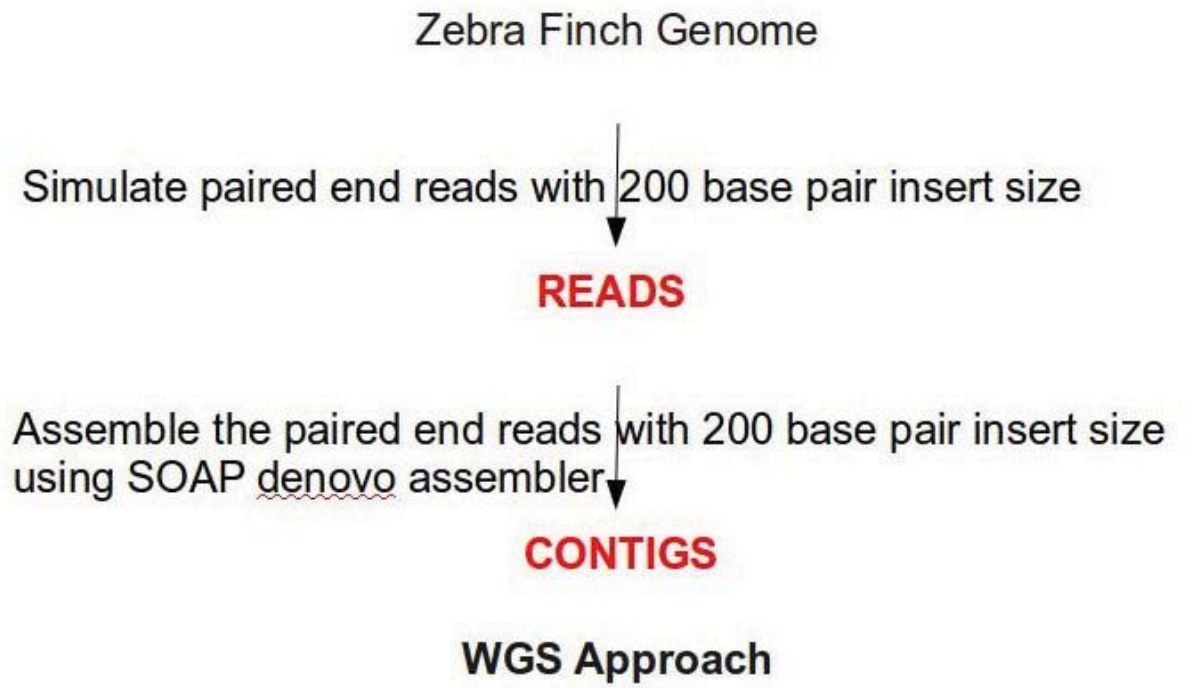

Figure 3: Whole Genome Shotgun (WGS) approach to genome assembly

Different enzymes can lead to differently sized fragments based on the frequency of the occurrence of the restriction recognition site. To select enzymes that could be suitable for preparation of reduced representation libraries, all the 755 enzymes available in REBASE version 909 were tested against the chicken and zebra finch genome. Among these the enzymes which were not sensitive to methylation or CpG sites were selected based on their availability due to obtain the expected size distribution of fragments and also avoid

heterogeneity in the DNA cleavage reaction. Following candidate enzymes (Figure 4) were

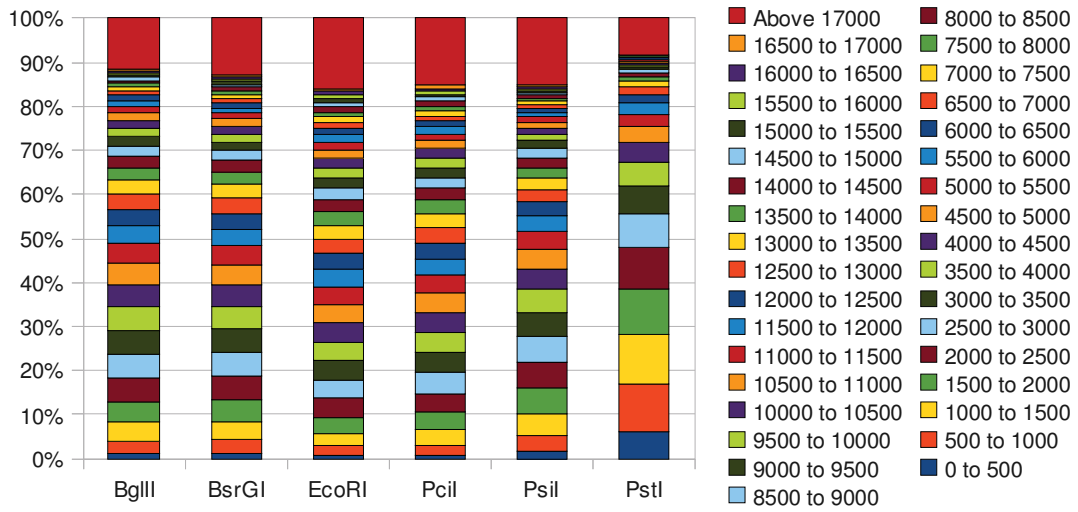

Figure 4: Fragment size distribution for some of the short listed enzymes, namely, BglII, BsrGI, EcoRI, PciI, PsiI and PstI

selected based on these criteria.

A high correlation between fragment size distributions between chicken and zebra finch suggests that a similar length profile can be found in the crow (Table 2).

Enzyme	Pearson Correlation (r) calculated using R function cor.test
BglII	0.91
BsrGI	0.92
EcoRI	0.78
PciI	0.79
PsiI	0.82
PstI	0.93

Table 2: Correlation (r) between the fragment size distribution between chicken and zebra finch Genomes

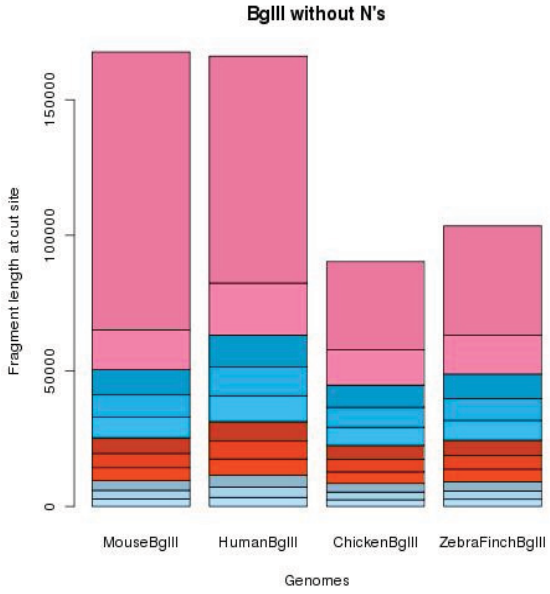

Figure 5.1: Variation in BglII fragment size distribution at Genomic partitions in human, mouse, chicken and zebra finch genome. Fragment sizes were determined after removing all gaps represented by N's in the genome.

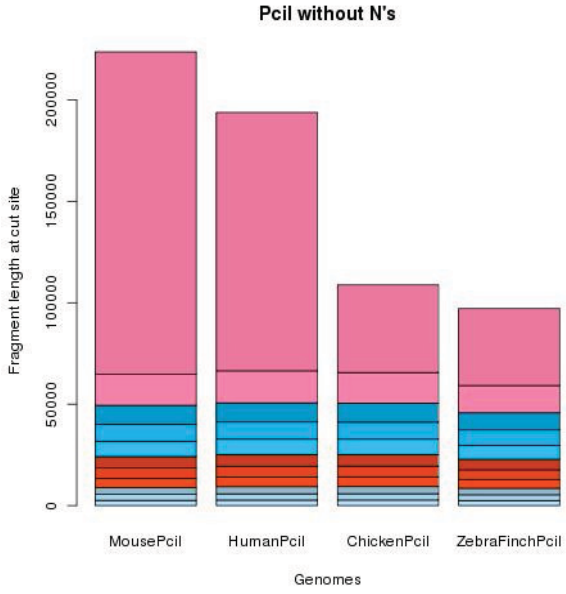

Figure 5.2: Variation in PciI Fragment size distribution at genomic partitions in human, mouse, chicken and zebra finch genome. Fragment sizes were determined after removing all gaps represented by N's in the genome

PciI and BglII were finally selected for the construction of the reduced representation libraries for the crow genome from the candidate set (Figure 5) due to the high degree of

correlation in the fragment size distribution based on simulation done using the chicken and zebra finch genome.

Reduced representation libraries need to be constructed in such a way that the genome is approximately equally partitioned among the different libraries. Zebra Finch is the most closely related species to the crow that has been sequenced and assembled. Hence, the size range which partitions the zebra finch genome into four equal partitions has to be used to partition the RR libraries. However, many large gaps are present in the Zebra Finch genome. To obtain robust values for the fragment size all analyses were repeated with chicken, human and mouse genomes to get an idea of the variation in the fragment size distribution between the different genomes.

The fragment size distribution in closely related genomes such as mouse, human, chicken and zebra finch genomes without any gaps was used to check for variation in fragment sizes. Based on similarity in the fragment size distribution in different genomes, fragment size distribution in chicken was used to select the cut sites for PciI and fragment size distribution in zebra Finch was used to select the cut sites for BglII.

Restriction sites were identified on both the positive and negative strands of the DNA to get the fragment size distribution. Restriction recognition sites (in PciI ACATGT) were used to identify the sites and cut sites (in PciI A'CATG) were used to obtain the different fragments.

Exact size range for each library was found at 75%, 50% and 25% of the total genome size after excluding fragments shorter than 1 KB. While cutting out the different libraries from the gel in the laboratory based on the position of the DNA ladder, fragment sizes can only be known approximately. Hence, we also checked the robustness of the library fragment size boundaries. Library sizes and changes in library fragment size boundaries with small changes in the library sizes (Table 3) provide the information needed to partition the genome.

Library (Enzyme)	Lower Bound	-5% error	Upper Bound	+5% error
Library-1(PciI)	4.17% (<1kb)	2639(20%)	3090	3541(30%)
Library-2(PciI)	3091	4996 (45%)	5549	6139 (55%)
Library-3(PciI)	5550	8357 (70%)	9320	10548(80%)
Library-4(PciI)	9321	NA	48000	NA
Library-1(BglII)	4.09%(<1kb)	2651(20%)	3106	3570(30%)
Library-2(BglII)	3107	5077(45%)	5654	6290(55%)
Library-3(BglII)	6291	8755(70%)	9956	11525(80%)
Library-4(BglII)	11526	NA	48000	NA

Table 3: Size range for the different libraries digested by different enzymes

3.1.2 de novo assembly

WGS method of genome assembly:

For performing the WGS method of genome assembly 100 bp reads with 400 bp insert size were generated with coverage of 20, 30, 40, 50 and 60 X. Each of the data sets with different coverage was assembled separately using the 64 bit version of SOAP denovo (Li et al., 2010)genome assembler (with 31, 63 and 127 kmer) trying out all kmer sizes from 21 to 63 and selecting the kmer size with the best N50 value.

RRL method of genome assembly:

For the reduced representation library approach, zebra finch genome was digested in-silico with PciI and BglII using the "Restriction Digestion" perl module (refer supplementary material for source code) to obtain the different fragments that would be obtained if genomic DNA was actually digested in the laboratory. These fragments were grouped into 4 libraries each based on the size range that has been selected (Table 3). In each of the 8 libraries 100 bp reads with 400 bp insert were generated with 20, 25 and 30 X coverage. Each of the eight libraries was assembled separately using the SOAP denovo assembler probing all kmer sizes from 21 to 63. For each of the datasets the kmer with the best N50 value was selected. Best kmer assembly contigs from each of the 4 libraries for each enzyme were pooled to get enzyme contigs.

The 2 sets of enzyme contigs were merged by meta-assembly using Minimus assembler (Sommer et al., 2007) from the AMOS package with "-D Refcount" option specifying the two sets of enzyme contigs as two distinct assemblies. Both possible orders of assemblies were tried and the order with a higher N50 value was chosen. Minimus2 makes use of Nucmer from MUMmer package. The MUMmer package was compiled with the 64 bit option to be able to assemble large datasets.

For each of the datasets the kmer with the best N50 value was selected. Various statistics such as N50, longest contig, total number of assembled bases, total number of contigs was calculated for these assemblies using the script "N50.pl" (refer supplementary material for source code). The assembled contigs were aligned against the reference genome using nucmer from MUMmer package. The percent of the genome that was assembled (recovery) and the correctly assembled proportion of the contigs (accuracy) were calculated using 'show-tiling' from MUMmer package.

3.2 Laboratory methods

Prior to sequencing, genomic DNA needs to be extracted and prepared into a form suitable for sequencing depending on the sequencing technology being used and sequencing strategy being adopted.

3.2.1 DNA extraction and quality check

DNA required for genomic sequencing was extracted from frozen blood samples preserved in Queen's Lysis buffer. Refer to the Supplementary material for detailed protocol used to obtain high quality DNA with minimal fragmentation. Purity and quantity of extracted DNA was measured by Nanodrop and Qubit (Invitrogen) measurements. For quality evaluation, DNA was run on a 1% gel beside a high molecular weight marker (GRHR+OR ladder, Fermentas) (example gel picture see Figure 7).

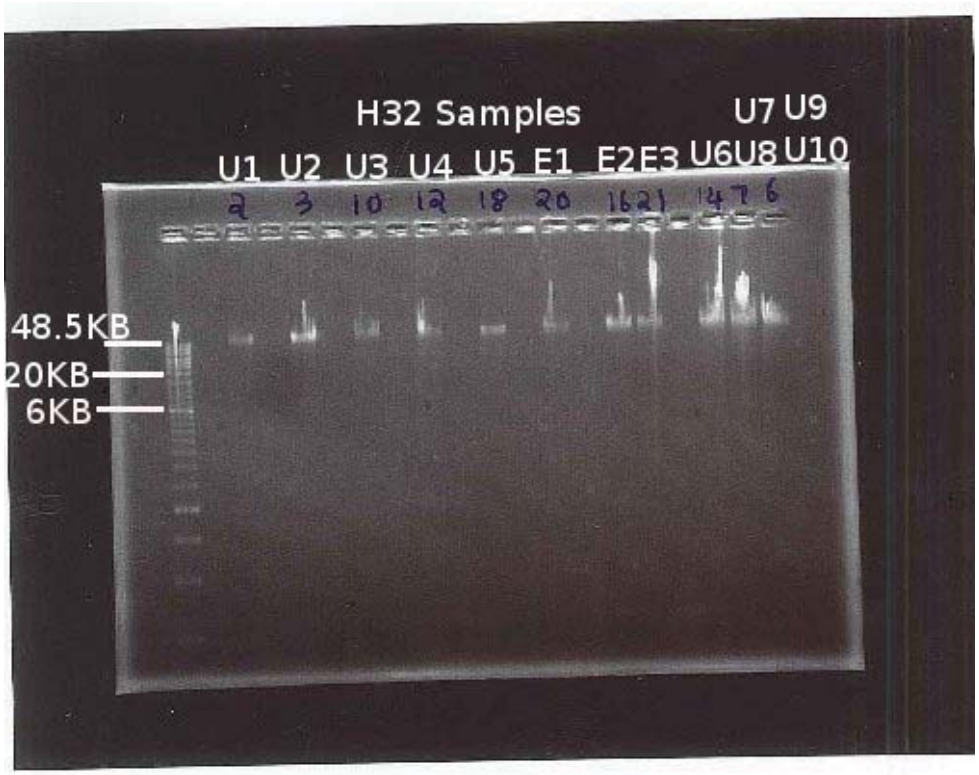

Figure 7: Crow genomic DNA samples fragmentation check after extraction from blood.

3.2.2 Reduced representation library construction

Reduced representation libraries were constructed using restriction enzymes selected using the zebra finch genome for finding fragment size distributions. Genomic DNA extracted from crow blood was checked for purity and quality as described above. Refer to the Supplementary material for the detailed protocol used to construct the reduced representation libraries. Purified genomic DNA was fully digested with the pre-selected enzyme (ex: BglII). The full digest was then run on 1% agarose gel to separate the fragments based on size.

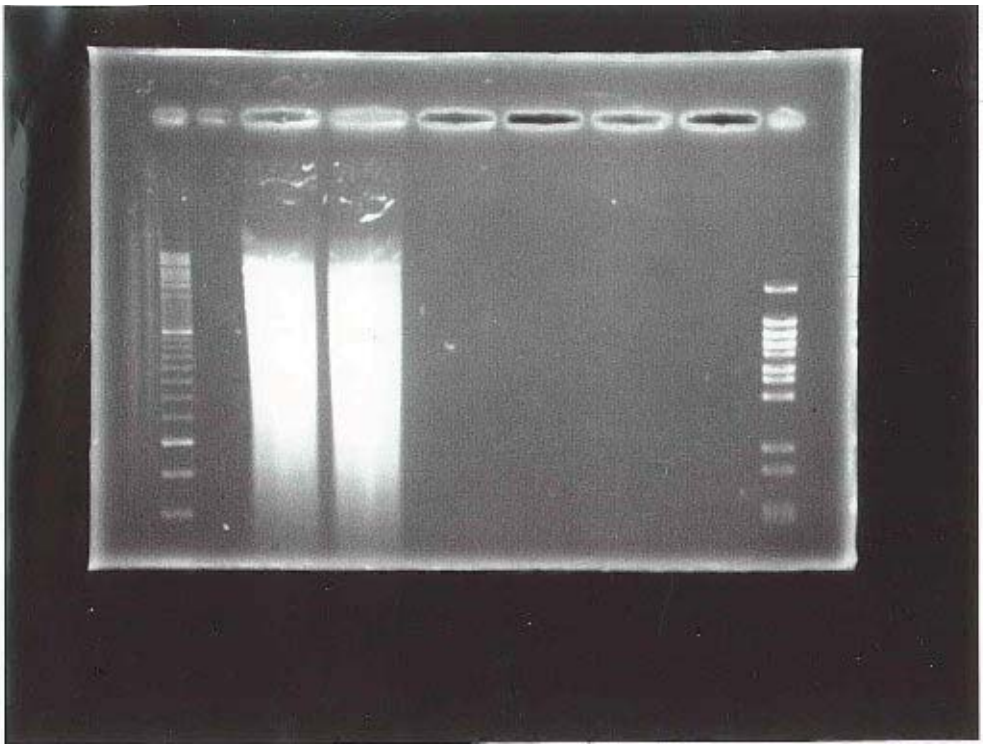

Figure 8: Crow Genomic DNA was digested with the selected restriction enzyme and run on a 1% agarose gel for partitioning the genome into libraries based on size

From the genomic DNA digest four differently sized (Table 3) libraries were cut out and purified from the gel (refer supplementary information for detailed protocol). Different methods (Qiagen, Gelase, Freeze and squeeze) for extracting DNA from the gel were tried to select a method that provided the best yield with minimal fragmentation (data not shown).

Multiple gels were run to obtain more than 10 ug of DNA per library. The four libraries were again run on 1% agarose gel to ensure proper separation of the fragments into the libraries of the required size.

4 Results

4.1 Simulated assemblies

We assembled 100bp reads with 400bp insert size simulated from the zebra finch genome for the entire genome and from reduced representation libraries separately with the goal to compare the two approaches outlined above.

4.1.1 Comparison of WGS vs RRL strategy

The reduced representation library approach gave (Table 4) longer continuous contigs (N50 that is 1kb more than the WGS sequencing strategy). However, the recovery from the WGS method was 3% higher while the accuracy remains the same. Since, a dataset with 1% sequencing error and 1% polymorphism was used an accuracy of 99% is expected.

4.1.2 Evaluation of RRL library preparation in the laboratory

The construction of the RRL libraries requires the separation of the digested DNA fragments based on size using gel electrophoresis into separate libraries. Resolution of the high molecular weight DNA requires the use of low (1 to 0.5%) concentration gels which can be difficult to handle, especially for cutting out bands of various sizes.

Library	Best Kmer	Number of Contigs	Longest Contig	Total number of bases	N50	Recovery	Accuracy
Merged with Minimus (20X +20X)	BP (BglII - PciI)	81370	101091	193094868	8177	92.88	99.9
Merged with Minimus (25X +25X)	BP (BglII - PciI)	78843	87214	192987837	8443	92.9	99.9
Wgs (20X)	47	3,13,964	1,10,286	203077063	6767	94.87	99.93
Wgs (25X)	49	3,00,744	1,07,314	203467969	6964	95.19	99.93
Wgs (30X)	49	3,02,712	1,15,974	203646071	7022	95.21	99.93
Wgs (35X)	49	3,03,677	1,03,197	203751991	7054	95.23	99.93
Wgs (40X)	49	3,05,229	1,06,435	203864219	7090	95.24	99.93
Wgs (45X)	49	3,06,161	1,05595	203940103	7099	95.28	99.93
Wgs (50X)	49	3,07,288	1,05,596	204020912	7085	95.27	99.93

Table 4: Comparison of Whole Genome Sequencing with the reduced representation library strategy based on simulated paired end reads

Correct size fractions can be cut out from the gel using high molecular weight markers as a reference. However, differences in the concentration of the DNA in the markers and the sample can lead to the marker and sample DNA running at different speeds. It could be seen from repeated tests that the DNA marker was moving master than the digested DNA sample due to its lower concentration. The use of high concentrations of DNA marker leads to smearing of the marker and results in loss of resolution. Ethidium bromide was used due to its compatibility with the downstream sequencing steps.

The bands of the appropriate size were cut out from the gel after visualizing them with Ultra-violet light for a short duration. Exposure of DNA to UV leads to degradation and fragmentation of DNA. Using a glass plate to shield from the UV will reduce the damage to

the DNA. After cutting out the bands from the gel, DNA has to be purified from the gel. Purification of DNA from gel results in fragmentation and loss of DNA. Hence, the method of purification which provides the best yield with least fragmentation needs to be used for purification of DNA from the gel.

Use of Qiagen QIAEX II beads, Gelase and freeze and squeeze methods for purification of DNA from the agarose gel were tried. Qiagen provides the best yields (data not shown) for the low molecular weight libraries while the Gelase method is better for the high molecular weight libraries. However, the ammonium acetate precipitation step in the Gelase method could lead to inactivation of the ligase enzyme while adding the adapters prior to sequencing. All the methods show similar amount of fragmentation of the DNA after purification and provide yields of approximately 50%. Loosing half the DNA in a single purification step requires the use of large amounts of starting material for the RRL approach.

The first step of separation of DNA based on size needs to be verified and validated by at least one more step of size separation. Each step of size separation using gel electrophoresis leads to further fragmentation and loss of DNA. Hence, to obtain a yield of 2 ug DNA for each library, a total of 80 to 100 ug of DNA has to be digested. Digestion and phenol chloroform purification to remove the restriction enzyme leads to a loss of about 20 ug of DNA. The first step of size separation leaves approximately 20 ug of DNA (5 ug/ library). Second step of size separation will give a yield of approximately 2 ug/ library for sequencing. Hence, the RRL approach requires 50 times more DNA than the WGS approach to sequencing.

5 Discussion

Obtaining high -quality draft genomes for an ever increasing number of species e.g. 10000 species genome project (Hered, 2009), requires improvements in both sequencing technologies and strategies. Although sequencing technology has improved drastically over the past decade, sequencing strategies have been mainly confined to 'WGS' and 'BAC' based whole genome sequencing or a hybrid of the two strategies.

Recently interest in different strategies for whole genome sequencing (Paszkiewicz et al., 2010) has lead to other approaches including reduced representation based whole genome sequencing (Springer et al., 2004, Barbazuk et al., 2005) and random PCR based genome sequencing (Nlshlgakl et al., 2000) being revisited with NGS technologies (Young et al., 2009). As more of the genomes available in nature are sequenced, it might be easier to use reference assisted assembly methods (Stratton 2009) to build reference guided genomes of closely related species from existing assemblies by simply mapping reads to closely related reference. Being able to sequence targeted reduced regions of genomes could be very useful in such comparative assembly approaches.

The RRL approach is immune to uneven coverage compared with WGS approach as regions with lower coverage alone can be sequenced in a separate run. RRL based approach may also serve as a genome finishing tool for WGS projects which have low coverage in specific

regions of the genome. While the BAC library approach requires the construction and sharing of expensive libraries, RRL's can be reproduced at a lower cost.

Due to variation between gel runs while constructing RRL's can lead to mixing of fragments between libraries which can lead to incorrect assemblies. Although RRL approach to genome assembly provides improvements in genome contiguity, latest developments in the use of highly overlapping paired end reads and long mate pair libraries (Sante et al., 2010) might be able to provide high quality assemblies without the need for constructing RRL's. The Minimus assembler used for merging the two sets of assemblies obtained from the two different enzymes is currently able to merge only the contigs generated by the initial assembly steps. Hence, the information available in paired end reads is not utilized effectively.

Coverage at different regions of the genome could be affected by the construction of the reduced representation libraries. These differences in coverage need to be analyzed and compared with the variation present in the whole genome approach. Being able to overcome biases in coverage can be very useful in various applications such as SNP calling and Exxon capture.

The reduced representation library approach to genome assembly was suggested for improving the quality of assemblies prior to the wide spread use of paired end and mate pair reads using Next-Generation sequencing methods. With recent advances in

construction of paired end and mate pair libraries with very large insert sizes the WGS method is able to produce assemblies with much better quality without the having to construct reduced representation libraries.

6 Conclusion

Based on results of various simulations the reduced representation library based method seems to produce more contiguous assemblies than traditional whole genome assembly even for mammalian sized genomes. The effectiveness of using a closely related genome to select the enzymes and fragment size range for the reduced representation libraries can be verified after using these values obtained from the zebra finch genome to assemble the crow genome.

Although the reduced representation library based approach is beneficial it requires special library construction which requires large (80 ug for constructing four 2 ug RRL's) amounts of DNA. Increased handling of the DNA can lead to increased contamination of the libraries. Cross contamination between the libraries can cause miss-assemblies and lead to collapsing of repeats during the initial assembly step. Checks would be required at each stage of the assembly to avoid such errors in the assemblies.

Software programs for assembly of such large datasets need to be improved to produce faster assemblies. Faster versions of Nucmer or alternative programs such as blat (Kent, 2002) should be used in the overlapping step of the assembler. Similarly layout (placement of reads relative to each other based on insert size estimates) and consensus (finding the appropriate base at a particular position identified given the possibility of different bases due to sequencing errors and polymorphism) steps need to be distributed for faster processing. Current assemblers such as Minimus are able to handle only merging of

singletons (continuous sequences of DNA with no gaps). Merging of scaffolds from two assemblies also requires newer assembly programs capable of handling mistakes in gap size estimates.

7 References

1. A. L. Young, H. O. Abaan, D. Zerbino, J. C. Mullikin, E. Birney and E. H. Margulies. A new strategy for genome assembly using short sequence reads and reduced representation libraries. 2010. Genome Res. 20: 249-256
2. A. Sharman. The many uses of a genome sequence. 2001. Genome Biology 2001, 2:reports4013-reports4013.4 doi:10.1186/gb-2001-2-6-reports4013
3. Barbazuk, W., Bedell, J., Rabinowicz, P., 2005. Reduced representation sequencing: a success in maize and a promise for other plant genomes. Bioessays 27(August(8)), 839–848.
4. D. R. Zerbino and E. Birney. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. 2008. Genome Res.18: 821-829 originally published online March 18, 2008
5. D. R. Zerbino. Genome assembly and comparison using de Bruijn graphs. 2009. A dissertation submitted to the University of Cambridge for the degree of Doctor of Philosophy. Darwin College, European Molecular Biology Laboratory, European Bioinformatics Institute.
6. D. Altshuler, V. J. Pollara, C. R. Cowles, W. J. Van Etten, J. Baldwin, L. Linton & E. S. Lander. An SNP map of the human genome generated by reduced representation shotgun sequencing. September 2000. Nature 407, 513-516 (28 September 2000) | doi:10.1038/35035083;
7. Denisov et al. Consensus Generation and Variant Detection by Celera Assembler. 2008. Bioinformatics 24(8):1035-40
8. DNAA. Whole Genome Simulation. 28 April 2010.

http://sourceforge.net/apps/mediawiki/dnaa/index.php?title=Whole_Genome_Simulation. April 26th 2011.

9. Dohm, J. C., Lottaz, C., Borodina, T., and Himmelbauer, H. (2008). Sub-stantial biases in ultra-short read data sets from high-throughput DNAsequencing. Nucleic Acids Research 36:e105.
10. EPICENTRE Biotechnologies. Gel-Digesting Preparation. Cat. Nos. G09050, G09100, G09200, G31050, G31200, G191ML, and G195ML. 6/2010. http://www.epibio.com/pdftechlit/025pl0610.pdf. April 26th 2011.
11. F. Sanger and A. R. Coulson. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. Journal of Molecular Biology Volume 94, Issue 3, 25 May 1975, Pages 441-446
12. J Hered. Genome 10K Community of Scientists. Genome 10K: A Proposal to Obtain Whole-Genome Sequence for 10 000 Vertebrate Species. 2009. 100 (6): 659-674. doi: 10.1093/jhered/esp086
13. J. C. Venter, M. D. Adams, E. W. Myers, et al. The Sequence of the Human Genome.2001. Science Vol. 291 no. 5507 pp. 1304-1351
14. J. R. Miller, S. Koren, G. Sutton. Assembly algorithms for next-generation sequencing data. 2010. Genomics 95 315–327
15. J B W Wolf, T Bayer, B Haubold, M Schilhabel, P Rosenstiel, D Tautz. Nucleotide divergence versus gene expression differentiation: comparative transcriptome sequencing in natural isolates from the carrion crow and its hybrid zone with the hooded crow. 2010. Mol Ecol 19: Suppl. 1. 162-175
16. K. Scheibye-Alsing, S. Hoffmann, A. Frankel, P. Jensen, P.F. Stadler , Y. Mang , N. Tommerup ,M.J. Gilchrist , A.-B. Nygård, S. Cirera , C.B. Jørgensen , M. Fredholm , J.

Gorodkin. Sequence assembly. 2009. Computational Biology and Chemistry 33 (2009) 121–136

17. K J Hoff. The effect of sequencing errors on metagenomic gene prediction. BMC Genomics 2009, 10:520doi:10.1186/1471-2164-10-520
18. K W James. 2002. BLAT--the BLAST-like alignment tool. Genome research 12 (4): 656–664. doi:10.1101/gr.229202. PMC 187518. PMID 11932250.
19. K Nishigaki, K Akasaka and T Hasegawa. Random PCR-Based Genome Sequencing: A Non-Divide-and-Conquer Strategy. 2000. DNA RESEARCH 1, 19 26 (2000)
20. K Paszkiewicz and D J. Studholme. De novo assembly of short sequence reads. 2010. Brief Bioinform (2010) 11 (5): 457-472. doi: 10.1093/bib/bbq020
21. L Ye, L W Hillier, P Minx, N Thane, D Locke, J C Martin, L Chen, M Mitreva, J R Miller, K V Haub, D Dooling, E R Mardis, R K Wilson, G M Weinstock, W C Warren. A vertebrate case study of the quality of assemblies derived from next-generation sequences. Genome Biology 2011, 12:R31. doi:10.1186/gb-2011-12-3-r31
22. Marra, M., Hillier, L., Waterston, R. Expressed sequence tags–Establishing bridges between genomes. 1998. Trends Genet. 14 (January (1)), 4–7.
23. M Boetzer, Christiaan V. Henkel, H J. Jansen, D Butler and W Pirovano. Scaffolding pre-assembled contigs using SSPACE. 2011. Bioinformatics (2011) 27 (4): 578-579. doi: 10.1093/bioinformatics/btq683
24. M C Schatz, A. L. Delcher and S L. Salzberg. Assembly of large genomes using second-generation sequencing. 2010. Genome Res. published online May 27, 2010. doi:10.1101/gr.101360.109
25. M L. Metzker. Sequencing technologies — the next generation. January 2010. Nature Reviews Genetics volume 11.

26. N Haiminen, F A Feltus, L Parida. Assessing Pooled BAC and Whole Genome Shotgun Strategies for Assembly of Complex Genomes. 2011. BMC Genomics 2011, 12:194. doi:10.1186/1471-2164-12-194

27. P Flicek & E Birney. Sense from sequence reads: methods for alignment and assembly. NOVEMBER 2009.nature methods supplement VOL.6 NO.11s.

28. Pop, M. Shotgun sequence assembly. 2004. Adv. Comput. 60, 193–248

29. QIAGEN. DNeasy® Blood & Tissue Handbook. 07/2006. www.qiagen.com/hb/dneasybloodtissuekit_en. April 26th 2011.

30. QIAGEN. QIAEX II Handbook. October 2008. www.qiagen.com/hb/qiaexii. April 26th 2011.

31. R. Baer, A. T. Bankier, M. D. Biggin, P. L. Deininger, P. J. Farrell, T. J. Gibson, G. Hatfull, G. S. Hudson, S. C. Satchwell, C. Séguin, P. S. Tuffnell & B. G. Barrell. DNA sequence and expression of the B95-8 Epstein—Barr virus genome. Nature 310, 207-211 (19 July 1984) | doi:10.1038/310207a0; Accepted 1 May 1984

32. RD Fleischmann, MD Adams, O White, RA Clayton, EF Kirkness, AR Kerlavage, CJ Bult, JF Tomb, BA Dougherty, JM Merrick and et al. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. Science 28 July 1995:Vol. 269 no. 5223 pp. 496-512 DOI: 10.1126/science.7542800

33. R Arratia, E S. Lander, T, S Tavar and Michael Waterman. Genomic Mapping by Anchoring Random Clones:A Mathematical Analysis. 1991. Genomics 11, 806-827 (1991)

34. Ruiqiang Li et.al. The sequence and de novo assembly of the giant panda genome. Vol 463 | 21 January 2010 | doi:10.1038/nature08696

35. R Li, H Zhu, J Ruan, W Qian, X Fang, Z Shi, Y Li, S Li, G Shan, K Kristiansen, S Li, H Yang,

J Wang and J Wang. De novo assembly of human genomes with massively parallel short read sequencing. 2010. Genome Res. 2010. 20: 265-272

36. S Gnerre, E S Lander, K Lindblad-Toh and D B Jaffe. Assisted assembly: how to improve a de novo genome assembly by using related species. Genome Biology 2009, 10:R88 (doi: 10.1186/gb-2009-10-8-r88).
37. S Gnerrea, I MacCalluma, D Przybylskia, F J. Ribeiroa, J N. Burtona, Bruce J. Walkera, T Sharpea, G Halla, Terrance P. Sheaa, S Sykesa, Aaron M. Berlina, D Airda, M Costelloa, R Dazaa, L Williamsa, R Nicola, A Gnirkea, C Nusbauma, E S. Landera, and D B. Jaffea. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. December 27, 2010, doi: 10.1073/pnas.1017351108
38. Sommer DD, Delcher AL, Salzberg SL, Pop M. Minimus: A fast, lightweight genome assembler.2007 BMC Bioinformatics 8:64, doi: 10.1186/1471-2105-8-64.
39. Springer, N., Xu, X., Barbazuk, W. Utility of different gene enrichment approaches toward identifying and sequencing the maize gene space. 2004. Plant Physiol. Prev. 136, 3023–3033.
40. Stratton, M. Genome resequencing and genetic variation. Nat. Biotechnol. 2008. 26,65-66
41. W. Fiers, R. Contreras, F. Duerinck, G. Haegeman, D. Iserentant, J. Merregaert, W. Min Jou, F. Molemans, A. Raeymaekers, A. Van den Berghe, G. Volckaert & M. Ysebaert. Complete nucleotide sequence of bacteriophage MS2 RNA: primary and secondary structure of the replicase gene. Nature 260, 500 - 507 (08 April 1976); doi:10.1038/260500a0
42. Wetterstrand KA. DNA Sequencing Costs: Data from the NHGRI Large-Scale Genome Sequencing Program Available at: www.genome.gov/sequencingcosts. Accessed 3rd

May 2011.

43. Wong GK, Liu B, Wang J, Zhang Y, Yang X, Zhang Z, Meng Q, Zhou J, Li D, Zhang J, et al. A genetic variation map for chicken with 2.8 million single-nucleotide polymorphisms. 2004. Nature 432:717–722.

8 Supplementary material

8.1 Source code for various scripts used in the project

readsim.pl

```
#! /usr/bin/perl

# readsim.pl

# October 2010

# Author:Nagarjun Vijay

#usage perl readsim.pl genome_fasta_file read_length required_coverage required_sequencing_error required_polymorphims insert_length

#example: perl readsim.pl chr42.fa 100 20 0.001 0.001 200

use strict;

use warnings;

if(@ARGV != 6){

print "usage perl readsim.pl genome_fasta_file read_length required_coverage required_sequencing_error required_polymorphims insert_length\n";

}

else{

my $file=$ARGV[0];#input file name
```

```
my $readlen=$ARGV[1];#required length of reads

my $coverage=$ARGV[2];#required coverage

my $error=$ARGV[3];#sequencing error to be simulated

my $poly=$ARGV[4];#degree of polymorphism in data

my $insert=$ARGV[5];#insert length

my ($genseq,$genseqst_temp,$totalbases,$n)=("","",0,0);

open(GENOME, $file);

while($genseq=<GENOME>){

        if($genseq!~ />/) {

        $genseqst_temp .= $genseq;

        }

}

close(GENOME);

$totalbases=length $genseqst_temp;

print "Total bases: $totalbases\n";

$n=int($totalbases/200)*$coverage;

print "./dwgsim -e $error -r $poly -N $n -1 $readlen -2 $readlen -d $insert $file
$file.bwa.read1.fq $file.bwa.read2.fq $file.bfast.fq\n";
```

```
system("./dwgsim -e $error -r $poly -N $n -1 $readlen -2 $readlen -d $insert $file $file.bwa.read1.fq $file.bwa.read2.fq $file.bfast.fq\n");
}
```

N50.pl

```
#!/usr/bin/perl

my ($seqlen,$gensize)=(0,0);
my @x;
while(<>){
	if(/^[\>\@]/){
		if($seqlen>0){$gensize+=$seqlen;push @x,$seqlen;}
	$seqlen=0;
	}
	else{
	s/\s//g;
```

```
        $seqlen+=length($_);}

        }

if ($seqlen>0){$gensize+=$seqlen;push @x,$seqlen;}



@x=sort{$b<=>$a} @x;

my ($count,$fifty,$ninety,$f)=(0,0,0,0);

#print "Number of contigs: $#x\n";

print "\t$#x\t";

#print "Longest contig: $x[0]\n";

print "$x[0]\t";

#print "Total bases covered: $gensize\n";

print "$gensize\t";

for (my $j=0;$j<@x;$j++){

        $count+=$x[$j];
```

```
        if (($count>=$gensize/2)&&($fifty==0)){

#       print "N50 size: $x[$j]\n";

#       print "N50 Contigs: $j\n";

        print "$x[$j]\t";

        print "$j\t";



        $fifty=$x[$j];

        }elsif (($count>=$gensize*0.9)&&($ninety==0)){

#       print "N90: $x[$j]\n";

#       print "N90 Contigs: $j\n";

#       print "$x[$j]\t";

#       print "$j\t";



        $ninety=$x[$j];
```

```
        }

        elsif (($x[$j]<=500)&&($f==0)){

#       print "Contig size smaller than 500 bp: $x[$j]\n";

#       print "Contigs larger than 500 bp: $j\n";

#       print "$x[$j]\t";

#       print "$j\t";



        $f=$x[$j];

        }

}

print "\n";
```

Restriction_Digest.pm:

```
package Restriction_Digest;
```

```
# before September 2010
# Restriction_Digest perl module

# Author:Till Bayer, Jochen Wolf



sub findres {


( $infile, $rebase_file, $min, $max, @enzymes)= @_;

#list of arguments passed to module
my $enzyme_db = parse_rebase($rebase_file);
#load fasta file into memory
my $seqs = parse_fasta($infile);
my %enzyme_stats;
unless (-d "./digest/"){
   mkdir "./digest/"
}
for my $enzyme (@enzymes) {
   my $outfile   = "./digest/".(split(/\./,$infile))[0] . '_' . $enzyme . '.fasta';
   my $statsfile = "./digest/".(split(/\./,$infile))[0] . '_' . $enzyme . '.stats.txt';
   open my $OUT, '>', $outfile
      or die "Could not open file $outfile: $!\n";
   unless ( exists $enzyme_db->{$enzyme} ) {
      die "Don't have enzyme $enzyme in enzyme database $rebase_file.\n";
   }
```

```
    my %stats;
    for my $name ( sort keys %$seqs ) {
        print STDERR "Digesting sequence $name with $enzyme.\n";
        my $sizes = digest( $seqs->{$name}, $enzyme_db->{$enzyme} );
        $stats{$name} = calc_stats( $sizes, length $seqs->{$name} );
        print_fragments( $name, $seqs->{$name}, $sizes, $OUT );
    }
    close $OUT;
    $enzyme_stats{$enzyme} = calc_enzyme_stats( \%stats );
    print_stats( \%stats, $enzyme_stats{$enzyme}, $statsfile );
}
my $summaryfile = "./digest/".(split(/\./,$infile))[0] . '.summary.txt';
print_summary( $summaryfile, \%enzyme_stats );
print STDERR "Done.\n";
}
######## SUBS ########
sub parse_fasta {
    my $file = shift;

    print STDERR "Loading fasta file.\n";
    open my $IN, '<', $file or die "Can't open file $file\n";
    my %seqs;
    my $name;
    while (<$IN>) {
        chomp;
        unless ($_) {next}
        elsif (/^>/) {
            s/>//;
```

```
            $name = $_;
            print STDERR "Loading sequence $name.\n";
        }
        else {
            $seqs{$name} .= uc $_;#added "uc" to ignore case on 6-Sep-2010
        }
    }
    close $IN;
    return \%seqs;
}
sub parse_rebase {
    my $file = shift;
    open my $IN, '<', $file or die "Can't open file $file\n";
    my %enzymes;
    while (<$IN>) {
        next if /^#/ or /^$/;
        my @data = split /\t/, $_;
        $enzymes{ $data[0] }->{len}    = $data[2];
        $enzymes{ $data[0] }->{ncuts}  = $data[3];
        $enzymes{ $data[0] }->{blunt}  = $data[4];
        $enzymes{ $data[0] }->{cut1_5} = $data[5];
        $enzymes{ $data[0] }->{cut1_3} = $data[6];
        $enzymes{ $data[0] }->{cut2_5} = $data[7];
        $enzymes{ $data[0] }->{cut2_3} = $data[8];
        ( $enzymes{ $data[0] }->{pattern}, $enzymes{ $data[0] }->{is_palindrome}, $enzymes{
$data[0] }->{fwd_regex} ) = make_pattern_regex( $data[1] );
    }
    close $IN;
    return \%enzymes;
```

```
}
sub make_pattern_regex {
   my $pattern = shift;
   $pattern = uc $pattern;
my $revpattern = reverse $pattern;
   $revpattern =~ tr/ACGTRYMKSWHBVDNX/TGCAYRKMSWDVBHNX/;
my $regex;
   my $is_palindrome = 0;
   my $fwd_regex;
   # if the sequence is palindromic, a simple regex will do
   if ( $pattern eq $revpattern ) {
      $regex = regexify_ambig_codes($pattern);
      $is_palindrome = 1;
   }
   # if it is not, this is used to make the decision later which pattern,
   # forward or reverse complement, matches
   else {
      my $fwd = regexify_ambig_codes($pattern);
      my $rev = regexify_ambig_codes($revpattern);
      $regex = "$fwd|$rev";
      $fwd_regex = $fwd;
   }
   $regex = qr/$regex/;
   if ( $fwd_regex ) { $fwd_regex = qr/$fwd_regex/; }
   else { $fwd_regex = 0 }
   return $regex, $is_palindrome, $fwd_regex;
}
sub regexify_ambig_codes {
```

```
    my $seq = shift;
    my %ambig_codes = (
        'M' => '[AC]',
        'R' => '[AG]',
        'W' => '[AT]',
        'S' => '[CG]',
        'Y' => '[CT]',
        'K' => '[GT]',
        'V' => '[ACG]',
        'H' => '[ACT]',
        'D' => '[AGT]',
        'B' => '[CGT]',
        'N' => '[ACGTN]',
    );
    my $regex_string;
    for my $char ( split //, $seq ) {
        if ( exists $ambig_codes{$char} ) {
            $regex_string .= $ambig_codes{$char};
        }
        else {
            $regex_string .= $char;
        }
    }
    return $regex_string;
}
sub digest {
    my $seq    = shift;
    my $enzyme = shift;
```

```
my $start = 1;
my $seqlength = length $seq;
my @fragments;
my $regex = $enzyme->{pattern};
while ( $seq =~ /$regex/g ) {
   my $match_start = $-[0] + 1;
   my ( $end, $size, $new_start );
   # if the pattern for this enzyme is not a palindrome, forward and
   # reverse cases need to be handled differently, the have different
   # regexes (see make_pattern_regex).
   if ( $enzyme->{is_palindrome} == 1 ) {
      $end       = $match_start + $enzyme->{cut1_5} - 1;
      if ( $end < 1 ) { next } # no cutting outside of seq
      $size = $end - $start + 1;
      # some enzymes cut in two positions. The start of the next fragment
      # is then differnt from the end fo the current.
      # the piece cut out is disregarded.
      if ( $enzyme->{cut2_5} != 0 ) {
         $new_start = $match_start + $enzyme->{cut2_5};
      }
      else {
         $new_start = $end + 1;
      }
   }
   # pattern is not a palindrome, fwd and rev handled differently

   else {
      # this decides whether the forward or reverse complement of the pattern
```

```
    # matched, and cuts accordingly. Gettin a substring is much faster
    # than using $&.
    my $is_fwd;
    my $match = substr $seq, $match_start - 1, $enzyme->{len};
    if ( $match =~ /$enzyme->{fwd_regex}/ ) {
      $is_fwd = 1;
    }
    if ( $is_fwd ) {
       $end        = $match_start + $enzyme->{cut1_5} - 1;
    }
    else {
       $end        = $match_start + $enzyme->{cut1_3} - 1;
    }
    if ( $end < 1 ) { next } # no cutting outside of seq
    $size = $end - $start + 1;
    # enzyme cuts twice?
    if ( $enzyme->{cut2_5} != 0 ) {
       if ( $is_fwd ) {
          $new_start = $match_start + $enzyme->{cut2_5};
       }
       else {
          $new_start = $match_start + $enzyme->{cut2_3};
       }
    }
    else {
       $new_start = $end + 1;
    }
}
```

```
        # if the enzyme cuts out of the sequence, i.e. after the pattern,
        # the fragment does not count and there can not be any cuts hereafter
        if ( $new_start >= $seqlength ) { last }
        # keep fragment if it fits the given size limits
        if ( $size <= $max && $size >= $min ) {
            push @fragments,
              { 'start' => $start, 'end' => $end, 'size' => $size, };
        }
        $start = $new_start;
    }
    # last piece of seq
    my $end  = $seqlength;
    my $size = $end - $start + 1;
    if ( $size <= $max && $size >= $min ) {
        push @fragments,
          { 'start' => $start, 'end' => $end, 'size' => $size, };
    }
    return \@fragments;
}
sub calc_stats {
    my $sizes     = shift;
    my $seqlength = shift;
    my %stats;
    my $fragment_sum;
    my $n;
    for my $fragment (@$sizes) {
        $fragment_sum += $fragment->{size};
        $n++;
```

```
    }
    $stats{tot_fragment_length} = $fragment_sum ? $fragment_sum : 0;
    $stats{seq_length}          = $seqlength;
    $stats{fragment_count}      = $n ? $n : 0;
    if ($fragment_sum) {
        $stats{percentage} = $fragment_sum * 100 / $seqlength;
    }
    else {
        $stats{percentage} = 'NA';
    }
    return \%stats;
}

sub print_fragments {
    my $name  = shift;
    my $seq   = shift;
    my $sizes = shift;
    my $OUT   = shift;
    for my $fragment (@$sizes) {
        my $subseq = substr $seq, $fragment->{start} - 1, $fragment->{size};
        print $OUT ">$name" . ":$fragment->{start}..$fragment->{end}\n";
        print $OUT uc $subseq, "\n";
    }
}
sub print_stats {
    my $stats       = shift;
    my $enzyme_stats = shift;
    my $file        = shift;
```

```
   open my $STATS, '>', $file
     or die "Could not open file $file: $!\n";
   for my $name ( sort keys %$stats ) {
      print $STATS <<END
$name
Number of fragments:    $stats->{$name}{fragment_count}
Sum of fragment length: $stats->{$name}{tot_fragment_length}
Sequence length:        $stats->{$name}{seq_length}
% covered in fragments: $stats->{$name}{percentage}
END
        ;
   }
  print $STATS <<END
Totals
Number of fragments:    $enzyme_stats->{fragment_count}
Sum of fragment length: $enzyme_stats->{tot_fragment_length}
Sequence length:        $enzyme_stats->{seq_length}
% covered in fragments: $enzyme_stats->{percentage}
END
     ;
   close $STATS;
}
sub calc_enzyme_stats {
   my $stats = shift;
   my %enzyme_stats;
   for my $name ( sort keys %$stats ) {
      $enzyme_stats{fragment_count} += $stats->{$name}{fragment_count};
      $enzyme_stats{tot_fragment_length}
```

```
            += $stats->{$name}{tot_fragment_length};
        $enzyme_stats{seq_length} += $stats->{$name}{seq_length};
        if ( $enzyme_stats{tot_fragment_length} ) {
            $enzyme_stats{percentage}
                = $enzyme_stats{tot_fragment_length} * 100
                / $enzyme_stats{seq_length};
        }
        else {
            $enzyme_stats{percentage} = 'NA';
        }
    }
    return \%enzyme_stats;
}


sub print_summary {
    my $summaryfile = shift;
    my $stats       = shift;
    open my $STATS, '>', $summaryfile
        or die "Could not open file $summaryfile: $!\n";
    print $STATS "Enzyme\tfragment number\tpercentage of total sequence\tsum of
fragments\ttotal sequence length\n";
    for my $enzyme ( sort keys %$stats ) {
        print $STATS
            "$enzyme\t$stats->{$enzyme}{fragment_count}\t$stats-
>{$enzyme}{percentage}\t$stats->{$enzyme}{tot_fragment_length}\t$stats-
>{$enzyme}{seq_length}\n";
    }
    close $STATS;
}
```

```
1;
```

run.pl:

```
#!/usr/bin/perl


use Restriction_Digest;


#script to used Restriction_Digest per module.

#Note:rebase_e.txt, Restriction_Digest.pm and the genome file should all be in the same directory from which it is being called or paths should be specified correctly.


Restriction_Digest::findres($ARGV[0],"rebase_e.txt",$ARGV[1],$ARGV[2],$ARGV[3]);
```

digest.sh:

```
#shell script to digest the input genome file based on specified cut sites

#run.pl and perl module needs to be in current working directory


        work_dir=`pwd`
```

```
        cd $work_dir


###First enzyme

cd $work_dir/

perl run.pl $1 1000 3090 "PciI"

cd $work_dir/digest

mv $1_PciI.fasta $1_PciI.1.fasta

cd $work_dir/

perl run.pl $1 3091 5549 "PciI"

cd $work_dir/digest

mv $1_PciI.fasta $1_PciI.2.fasta

cd $work_dir/

perl run.pl $1 5550 9320 "PciI"

cd $work_dir/digest
```

```
mv $1_PciI.fasta $1_PciI.3.fasta

cd $work_dir/

perl run.pl $1 9321 100000000000000000000 "PciI"

cd $work_dir/digest

mv $1_PciI.fasta $1_PciI.4.fasta



cd $work_dir/

##

###second enzyme

perl run.pl $1 1000 3600 "BglII"

cd $work_dir/digest

mv $1_BglII.fasta $1_BglII.1.fasta

cd $work_dir/

perl run.pl $1 3601 6642 "BglII"

cd $work_dir/digest
```

```
mv $1_BglII.fasta $1_BglII.2.fasta

cd $work_dir/

perl run.pl $1 6643 11900 "BglII"

cd $work_dir/digest

mv $1_BglII.fasta $1_BglII.3.fasta

cd $work_dir/

perl run.pl $1 11901 100000000000000000000 "BglII"

cd $work_dir/digest

mv $1_BglII.fasta $1_BglII.4.fasta
```

8.2 DNA extraction from blood (minimal fragmentation)

1. *3M NaAc pH=5.3*
2. *95 % Ethanol (-20°C)*
3. *70% Ethanol (-20°C)*
4. *1x QUEEN'S LYSIS BUFFER (1L)*

- *1.21 g Tris (0.01M Tris-Cl)*
- *0.58g NaCl (0.01M NaCl)*
- *20ml EDTA (0.5M) pH 8.0 (0.01M EDTA)*
- *10g n-lauroylsarcosine (1% n-lauroylsarcosine)*
- *Adjust pH to 8*
- *Mix in 800ml ddH2O and bring vol. to 1L*

1. Take 20 µl of proteinase K and add 20 µl of blood
2. Adjust volume to 220 µl with PBS. Add 200 µl of buffer AL (from DNEasy blood and tissue kit) or use 200 µl of Queen's lysis buffer.
3. Incubate in rotor at 56˚C for 1 hour. Vortexing causes fragmentation.
4. Add 4 µl RNAseA and incubate at 37˚C for 30 minutes.
5. Add 200 µl of phenol-chloroform and mix by keeping in incubator at room temperature for 10 minutes.
6. Centrifuge at 13,000 rpm for 10 minutes and transfer the supernatant to another tube.
7. Add 200 µl of chloroform and mix by keeping in incubator at room temperature for 10 minutes.
8. Centrifuge at 13,000 rpm for 10 minutes and transfer the supernatant to another tube.
9. Add 0.1 volumes (approximately 20 µl) of 3M NaAc to each tube .
10. Add 2 volumes (approximately 400 µl) of 95% EtOH to each tube.
11. Turn the tubes upside down so that everything gets mixed in the tube. Mix gently!

12. Let it precipitate overnight in the fridge.
13. Centrifuge for 30 minutes at 13000 rpm.
14. Carefully remove the EtOH (all liquid) with a pipette and a KimWipe at the edge. If a large DNA pellet can be seen, you can pour off the EtOH instead of suction.
15. Add 200 µl 70% EtOH to wash the pellet (salt will dissolve in the water and the DNA will stay in pellet). Do not touch the pellet!
16. Centrifuge for 30 minutes at 13000 rpm.
17. Carefully remove the EtOH. If a large DNA pellet can be seen, you can pour off the EtOH instead of suction. The more EtOH that can be removed, the faster the pellet will dry. But it is also very easy to remove the pellet if not being careful!
18. Dry pellet completely by leaving the lid of the tube open in room temperature for 15 minutes.
19. Add 200 µl of TE buffer and let the pellet dissolve overnight in the fridge or on hot water bath at 50 C for 3 hours.

8.3 Reduced representation library construction

<u>DNA digestion:</u> Digest 80 ug of genomic DNA(concentration 125 ng/ul) using 5 units of restriction enzyme / ug of DNA.

10x NEBuffer3

10,000 U/ml = 10 U/µl (BglII)

- 100 µl reaction:
- mix 0.1 volume NEbuffer3 (10 µł) with DNA (in TE) with 10 µg of DNA (10 ug/ 80 ul)
- fill up with H_20 to 90 µl
- add 10µl (50 U) enzyme (10% of final volume and 5% of glycerol!)
- mix well ! (flicking, gently pipetting with tip cut-off)
- incubate at 37°C for 2 h (No BSA)

Getting rid of the enzyme: phenol chloroform

- Add 1:1 volume phenol chloroform (200 ul) to 200 ul(pool 4 tubes) of the reaction digest.
- Centrifuge and transfer supernatant to new tube.
- Add 1:1 volume chloroform (200 ul) to 200 ul of supernatant.
- Centrifuge and transfer supernatant to new tube.
- After phenol chloroform extraction we reduce the volume to fit into the gel lanes(10 ug /50 ul) by heating the supernatant at 70 °C.

Gel electophoresis

Digested genomic DNA was run on a 1% low melt agarose gel (UltraPure LMP(low melting point) Agarose invitrogen Cat. no. 16520-050) to separate the frgaments based on size. Two markers [marker 1:(Gene Ruler High Range DNA Ladder Fermentas #SM1351 +O RangeRuler 500bp DNA Ladder Fermentas #SM0643)

and marker 2:(Lambda DNA/Eco91I(BstEII) Marker, 15)Fermentas #SM0111

)] are run beside the genomic DNA digest to indentify size of size of DNA fragments.

1x Loading dye:

- *2.5 % Ficoll 400*
- *11 mM EDTA*
- *3.3 mM Tris-HCl*
- *0.017 % SDS*
- *0.015 % Bromophenol Blue*

1% low melt agarose gel was run at 50 volts for 200 minutes to ensure proper separation of the fragments based on size with no fragmentation. Using low melt agarose makes the extraction of DNA from the gel more effecient. Based on few trials it was determined that 5 ug/lane of DNA at a concentration of 140 ng/ul could be used with a ratio of 1 (loading dye) : 5 (DNA). DNA was stained by keeping the gel in a Ethidium Bromide bath for 10 to 15 mins and visaulised with Ultraviolet light.

DNA purification with QIAEXII (Qiagen) Cut out the different libraries using the markers to indetify the selected size range. Within each library vertical pieces are cut out and purified to obtain 40 ul for each piece. Each library should have atleast 2 ug in the end.

1. Add 3 volumes of Buffer QX1 and 2 volumes of H2O to 1 volume of sample.
2. Check that the color of the sample mixture is yellow.
3. Resuspend QIAEX II by vortexing 30 s.
4. Add 10 µl of QIAEX II per 5 µg of DNA and mix(using incubator for 20 mins[30c]).

5. Centrifuge the sample for 2 mins and remove supernatant.
6. Add 500µl of Buffer PE and mix(in incubator for 20 mins[30c]). Centrifuge for 2 mins and remove supernatant.
7. Again perfrom step 6.
8. Air-dry the pellet for 10–15 min.
9. Add 20 µl of 10 mM Tris·Cl, pH 8.5 and mix (in incubator for 20 mins[50c])
10. Centrifuge for 2 mins. Remove the supernatant with DNA into a clean tube.
11. Perform step 9 and 10 again and add the supernatant together.

8.4 List of figures

8.5 List of tables

8.6 List of Acronyms

BP : Basepair

DNA : Deoxyribonucleic acid

Gb : Giga-basepairs

GRHR : Gene Ruler High Range DNA Ladder

Kb : Kilo-basepair

Mb : Mega-basepair

NGS : Next-generation sequencing

OR : O RangeRuler 500bp DNA Ladder

RRL: reduced representation library

WGS: whole genome shotgun

9 Acknowledgments

I would like to thank my supervisor Dr. Jochen Wolf, for all the patient instruction, inspiration and many hours discussing the various ways to solve the problem at hand. It has been an exciting, adventurous and marvellous learning experience. Many thanks to Axel Künstner, Linnea Smeds and Dr. Páll Ólason for suggestions and guidance with genome assembly.

The computations were performed on resources provided by SNIC through Uppsala Multidisciplinary Centre for Advanced Computational Science (UPPMAX) under Project b2010060 and b2010059. I would like to thank the UPPMAX support team for help with running the very long jobs with very high memory requirements.

Thanks to Masters Coordinator Dr. Margareta Krabbe for guidance throughout the masters program.

I am also grateful for the timely help and support given by Deepak Menon, Mukil GP, Shanoob K T, Vignesh Thiyagarajan and others at TCS - Kochi. Thanks to the flexibility offered by TCS (TATA Consultancy Services), I could take long leave and complete my master's degree.

I wish to thank my mother Dr Shantala Priyadarshini for her encouragement, steadfast belief and support.