Transforming Genomes Using MOD Files with Applications

Shunping Huang
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599, USA
sphuang@cs.unc.edu

Chia-Yu Kao
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599, USA
katykao@cs.unc.edu

Leonard McMillan
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599, USA
mcmillan@cs.unc.edu

Wei Wang
Dept. of Computer Science
University of California
Los Angeles, CA 90095, USA
weiwang@cs.ucla.edu

ABSTRACT

Next generation sequencing techniques have enabled new methods of DNA and RNA quantification. Many of these methods require a step of aligning short reads to some reference genome. If the target organism differs significantly from this reference, alignment errors can lead to significant errors in downstream analysis. Various attempts have been tried to integrate known genetic variants into the reference genome so as to construct sample-specific genomes to improve read alignments. However, many hurdles in generating and annotating such genomes remain unsolved.

In this paper, we propose a general framework for mapping back and forth between genomes. It employs a new format, MOD, to represent known variants between genomes, and a set of tools that facilitate genome manipulation and mapping. We demonstrate the utility of this framework using three inbred mouse strains. We built pseudogenomes from the mm9 mouse reference genome for three highly divergent mouse strains based on MOD files and used them to map the gene annotations to these new genomes. We observe that a large fraction of genes have their positions or ranges altered. Finally, using RNA-seq and DNA-seq short reads from these strains, we demonstrate that mapping to the new genomes yields a better alignment result than mapping to the standard reference.

The MOD files for the 17 mouse strains sequenced in the Wellcome Trust Sanger Institute's Mouse Genomes Project can be found at
http://www.csbio.unc.edu/CCstatus/index.py?run=Pseudo
The auxiliary tools (i.e. MODtools and Lapels), written in Python, are available at http://code.google.com/p/lapels/.

BCB '13, September 22 - 25, 2013, Washington, DC, USA

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics

General Terms

Experimentation, Performance

Keywords

Pseudogenomes, Reference, RNAseq, DNAseq, Alignment, Reference bias

1. INTRODUCTION

High-throughput sequencing technologies have enabled a multitude of new quantitative sequence analysis methods, many of which are based upon an initial alignment to some reference genome. By *quantitative* we mean methods where depth of coverage and the consistency of reads at a given genomic position factor into some measure of interest. In the case of RNA-seq analysis, this includes estimation of relative or absolute transcript abundance [18, 29], assessing parent-of-origin effects [9, 5], and estimating RNA-editing rates [21, 20]. Whereas ascertaining copy-number and calling genomic variants are common DNA-seq quantitative analysis examples.

A common prerequisite of nearly all sequence analysis pipe-lines is to align read fragments to a high-quality reference genome sequence. Typically this reference genome is of a genetically close organism with the same karyotype and genomic arrangement as the target organism. Moreover, a large amount of annotation effort has generally been applied to the reference genome. In particular, the placements and extents of genes and exons [6], functional elements [27], and genetic variants [26, 12] are given in coordinates relative to a reference genome. As the genetic distance between the reference and target genomes increases the quality of the alignment decreases as measured by the numbers of unmapped and misaligned fragments.

Aligning reads to a reference genome can also introduce local alignment biases, (i.e., regions that better match the reference sequence tend toward higher coverage than regions with variations [3, 19]) which confounds downstream quantitative analyses. An obvious alternative is to incorporate all a priori known variations from the target into a new reference that is used for alignment. The problem with this

approach is that it is hard to represent genomic positions and leverage existing genomic annotations for these synthetic genomes. Positions of the reference will be shifted due to indels and other structure variations; likewise, newly inserted sequences might have no corresponding coordinate in the original reference genome.

Nonetheless, many researchers have addressed this issue by incorporating variants, to various extents, into a *pseudo*-reference genome sequence. Incorporating only single nucleotide polymorphisms (SNPs) is commonplace and straightforward [24], since it does not change the coordinates of the constructed genome sequence. Alternatively, Degner *et al.* [3] masked every known polymorphic location in the reference genome by introducing a third allele, thus increasing the genetic distance between the target and the reference in an unbiased fashion. However, these methods only make use of the SNP data, and they are not applicable to other types of variants, such as indels. In general, sequence aligners are less sensitive to the point errors caused by SNPs than to the frame shifts introduced by indels. Therefore, the impact of incorporating only SNPs into a new alignment reference sequence is probably minimal.

There have also been some attempts to utilize other variants besides SNPs. Rivas-Astroza *et al.* [22] developed a software (perEditor) to build a personal genome with different variant types, but they only focussed on genome construction without resolving the coordinate inconsistency after read alignment. The analysis pipeline for allele-specific gene expression and binding, proposed by Rozowsky *et al.* [23], generated both individual personal diploid genomes and equivalence maps. However, these maps of equivalent positions are only employed to map gene annotation between the reference and individualized genome, while the read alignments remain unchanged.

In this paper, we first propose a general-purpose framework for mapping back and forth between genomes, that is suitable for both short reads and genomic annotations. It is facilitated by a mapping file, called a MOD file, that describes all variations between a reference and a target genomic sequence. MOD files provide a generative mapping from reference sequence to a target sequence that incorporates all known structural variants (SNPs, indels, translocations, and inversions). We call these newly generated genomes, which will be used in place of the standard reference during the alignment process, **pseudogenomes**. Mapping positions back and forth between the reference and the pseudogenomes becomes convenient using MOD files, which can be applied to both gene annotation and remapping of read-fragment alignments. Not only are the start and end positions of each fragment remapped (essentially realigned), but the MOD file provides the information necessary to modify the associated CIGAR string [16] for each fragment. This remapping back to the reference coordinates is a crucial step after alignment to a pseudogenome.

In summary, we propose a mechanism for mapping and remapping between genomes that is composed of a single file per target and a set of tools that use and interpret this file. Using it, we are able to

1. efficiently construct pseudogenomes for use in short-sequence alignments so as to overcome the problems associated with reference bias,
2. map positions and intervals between pseudogenomes and a reference genome, so that we can continue to utilize annotations rooted in the reference coordinate system, and
3. manipulate multiple genomes easily, as a result of the properties of the MOD format.

2. METHODS

2.1 Design of the MOD format

The MOD format is composed of instructions that transform one genome sequence into another. It is essentially an edit transcript relating two strings [10], and it provides a basis for quantifying the similarity of two sequences. A MOD file is not necessarily unique, nor do we make any claims with regard to minimality. We call the genome before transformation **the source** and the one after **the destination**. Each MOD file is directional, i.e. always from the source to the destination.

A MOD file consists of two parts (Fig. 1a): a header and a body. The header includes the metadata of the transformation, such as, the version of the MOD format, the source, the destination, and so forth. The body holds the instructions, each of which has its affected position and arguments. Positions are all stored in the source coordinate system, and the bases before and after modification are included in the arguments.

There are three basic types of instructions defined in the MOD format: **s-**, **d-**, and **i-instructions**. They describe single-base substitutions, single-base deletions, and insertions, respectively. All instructions are **atomic**, in that they reference no more than one position from the source. It is obvious that both s-instructions and d-instructions are atomic. For i-instructions, we merely add new sequence after an anchor position in the source without altering any base; thus they are also atomic.

One way to generate a MOD file is to convert common variant calls into instructions. For example, SNPs and genomic insertions can be directly changed to s-instructions and i-instructions, respectively. For genomic deletions, we need to break each of them up into single-base deletions before converting to d-instructions (Fig. 1a and 1b). Notice that the position information in adjacent d-instructions is redundant. However, the design choice of keeping all instructions atomic facilitates later MOD-file manipulations, whose advantages are considered to outweigh this slight redundancy. Moreover, the additional space overhead is recovered when MOD files are compressed.

Complex structure variants, such as tandem duplications, inversions, and translocations, can be described by the current set of instructions. For example, a tandem duplication is represented by repeating an i-instruction at the same location, while inversions (or translocations) are implemented by a series of d-instructions at the source sequence position and a corresponding i-instruction of the inverted (or transferred) sequence at its new position. We recommend annotating such coupled sets of instructions using comments following the instructions.

We can also derive new MOD files from other MOD files by leveraging various properties of the format. This will be discussed in Section 2.3 .

2.2 Properties of the MOD format

2.2.1 *Invertibility*

A MOD file specifies all changes from the source genome to the destination genome; this includes bases both before and after each change. Therefore, we are able to exchange the source and the destination by inverting the instructions in the file (Fig. 2).

For example, each s-instruction specifies a position and a nucleotide from the source and its replacement nucleotide in the destination, so it can be inverted by merely swapping the two nucleotides. The d-instructions contain bases that they remove from the source, so inverting them will result in inserting these deleted bases back to the destination, i.e. i-instructions. Moreover, since d-instructions are restricted to be one-based but i-instructions are not, adjacent i-instructions can be combined into one as an optimization. Similarly, i-instructions are broken up into multiple d-instructions during inversion.

The position of each instruction must also be modified when inverting a MOD file. Positions in the source coordinate system are changed into destination coordinates in the output as described in Section 2.3.2 .

2.2.2 *Concatenability*

The MOD files with the same source genome can be concatenated. In other words, we can combine a prefix sequence generated from one MOD file with a suffix sequence from another MOD file without messing up the coordinates or missing any variants on the segment boundaries, as long as the two MOD files have the same source (Fig. 2). Concatenation is used to construct pseudogenomes for hybrid organisms (e.g., F2s and backcrosses) to account for recombinations.

Concatenability results from the use of atomic instructions in a single source coordinate system. Given a genomic region, every instruction in a MOD file will be either inside or outside the region; there is no case where an instruction crosses a region boundary. Therefore, unlike variant calls that require special care in the boundary cases, MOD files can be safely cropped.

If the cropped regions from different MOD files are disjoint, their instructions can simply be stacked together; otherwise, it is possible that some positions may be affected by

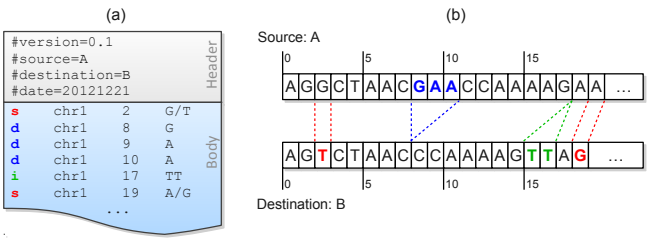

Figure 1: A MOD file example (a) and the corresponding sequences of the source and the destination (b). There are two SNPs between these sequences, and they are represented as two s-instructions at source positions 2 and 19. A three-base deletion (from source positions 8 to 10) is observed, and it is broken up into three d-instructions. The insertion after position 17 is directly added to the MOD file without any conversion due to its atomicity.

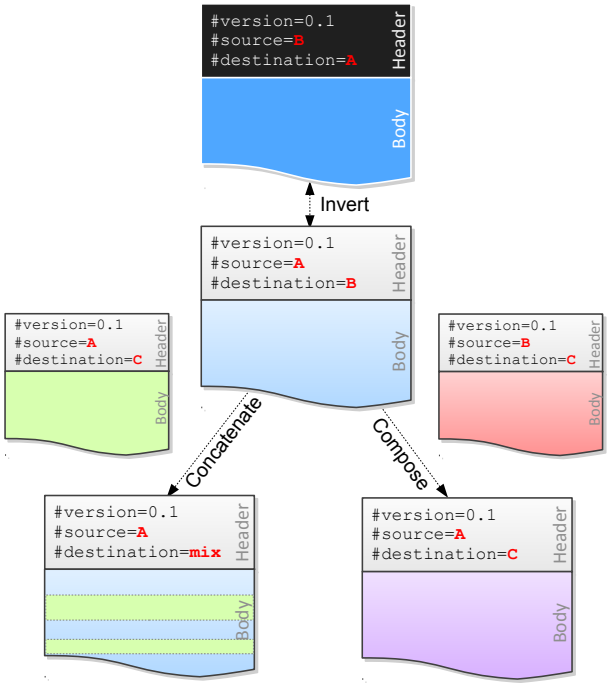

Figure 2: The three main properties of the MOD format enable a wide range of operations on MOD files. The original MOD file is in the middle, with A as the source and B as the destination. Inversion only involves one MOD file, and it will exchange the source and the destination (top part). Concatenation and composition, however, need two or more MOD files. Concatenating two MOD files with the same source (A in the figure) will generate a mixed destination (bottom left part). Composing two MOD files, from A to B and B to C, can be considered as transferring from A to C through B (bottom right part).

more than one instruction. When multiple instructions refer to the same genomic coordinate, several rules are used as the tiebreaker. Generally speaking, if instructions of different types are performed on the same location, d-instructions take precedence over s-instructions. If instructions have the same type, only the last one will be used. Notice that there is no preference between i-instructions and other instruction types, because insertions have no footprint in the source, and therefore will not contradict other instructions. If two or more i-instructions specify the same position, they are added in order.

2.2.3 *Composability*

Given two MOD files containing instructions to transform genome A to genome B and genome B to genome C, respectively, one can construct a MOD file transforming genome A to genome C (Fig. 2). This property is called composability.

Let $\mathbf{P}(A \mapsto B)$ and $\mathbf{Q}(B \mapsto C)$ represent the two MOD files to be composed, and $\mathbf{R}(A \mapsto C)$ be the resulting MOD file. The procedure of composition is briefly described as follows. First, we invert $\mathbf{P}$ to obtain $\mathbf{P}'$, which contains instructions mapping from genome B to genome A. Second, we compute the intersection of $\mathbf{P}'$ and $\mathbf{Q}$, i.e., $\mathbf{P}' \cap \mathbf{Q}$. These

shared instructions indicate that A and C are identical at the corresponding positions, because same changes should be made to change B to A and B to C. Third, we remove the intersection from $\mathbf{P}'$ and $\mathbf{Q}$ obtaining $\bar{\mathbf{P}}'$ and $\bar{\mathbf{Q}}$ separately. These two MOD files are the actual difference between A and C. Finally, we map the instruction positions in $\bar{\mathbf{Q}}$ from genome B coordinates to genome A coordinates (described in Section 2.3.2), and combine the result with the inversion of $\bar{\mathbf{P}}'$. This gives the expected MOD file $\mathbf{R}$.

2.2.4 Other properties

In addition to these three properties, the MOD format has other virtues. For example, it can be easily converted from the VCF format [2], which is commonly used to store variant calls. Also, the MOD files can be compressed by bgzip [16] and indexed by tabix [15], so that the file sizes are reduced and they can be efficiently queried.

It is also convenient to edit a MOD file to incorporate new variants or to mask obsolete ones. Since all positions are in the same coordinate system for each MOD file, there is no need to worry about adjusting positions when adding or removing variants.

2.3 Use of the MOD format

2.3.1 Pseudogenome construction

MOD formatted files provide a generative procedure for transforming a source sequence to a destination; thus, they are ideally suited for constructing an in silico target genomic sequence from a given reference. We call this generated genome a **pseudogenome**.

One can easily construct MOD files for entire catalogs of common inbred strains using readily available variant calls. The property of concatenability makes it convenient to create the pseudogenomes for arbitrary crosses between inbred strains and recombinant inbred lines (RILs)[25]. The genomes of RILs are a mosaic of two or more founder genomes. Once the haplotype structure of a RIL is inferred [17, 7], one can concatenate the regions of instructions from founder MOD files to form a new MOD file for the RIL, which can then be utilized for alignments.

When using MOD files we often assume a common source genome, or reference, but this restriction is unnecessary. The MOD format can be used to map between any two genomes or genome versions, allowing the source sequence to be transformed to any destination sequence.

2.3.2 Position mapping

The MOD format also provides the capability to map coordinates or intervals from the source to the destination, and vice versa. This is done by scanning a MOD file and accumulating the number of shifted bases affected by d-instructions and i-instructions. For every pair of corresponding regions in the two genomes, we record a pair of offsets. Given a position in the source, we first look up in the source offsets to find out in which region it falls, and then compute its destination position.

The invertibility of the MOD files guarantees that we are able to map positions back and forth between the source and the destination. The composability also extends the mapping ability. For example, given two MOD files, from the reference to two non-reference strains, we can invert one and compose them to get a third. With the help of this MOD file, we are able to map positions between the two non-reference strains.

Position mapping can be applied to genome annotations, which is usually presented in the reference coordinate system, to get a new target-specific annotation. Position mapping can also be applied to genome alignment results, so we can remap the alignments from one genome to another as described in the following pipeline.

2.3.3 The alignment pipeline for inbred strains

Traditionally, DNA-seq or RNA-seq read fragments are align-ed to a reference genome or transcriptome sequence, which is subject to a reference bias. Here we propose a new pipeline for aligning reads to other inbred strains.

Executing the instructions of a MOD file for an inbred strain, incorporates variants into the reference genome sequence to obtain a pseudogenome. Reads can then be mapped to the pseudogenome using an alignment tool such as Tophat [28] or Bowtie [14, 13]. The aligned read file (typically a BAM file) can then be remapped back to the reference genome's coordinates for analysis using the same MOD file. We have developed a tool for this purpose called *Lapels*, which remaps the positions of the read fragment alignments, modifies their associated CIGAR string, and annotates the observed variants seen in each fragment (Fig. 3) .

The reason for remapping alignments back to reference coordinates is twofold. First, there are abundant resources specified in the reference coordinate system, including databases of genetic variants, gene/exon annotations, and catalogs of other functional genomic elements. Second, many studies involve multiple strains. It is convenient to have a common coordinate system so that comparisons between strains become feasible.

3. RESULTS

In our experiments, we used the mouse as the model organism, but the MOD format, the tools, and the analysis pipeline we propose are also applicable to other organisms.

We used three wild-derived inbred mouse strains in our experiments: CAST/EiJ, PWK/PhJ, and WSB/EiJ, all of which are highly diverged from the *Mus musculus* reference genome derived largely from C57BL/6J. The SNP and indel variants for these strains were downloaded from the Wellcome Trust Sanger Institute [12], while the mouse reference genome data is from NCBI MGSCv37.

To generate MOD files for the three target strains, we first extracted SNPs and indels from the VCF files (downloaded from `ftp://ftp-mouse.sanger.ac.uk/REL-1105/`). Only
high-confidence SNPs and indels for the 19 autosomes and X were incorporated into the MOD files. Variants on Y and mitochondria (M) were extracted from other sources (`http://cgd.jax.org/datasets/popgen/diversityarray/yang2009.shtml`). The MOD files used in this paper can be found at `http://www.csbio.unc.edu/CCstatus/index.py?run=Pseudo`. For each MOD file, the statistics for the whole genome are summarized in Table 1 .

The total number of bases involved in all instructions of a MOD file can be used as an estimation of genomic distance between a strain and the reference. Fig. 4 shows such distance for the three strains studied. The CAST strain is the most distant genetically from the reference and the WSB strain is the genetically closest to the reference.

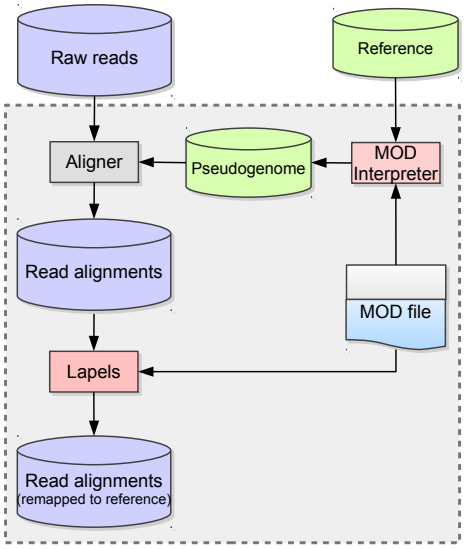

Figure 3: A new alignment pipeline for inbred strains. The standard reference genome, which is used directly for alignment in the traditional pipeline, is replaced by a MOD file generated pseudogenome. The MOD file, which captures variants between the reference and the inbred strain, is used twice, once to construct the pseudogenome used for alignment, and next to map the aligned reads back to their position in the reference genome where existing annotations can be leveraged.

To illustrate the density of the genomic variants (currently described in the MOD files) of the three strains and their potential impact to read alignment, we divided the pseudogenome into 100bp windows and counted the number of bases that are modified by any instruction in each window. In about 11.72 % of windows three or more bases are affected by CAST/EiJ variants, while the percentages are 11.22% and 3.80% for PWK/PhJ and WSB/EiJ, respectively. In general, high-variant windows are uniformly distributed along the genome, suggesting that, if we use the reference genome for read alignment, the alignment quality may be substantially compromised over the entire genome. We show the distribution of the counts for CAST/EiJ in Fig. 5.

3.1 Position mapping on genetic annotation

In this study, we constructed strain-specific gene annotations for the CAST, PWK and WSB pseudogenomes, and investigated how many exons, transcripts, and genes were changed after variants were incorporated in the reference.

The gene annotation of mm9 was from Ensembl [6]. There are, in total, 688,311 exons, 97,251 transcripts, and 37,620 genes in the latest release (release 67).

To accomplish this, we developed a tool, *modmap*, for mapping positions and intervals from source to the destination. Modmap takes a MOD file and an annotation file as input. It first builds a position mapping between genomes internally and then changes the annotation file's position columns from source coordinates to destination coordinates. In our current setting, the source is the reference genome, and the destination is the pseudogenome.

After the strain-specific annotation was obtained, we compared it with the original annotation in terms of the start positions and the ranges of exons, transcripts, and genes.

Not surprisingly, because of the integrated indels and structure variants, the start positions of almost all exons, transcripts and genes (over 99%) are shifted in the pseudogenome annotation. In addition, about 6% of exons, 78% of transcripts and 62% of genes have different lengths (Table 2) in the CAST pseudogenome. There is a strong correlation between the number of changes in a strain and the genetic distance of the strain from the reference. (Fig. 4)

It is worth noting that the pseudogenome annotations could still be inaccurate. In fact, Gan *et al.* [8] recently

Strain	s-instructions	d-instructions	i-instructions
CAST	17,674,364	4,834,899	4,206,776
PWK	17,202,935	4,715,249	3,457,436
WSB	6,045,875	2,026,461	1,579,714

Table 1: Statistics of MOD files for CAST/EiJ, PWK/PhJ and WSB/EiJ. The counts are in units of base-pairs. For s-instructions and d-instructions, they are just the numbers of instructions, respectively. For i-instructions, the counts are derived from adding up the number of bases in each inserted sequence.

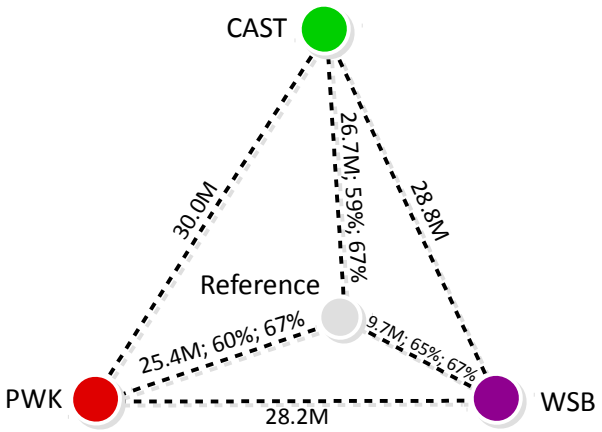

Figure 4: The estimated genetic distances between the reference and each of the three strains tested. The first number on edge label is the total number of MOD file instructions necessary to transform the reference into the target strain's pseudogenome. This number is proportional to the edit distance between the two genome sequences. The second number represents a typical percentage of reads from the target that can be uniquely aligned using the reference sequence. Notice that these numbers are inversely proportional to the strain's genetic distance from the reference, and they illustrate the so-called reference bias. The third number is a typical percentage of those reads that uniquely align to a MOD-file generated pseudogenome.

Strain	Exons	Transcripts	Genes
CAST	5.98%	78.04%	62.37%
PWK	5.87%	76.68%	61.42%
WSB	3.01%	65.76%	52.75%

Table 2: Percentage of exons, transcripts, and genes that have different lengths in the pseudogenomes.

raised the issue that simply mapping gene annotations back and forth may lead to incorrect annotations. Nevertheless, MOD-file derived pseudogenomes can serve as a first-cut approximations to facilitate initial reannotations while also supporting efficient remapping back to reference. As a more accurate picture of the actual genomic structure develops it can easily be incorporated into the MOD-file.

3.2 RNA-seq read alignment of inbred mice

The mouse samples we sequenced were derived from the aforementioned wild-derived mouse strains. We sequenced >1.2G reads on the Illumina HiSeq 2000 platform of mRNA from the brain tissue extracted from 12 samples (4 samples per strain) using paired-end reads with 100 bp (2x100). The number of reads per sample is shown in Table 3 .

For each strain, we applied the pipeline described in Section 2.3.3 . We used TopHat (v.1.4.0), with default parameter settings, to map reads to pseudogenomes derived from MOD files and the NCBI MGSCv37 *Mus musculus* reference genome. After aligning, the read fragments from the resulting BAM files were remapped back to the reference genome and tagged with the number of observed variants (i.e., the number of variants incorporated in the pseudogenome and observed in the read). Based on the number of alignments in the resulting BAM file, we categorized each read into one of the three classes: unmapping (0), unique mapping (1), and multiple mapping (>1).

We also aligned the same reads to the standard reference genome and compared them to the reads mapped to the pseudogenome. The percentage of reads by category and the average percentages of biological replicates are shown in Table 4 .

Observe that more reads map uniquely to the pseudogenome than to the reference (shaded cells). The percentage increase is 7.46% for CAST, 6.76% for PWK, and 2.38% for WSB.

On one hand, the percentage of reads that uniquely map to the reference increases as we move from CAST (59.35%) to PWK (60.43%) to WSB (65.02%). This again accurately reflects the genetic distance of each strain from the reference (Fig. 4) . The more different that a strain is from the reference, the fewer of its reads are mapped to reference genome, thus illustrating a reference bias.

In contrast, the percentage of reads that uniquely mapped to the pseudogenome is consistent among the three strains (around 67%). Thus, with pseudogenomes, the different strains are brought to comparable levels of mappability, which implies that the reference bias has been largely diminished. In all samples around 30% of reads remain unmapped, we believe this residual represents current limitations of the aligner in combination with the remaining inaccuracies in the pseudogenome sequence.

Notice that the largest percentage increase is due to reads that are mapped uniquely to the pseudognome but unmapped to the reference (cells with pink shading). This suggests that our pipeline has rescued many reads that are discarded in the traditional method, especially when the strain is distant from the reference.

In order to understand how the embedded variants affect the alignment result, we investigate what percentage of reads have observed variants in the each categories. The result is shown in Table 5 . Notice that categories involving unmapped reads in pseudogenomes are not included in the table, because such reads have no alignment and, thus, no observed variants of the strains.

Most of the reads that were mapped uniquely to the pseudogenome but unmapped in the reference alignments have variants: 88.74% for CAST, 88.08% for PWK, and 84.25% for WSB. Similarly, around 78% of the reads that were unmapped in the reference alignment, but mapped to multiple positions in the pseudogenome, contained a strain-specific variant. Such unmapped-to-multi-mapped reads, however, account for less than 0.01% of all aligned reads. Another group (0.05-0.1%) of the reads mapped to multiple positions in the reference alignment became unique mapping in the pseudogenome alignment owing to the added variants. In short, by incorporating the variants in the reference, we have provided the reads a better genome sequence to map to; such reads are neither thrown away nor misplaced by the aligner in the new pipeline.

Of those reads that mapped uniquely in both the reference and pseudogenomes, only a small fraction of them contain the strain-specific variants. This is to be expected since there are plenty of conserved regions in the genome. If reads originate from one of these regions, it is possible that they will be mapped uniquely to both genomes without any observed variant. Moreover, both alignments of this kind of reads should have the same mapping position, because we remapped the pseudogenome-aligned reads back to the reference coordinate system. In order to verify our hypothesis, we looked further into this category of reads by comparing the two alignments in terms of their positions and CIGAR strings. As shown in Table 6 , over 99% of reads in this category have the same mapping position and CIGAR string in both alignments, which justifies our previous assumption. We also observed that a large portion of remaining reads contain variants, the percentage of which is shown in parenthesis. Lastly, we find that for such reads fewer mismatches are seen in the pseudognome alignments than in the reference alignments, suggesting a better alignment result is achieved by using the pseudogenome.

3.3 DNA-seq read alignment of inbred mice

Our alignment pipeline using pseudogenomes can also be used to align DNA-seq short reads. In this experiment, the DNA-seq data set was provided by the Wellcome Trust Sanger Institute (`ftp://ftp-mouse.sanger.ac.uk/current_bams/`), in which mouse strains were sequenced with Illumina HiSeq platform with over 40-fold coverage. Reads are 100bp paired-end.

We extracted the raw reads from the BAM file of CAST/EiJ and obtained 646,514,920 reads in total. Then we realigned them to both the standard reference genome and the CAST pseudogenome using Bowtie (v.2.0.5). Default parameter settings were used, and only the best alignment for each read was reported. The comparison of read alignments to the pseudogenome and the reference genome is shown in Table 7 .

Table 3: Number of reads for 12 samples in 3 strains.

Strain	Sample 1	Sample 2	Sample 3	Sample 4	Total
CAST	88,520,554	78,903,440	78,976,480	135,080,364	381,480,838
PWK	140,642,004	96,388,598	96,735,248	132,859,376	466,625,226
WSB	130,888,744	123,814,138	66,178,922	92,044,920	412,926,724

Table 4: Average percentage of RNA-seq reads from CAST, PWK, and WSB samples mapped to the pseudogenome and the reference.

Alignment		**Reference**			
		Unique	Multiple	Unmapped	Total
CAST Pseudogenome	Unique	58.91%	0.10%	7.80%	66.81%
	Multiple	0.11%	2.31%	0.01%	2.43%
	Unmapped	0.33%	0.01%	30.42%	30.76%
	Total	59.35%	2.42%	38.23%	100.00%
PWK Pseudogenome	Unique	60.01%	0.09%	7.09%	67.19%
	Multiple	0.14%	2.37%	0.01%	2.52%
	Unmapped	0.28%	0.01%	30.00%	30.29%
	Total	60.43%	2.47%	37.10%	100.00%
WSB Pseudogenome	Unique	64.83%	0.04%	2.53%	67.40%
	Multiple	0.06%	2.47%	$< 0.01\%$	2.54%
	Unmapped	0.13%	$< 0.01\%$	29.93%	30.06%
	Total	65.02%	2.52%	32.46%	100.00%

Table 5: Average percentage of RNA-seq reads with observed variants from CAST, PWK, and WSB samples in each category.

Alignment		**Reference**			
		Unique	Multiple	Unmapped	Total
CAST Pseudogenome	Unique	21.50%	64.59%	88.74%	29.58%
	Multiple	74.99%	2.74%	78.80%	6.10%
PWK Pseudogenome	Unique	21.68%	61.38%	88.08%	28.85%
	Multiple	76.46%	4.01%	77.48%	8.22%
WSB Pseudogenome	Unique	7.79%	56.51%	84.25%	10.71%
	Multiple	75.96%	2.42%	78.37%	4.40%

Table 6: Comparison of mapping positions and CIGAR strings for reads uniquely mapped to both genomes.

Pseudogenome Strain	Unequal Start	Equal Start but Unequal CIGAR	Equal Start and CIGAR
CAST	0.32% (82.43%)	0.41% (97.70%)	99.26% (20.98%)
PWK	0.32% (79.78%)	0.39% (97.41%)	99.29% (21.20%)
WSB	0.12% (79.03%)	0.20% (97.62%)	99.68% (7.53%)

Most reads (over 95%) aligned to both genomes, which suggests that the aligner, in general, compensates for differences in genomic sequences. However, the pseudogenome, recovers about 0.70% more reads. Notice that this number is much smaller than the recovery rate of the RNA-seq experiments described in Section 3.2 (i.e., 7.48%). We speculate that the reason for this is two-fold. First, without splice junctions, DNA-seq reads are intrinsically easier to align than RNA-seq reads. The 100bp DNA-seq reads have a higher mappability than the short fragments that are separated by exon junctions as in RNA-seq reads. Second, the tolerance of mismatches and gaps is different between Bowtie and Tophat. With default settings, as used in our experiments, Bowtie allows more mismatches than Tophat.

We further investigated the 95% of reads by grouping them based on alignment positions and edit distances. For each read, we compared the alignments to the pseudogenome with those to the reference genome. Alignments were considered to occupy the same position only if their starting positions and CIGAR strings were identical. The edit distance is represented by the NM tag of each alignment. The results are shown in Table 8 . On one hand, 91% of the reads aligned to exactly the same position. While over 54% of the reads have no mismatches or an equal edit distance in the two alignments, about 36% of the reads aligned to the reference genome have a larger edit distance than the same read's alignment to the pseudogenome. It is worth mentioning that this percentage may vary for different aligner parameter set-

tings and may also be influenced by noise. On the other hand, around 4.7% of the reads have different alignment locations. Since we have incorporated known variants into the pseudogenome, it is expected that the alignment positions in the pseudogenome are more accurate than those in the standard reference.

Although our pipeline does not directly rescue as many reads in the DNA-seq data as it did in RNA-seq data, it greatly increases the robustness and accuracy of the alignment results.

4. DISCUSSION

In this paper, we propose an approach for representing the mappings and variations between genomes as a generative procedure, which we call a MOD file. MOD files can be used to directly transform a reference genome into any other target sequence. We have also developed a set of auxiliary tools (http://code.google.com/p/lapels/) that use MOD files to perform a wide range of useful genomic transformations. Moreover, we integrate the MOD files into the traditional Hi-seq alignment pipelines to improve mapping quality and reduce the biases inherent in any reference based approach.

We should point out that despite its incorporated variants, a pseudogenome may still differ significantly from the actual one, because to some extent we are limited by our current knowledge of the genomic variation. It can be predicted that as we gain more understanding of the variants between strains, the pseudogenomes will become more accurate and useful. However, results from the experiments suggest that incorporating known variants into a reference sequence used for alignment has considerable benefits, and always outperforms a standard reference. The primary impediment of incorporating variants into a genome has been the difficulty in relating the result to various annotations defined for the reference. This is precisely utility of MOD files; they provide a simple means of mapping to and from a related pair of sequences.

Several existing databases and file formats have similarities to our proposed MOD file description. The VCF format, for example, is widely used to annotate variant information. The main difference between the MOD file and the VCF format is the content: a MOD file describes a mapping to transform genomes, while a VCF file catalogs only sequence variants. Also, the VCF format supports variant calls for multiple sequences in a single file, whereas the MOD format relates only two sequences, i.e., the source and the destination. Furthermore, other information of variants, such as genotype quality, allele frequency, and read depth, can be placed in the VCF files; whereas MOD files contain only the necessary metadata and instructions for transforming one genome sequence to another. The UCSC Chain format and LiftOver tool [1] are widely used to convert genome coordinates between assemblies, while the delta file from the MUMmer package [4] is designed to contain encoded representation of coordinate and distance for pair-wise alignments. However, both formats are just a subset of the MOD format, which not only provides the functionality of position mapping, but also includes both original and new genomic subsequences. In fact, we can derive and store the internal position mapping of a MOD file into the Chain format or the delta format. To sum up, the MOD format we propose does not replace any existing formats; it should be considered as a novel format that brings the convenience of a single file to describe sequence transformations and mappings, thus aiding high-throughput sequence alignment pipelines.

Our MOD format is also applicable to non-inbred, diploid, strains. In an inbred strain, the two diploid genomes are identical, thus requiring only one MOD file to relate its genome sequence to the reference. For an outbred sample, we use two MOD files, one for each haplotype sequence. From these MOD files we construct two genomes and perform separate alignments to each one, and remap both alignments back to reference coordinates using Lapels as shown in Fig. 3 . We are then able to ascertain the originating sequence for reads where possible as determined by informative differences between the diploid sequences. We have demonstrated this capability on F1 hybrids between two inbred strains [11].

Furthermore, we can employ the MOD files in DNA-seq alignment pipelines to increase mapping quality. To do that, we first use the traditional alignment procedure to map the DNA-seq reads against the reference and called variants based on the read alignments. Then we convert the variant calls into the MOD format, and construct a pseudogenome, to which we re-align the original set of DNA-seq reads iteratively. Subsequent alignments will go on to contribute new variant calls, as the procedure repeats. This gradually results in a more accurate pseudogenome.

5. ACKNOWLEDGMENT

We thank the members in the UNC CEGS group for strong support and helpful advice to this work. We also acknowledge the variant calls and the DNA-seq read alignments from the Wellcome Trust Sanger Institute. Finally, we would like to thank the three anonymous reviewers for their insightful questions and comments.

This work is supported by NIH P50 MH090338 and NSF IIS-0812464, IIS-1313606.

6. REFERENCES

[1] Ucsc liftover. `http://genome.ucsc.edu/cgi-bin/hgLiftOver`. Accessed: 2013-07-26.

[2] P. Danecek, A. Auton, G. Abecasis, C. Albers, E. Banks, M. DePristo, R. Handsaker, G. Lunter, G. Marth, S. Sherry, et al. The variant call format and vcftools. *Bioinformatics*, 27(15):2156–2158, 2011.

[3] J. Degner, J. Marioni, A. Pai, J. Pickrell, E. Nkadori, Y. Gilad, and J. Pritchard. Effect of read-mapping biases on detecting allele-specific expression from rna-sequencing data. *Bioinformatics*, 25(24):3207–3212, 2009.

[4] A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg. Alignment of whole genomes. *Nucleic Acids Research*, 27(11):2369–2376, 1999.

[5] B. DeVeale, D. van der Kooy, and T. Babak. Critical evaluation of imprinted gene expression by rna–seq: A new perspective. *PLoS Genetics*, 8(3):e1002600, 2012.

[6] P. Flicek, M. R. Amode, D. Barrell, K. Beal, S. Brent, D. Carvalho-Silva, P. Clapham, G. Coates, S. Fairley, S. Fitzgerald, et al. Ensembl 2012. *Nucleic Acids Research*, 40(D1):D84–D90, 2012.

Table 7: Percentage of DNA-seq reads from CAST samples mapped to the pseudogenome and the reference.

Alignment		Reference		
		Mapped	Unmapped	Total
CAST Pseudogenome	Mapped	95.74%	0.74%	96.48%
	Unmapped	0.04%	3.47%	3.51%
	Total	95.78%	4.21%	100.00%

Table 8: Breakdown percentage of reads that mapped in both genomes. NM_1 and NM_2 are the edit distances from read alignments to the CAST pseudogenome and the reference, respectively.

	$NM_1 < NM_2$	$NM_1 = NM_2 = 0$	$NM_1 = NM_2 > 0$	$NM_1 > NM_2$	Total
Same Position	36.19%	30.70%	24.01%	0.10%	91.00%
Diff. Positions on the Same Chrom.	1.78%	0.28%	0.59%	0.17%	2.81%
Diff. Chrom.	0.41%	0.56%	0.62%	0.35%	1.94%

[7] C.-P. Fu, C. E. Welsh, F. P.-M. de Villena, and L. McMillan. Inferring ancestry in admixed populations using microarray probe intensities. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, BCB '12, pages 105–112, New York, NY, USA, 2012. ACM.

[8] X. Gan, O. Stegle, J. Behr, J. G. Steffen, P. Drewe, K. L. Hildebrand, R. Lyngsoe, S. J. Schultheiss, E. J. Osborne, V. T. Sreedharan, et al. Multiple reference genomes and transcriptomes for arabidopsis thaliana. *Nature*, 477(7365):419–423, 2011.

[9] C. Gregg, J. Zhang, B. Weissbourd, S. Luo, G. P. Schroth, D. Haig, and C. Dulac. High-resolution analysis of parent-of-origin allelic expression in the mouse brain. *Science*, 329(5992):643–648, 2010.

[10] D. Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge University Press, 1997.

[11] J. Holt, S. Huang, L. McMillan, and W. Wang. Read annotation pipeline for high-throughput sequencing data. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*. ACM, 2013.

[12] T. Keane, L. Goodstadt, P. Danecek, M. White, K. Wong, B. Yalcin, A. Heger, A. Agam, G. Slater, M. Goodson, et al. Mouse genomic variation and its effect on phenotypes and gene regulation. *Nature*, 477(7364):289–294, 2011.

[13] B. Langmead and S. L. Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357–359, 2012.

[14] B. Langmead, C. Trapnell, M. Pop, S. L. Salzberg, et al. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biol*, 10(3):R25, 2009.

[15] H. Li. Tabix: fast retrieval of sequence features from generic tab-delimited files. *Bioinformatics*, 27(5):718–719, 2011.

[16] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, et al. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[17] E. Liu, Q. Zhang, L. McMillan, F. de Villena, and W. Wang. Efficient genome ancestry inference in complex pedigrees with inbreeding. *Bioinformatics*, 26(12):i199–i207, 2010.

[18] J. Marioni, C. Mason, S. Mane, M. Stephens, and Y. Gilad. Rna-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research*, 18(9):1509–1517, 2008.

[19] R. McDaniell, B.-K. Lee, L. Song, Z. Liu, A. P. Boyle, M. R. Erdos, L. J. Scott, M. A. Morken, K. S. Kucera, A. Battenhouse, et al. Heritable individual-specific and allele-specific chromatin signatures in humans. *Science*, 328(5975):235–239, 2010.

[20] Z. Peng, Y. Cheng, B. Tan, L. Kang, Z. Tian, Y. Zhu, W. Zhang, Y. Liang, X. Hu, X. Tan, et al. Comprehensive analysis of rna-seq data reveals extensive rna editing in a human transcriptome. *Nature biotechnology*, 30(3):253–260, 2012.

[21] E. Picardi, D. Horner, M. Chiara, R. Schiavon, G. Valle, and G. Pesole. Large-scale detection and analysis of rna editing in grape mtdna by rna deep-sequencing. *Nucleic acids research*, 38(14):4755–4767, 2010.

[22] M. Rivas-Astroza, D. Xie, X. Cao, and S. Zhong. Mapping personal functional data to personal genomes. *Bioinformatics*, 27(24):3427–3429, 2011.

[23] J. Rozowsky, A. Abyzov, J. Wang, P. Alves, D. Raha, A. Harmanci, J. Leng, R. Bjornson, Y. Kong, N. Kitabayashi, et al. Alleleseq: analysis of allele-specific expression and binding in a network framework. *Molecular systems biology*, 7(1), 2011.

[24] R. Satya, N. Zavaljevski, and J. Reifman. A new strategy to reduce allelic bias in rna-seq readmapping. *Nucleic Acids Research*, 40(16):e127–e127, 2012.

[25] L. Silver et al. *Mouse genetics: concepts and applications*. Oxford University Press, 1995.

[26] The 1000 Genomes Project Consortium. A map of human genome variation from population scale sequencing. *Nature*, 467(7319):1061–1073, 2010.

[27] The ENCODE Project Consortium et al. A user's guide to the encyclopedia of dna elements (encode). *PLoS Biol*, 9(4):e1001046, 2011.

[28] C. Trapnell, L. Pachter, and S. Salzberg. Tophat:

discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111, 2009.

[29] C. Trapnell, B. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. Van Baren, S. Salzberg, B. Wold, and L. Pachter. Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5):511–515, 2010.

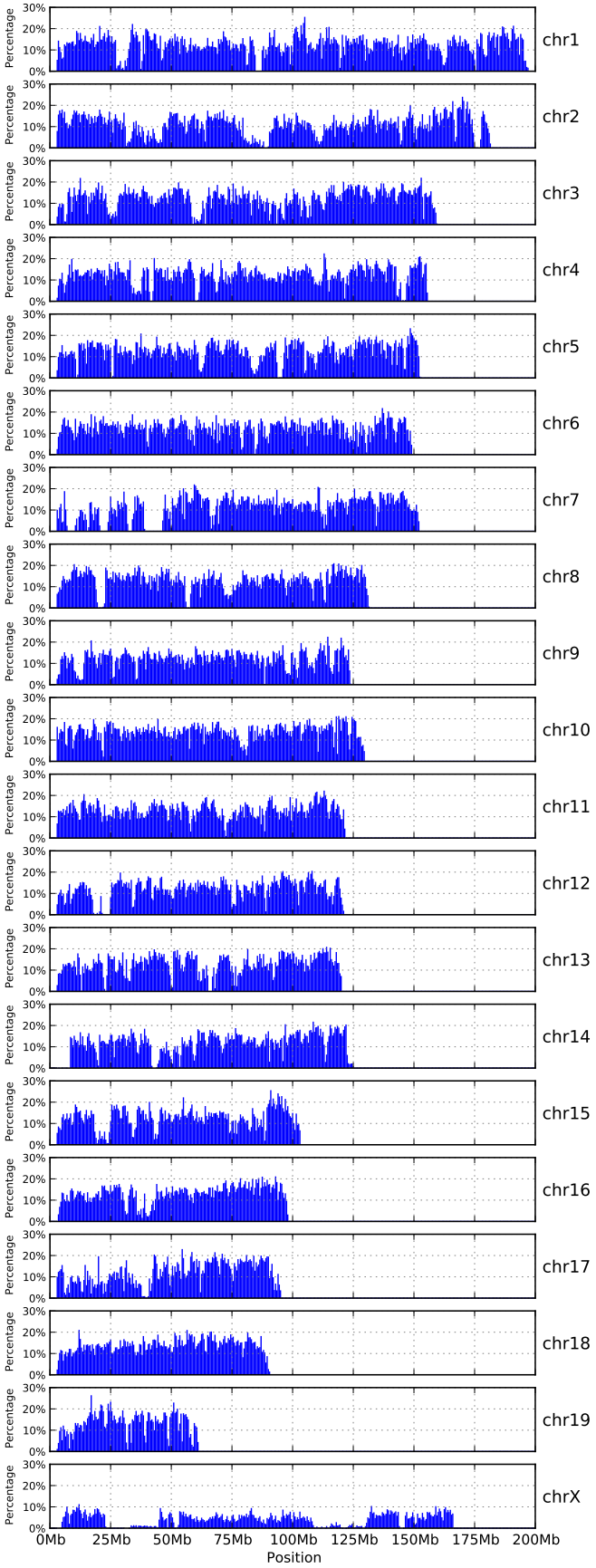

Figure 5: The bar chart of high-variant intervals of CAST pseudogenome. Each bar represents the percentage of 100bp windows within a 500Kb region that contain 3 or more sequence variations relative to the reference strain. Such regions present problems for most sequence aligners.