Chromosome-length haplotypes with StrandPhaseR and Strand-seq

Vincent C. T. Hanlon[1,*], David Porubsky[2,*] and Peter M. Lansdorp[1,3]

[1]Terry Fox Laboratory, BC Cancer Agency, Vancouver, BC V5Z 1L3, Canada

[2]Department of Genome Sciences, University of Washington School of Medicine, Seattle, WA, USA

[3]Department of Medical Genetics, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada

*equal contributions

Correspondence: vhanlon@bccrc.ca or porubsky@uw.edu

Running head: Phasing SNVs with Strand-seq

Abstract

Dense local haplotypes can now readily be extracted from long-read or droplet-based sequence data. However, these methods struggle to combine subchromosomal haplotype blocks into global chromosome-length haplotypes. Strand-seq is a single cell sequencing technique that uses read orientation to capture sparse global phase information by sequencing only one of two DNA strands for each parental homolog. In combination with dense local haplotypes from other technologies, Strand-seq data can be used to obtain complete chromosome-length phase information. In this chapter, we run the R package StrandPhaseR to phase SNVs using publicly-available sequence data for sample HG005 of the Genome in a Bottle project.

Key Words

phasing, haplotype, Strand-seq, StrandPhaseR, Genome in a Bottle

1. Introduction

Read orientation encodes phase information in single cell Strand-seq data. Strand-seq libraries capture only the template strand of every homologous chromosome (for details of the library preparation protocol, see ***(1, 2)***). Since DNA strands are directional, all reads from a single parental homolog map to the reference genome in the same orientation. Thus, when a given chromosome in a diploid cell inherits the Watson template strand from one parent (all reads map in the minus orientation) and the Crick template strand from the other parent (all reads map in the plus orientation), we can readily distinguish SNVs covered by reads mapped in the minus and plus orientations, respectively. This allows us to unambiguously group parental alleles captured by reads mapped in the same orientation into two chromosome-length haplotypes ***(3)*** (Figure 1).

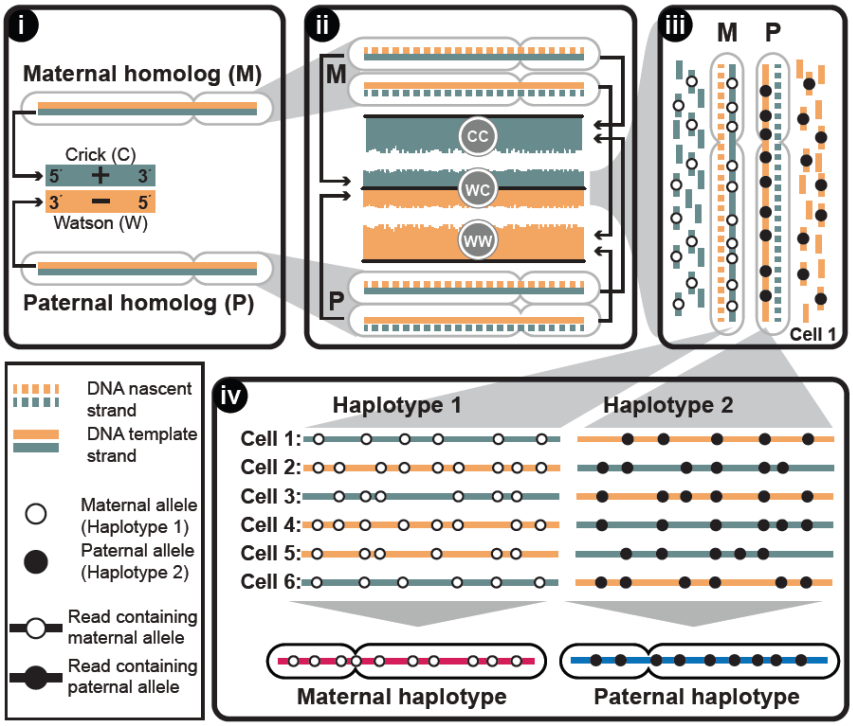

Figure 1: Strand-seq based phasing. **i**) In diploid organisms there are maternal and paternal homologs for each chromosome. Each homolog is composed of one positive template strand (Crick; teal) and one negative template strand (Watson; orange). **ii**) To generate Strand-seq libraries, single cells replicate their DNA in the presence of bromodeoxyuridine (BrdU), which is a thymidine analogue, for one round of replication. A cell incorporates BrdU into the newly synthesized (nascent) DNA strands. This results in sister chromatids that contain one original template strand (solid lines) and one nascent DNA strand (dashed lines) with BrdU incorporated. Subsequent cell division distributes paternal and maternal sister chromatids to daughter cells, with three possible combinations of template strands: Option 1 (WC), option 2 (CC) and option 3 (WW). Note that newly formed DNA strands containing BrdU are selectively removed in daughter cells during library preparation, such that only the original template DNA strands are sequenced ***(1, 2)***. **iii**) For haplotype phasing only option 1 (WC) is informative as template strands of different directionality (Watson and Crick) are inherited from both parental homologs and thus can be distinguished upon mapping to the reference genome. This way reads that map in the forward or reverse orientation to the reference genome contain homolog-specific alleles inherited from a single parental homolog. **iv**) StrandPhaseR is an R package which is able to organize single cell haplotypes from WC regions of multiple cells in such a way that alleles contained in Watson or Crick reads match only a single haplotype for a given homologous chromosome. The overall density of alleles is increased by the cumulative coverage of each single cell Strand-seq library. Note that additional data is typically required to assign chromosomal haplotypes to individual parents.

In each Strand-seq library, such phase information is restricted to chromosomes with reads in both orientations. That is, expecting diploid chromosomes, one parental homolog must inherit a Watson template strand and the other must inherit a Crick template strand (called "WC" chromosomes). We cannot extract phase information from chromosomes that inherit two Watson or two Crick template strands from both parents (called "WW" or "CC"), because SNVs are only covered by reads mapped in the same orientation ***(3)*** (Figure 1). Assuming the random segregation of sister chromatids, there is a 50% chance for each chromosome to inherit one Watson and one Crick template from different parental homologs. This means that in a single Strand-seq library about half of the chromosomes are haplotype informative, and in theory at least 5 libraries are needed to phase the full length of the genome ***(4)***, although more are generally used. In addition, the occurrence of double-strand breaks during DNA replication, which are repaired using sister chromatids and cause template strand switches termed Sister Chromatid Exchanges (SCEs), complicates this picture ***(5)***. SCEs alter the orientation of reads within a parental chromosome, so that the template strand state of genomic regions within chromosomes (WC, WW, or CC) must be checked to determine whether they are suitable for phasing (WC regions; Figure 1). In practice it is straightforward to identify the appropriate (haplotype informative) genomic regions in each library using the R package BreakpointR ***(6)***.

In a typical single cell Strand-seq library, less than 20% of heterozygous SNVs are covered by even a single read ***(7)***. The sparse breadth of coverage of the libraries—generally between 0.01x and 0.2x the haploid genome—means that phase information from many libraries must be combined (Figure 1). This can be done with StrandPhaseR ***(8)***. The central challenge is that when comparing a WC region in two or more libraries, it is unknown whether reads with the same orientation originate from the same homolog. That is, a homologous chromosome may have inherited a Watson template strand in one cell and a Crick template strand in another, giving rise to reads with opposite orientations that cover SNVs on the same haplotype (Figure 2). StrandPhaseR flips single cell haplotypes one by

one and tests whether this improves the concordance of single cell haplotypes at the SNVs they have in common. This allows it to produce a consensus haplotype iteratively using SNVs that are covered by reads in multiple libraries.

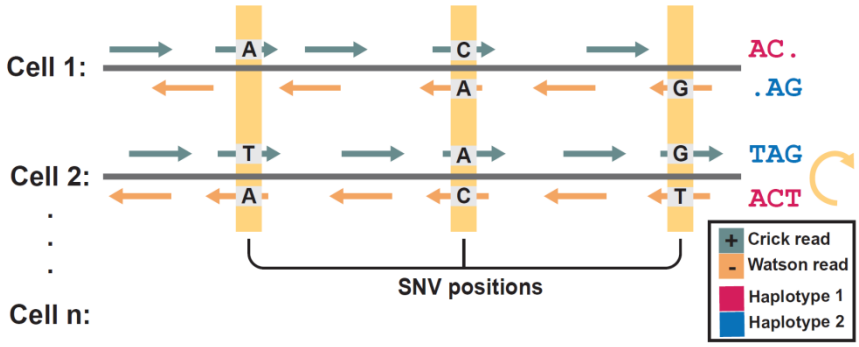

Figure 2. Combining single cell haplotypes by comparing overlapping WC regions. Reads containing the same allele in cells 1 and 2 map in opposite orientations: for example, at the leftmost SNV, the A allele is in a Crick/plus read in cell 1 but a Watson/minus read in cell 2. This means haplotype information from cell 1's Crick reads should be combined with haplotype information from cell 2's Watson reads and vice versa (i.e., flipping the haplotypes before combining).

Most users of StrandPhaseR will already have selected Strand-seq data for their analysis. For those who have not, there are now a number of publicly-available Strand-seq datasets, often from publicly-available human cell lines. The Genome in a Bottle Consortium (GIAB) hosts Strand-seq libraries for two trios (HG002-HG004 and HG005-HG007; ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/; ***(9)***). There exist multiple datasets for HG001/NA12878 ***(3, 7)***, as well as for 33 other samples related to the 1000 Genomes project ***(10)***, for a total of 8 BioProjects with human Strand-seq libraries (Table 1).

Droplet-based Chromium sequencing or improved phasing algorithms for NanoPore data ***(11, 12)*** can now span chromosomes with just a handful of large haplotype blocks, although they cannot yet produce chromosome-length haplotypes. At present, Strand-seq's global phase information is rivalled only by trio-based phasing and by Hi-C, which exploits the spatial proximity of pairs of loci in chromatin ***(13)***. Unlike trio phasing, however, Strand-seq does not require parental information; and unlike Hi-C, the accuracy of Strand-seq phasing

does not decay with increasing genomic distance (Figure 1) and guarantees each chromosome is phased from telomere to telomere. In Strand-seq all alleles represented in reads specific to either Watson or Crick parental templates are linked physically irrespective of their genomic distance. For the foreseeable future, the winning strategy will be to combine technologies that offer global phase information with technologies that produce dense intermediate- or local-scale haplotypes.

Table 1. BioProject accession numbers for human samples with Strand-seq data on the NCBI Sequence Read Archive.

BioProject	**Donor details**
PRJNA742746	Coriell NA12878
PRJEB39750	34 donors from the 1000 Genomes project
PRJEB33731	3 RPE-1-derived cell lines
PRJEB24615	4 healthy donors, 4 with Bloom syndrome
PRJEB14185	Coriell NA12878, NA12891, NA12892
PRJEB13795	Coriell NA16375, NA12891, NA03402, NA07492
PRJEB12849	3 trios from the 1000 Genomes project
PRJNA273996	47 cells, each likely from a unique donor

2. Materials

The full method described in this chapter requires a Linux working environment and an internet connection, as well as at least four cores (more is better) and 130 GB free space on the hard drive. However, the core of the method (steps 3 and 5) requires only the R packages BreakpointR and StrandPhaseR and their dependencies, as well as a set of good-quality Strand-seq BAM files and SNVs, and can be run on other operating systems.

To begin, acquire some Strand-seq data and install the required software:

1. Create a working directory and download the example data. We will use the Strand-seq libraries for HG005/NA24631/GM24631 hosted by GIAB on their FTP server. You may also download GIAB's SNV calls for HG005 (optional). Alternatively, you can use Strand-seq libraries of your choice, which may be supplemented with a matching SNV callset. Open the Linux command line and run the following commands:

```
mkdir SPR
cd SPR
wget  ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/ChineseTrio/HG005_NA24631_son/Strand-Seq_BCCRC/*
wget ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/ChineseTrio/HG005_NA24631_son/NISTv4.2.1/GRCh38/HG005_GRCh38_1_22_v4.2.1_benchmark.vcf.gz
```

2. Confirm that the 90 FASTQ files downloaded correctly. The GIAB data come with MD5 hash values for each file. When you generate new MD5 hash values for the files

once they have downloaded, they should match the original hash values. Skip this step if you are not using the HG005 dataset.

```
for i in *.fastq.gz; do md5sum $i; done > new.md5sum.txt
diff new.md5sum.txt gm24631.md5sum.txt
```

If this command produces no output, the files are complete and you should continue to step 3.

3. If you do not already have the software packages below, install them. It may be easiest to use a conda environment (Note 1).
    a. R 4.1.0
    b. Samtools ≥1.12 ***(14)***
    c. bwa-mem2 (or an aligner of your choice; ***(15)***)
    d. Bbtools ***(16)***
    e. GNU parallel
    f. Miniconda3
    g. WhatsHap ***(17)***

4. Install ASHLEYS QC following the instructions on the GitHub page (commit 806cd3e8357db9f570279a6b073b9d300d0494ed; github.com/friendsofstrandseq/ashleys-qc; ***(18)***). This is the first version of the software, so future updates may change the installation instructions. However, for the current (January 2022) version, run:

```
git clone https://github.com/friendsofstrandseq/ashleys-qc.git ashleys-qc
cd ashleys-qc
conda env create -f environment/ashleys_env.yml
conda activate ashleys
python setup.py install
conda deactivate
cd ..
```

5. Install StrandPhaseR and BreakpointR (Notes 2 and 3). If you used a conda environment in step 3, activate it first (Note 1). Open R by typing "R" on the Linux command line, then write:

```
install.packages(c("devtools","BiocManager"))
BiocManager::install(c("BSgenome", "BSgenome.Hsapiens.UCSC.hg38", "zoo",
"breakpointR"))
devtools::install_github("daewoooo/StrandPhaseR")
```

3. Methods

1. First, we must align reads to the reference genome, select properly paired reads with mapping quality of at least 10, sort reads by position, mark duplicates, and index BAM files (some FASTQ files may require adapter trimming; see Note 4; see Figure 3 for an overview of the whole method). This is standard processing for short-read Illumina sequence data, so you may wish to apply your routine alignment pipeline and skip ahead to step 2. Otherwise, you can save the commands in step 1b to a file, say "alignment.sh", and run it as shown below (this script and other code in this chapter can be found on GitHub at github.com/vincent-hanlon/MiMB-StrandPhaseR). If you installed the software in the Materials section as a conda environment, you will need to activate it first.

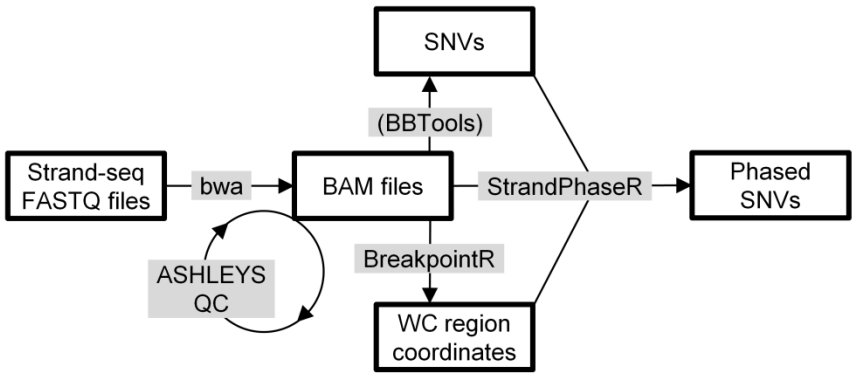

Figure 3. An overview of the method described in this chapter (Note 5). Grey boxes indicate the software used. BBTools is shown in brackets because SNV calling can be skipped if SNVs are provided independently.

a. If the GRCh38/hg38 human reference genome you are using is not already bwa-indexed, run the following:

```
bwa index -a bwtsw /path/to/reference/fasta/GRCh38
```

b. Run the following as a script. In the command, substitute the path to the reference genome and the number of CPUs you wish to use (numCPU).

command:

```
bash alignment.sh numCPU "/path/to/reference/fasta/GRCh38.fasta"
```

alignment.sh:

```
# 1st argument ($1) is the name of the library to process.
# 2nd argument ($2) is the path to the reference genome.

# Run like this:
# bash alignment.sh numCPU /path/to/reference/genome.fasta.

# At the moment, this is set up for PE reads, but it can easily be
```

```
altered for SE reads.
# Similarly, this script expects FASTQ files ending in "_R1_001.fastq.gz"
and "_R2_001.fastq.gz".

for i in *R1_001.fastq.gz
do
       name=$(echo $i | sed 's/_R1_001\.fastq\.gz//')
       bwa mem -t "$1" "$2" "$i" "$name"_R2_001.fastq.gz  > "$name".sam
done

# compresses, sorts, marks duplicates, and indexes files with aligned
sequence reads.
process() {

        name=$(echo $1 | sed 's/\.sam//')

        samtools view -q10 -f2 -bS "$name".sam | samtools sort -n -o
"$name".bam
        samtools fixmate -m "$name".bam - | samtools sort -o - - |
samtools markdup - "$name".sorted.mdup.bam
        samtools index "$name".sorted.mdup.bam

}

export -f process
```

```
parallel --jobs "$1" process ::: *.sam
```

c. It is helpful to organise the files used or created during the alignment process. Delete the SAM files to save space if needed.

```
mkdir intermediate fastq

mv *.fastq.gz fastq

mv *.sam *_???.bam intermediate
```

2. Next, select good-quality Strand-seq libraries using the BAM files created in step 1. While experienced users may select good-quality Strand-seq libraries by examining BreakpointR ideograms, the easiest and most reproducible method is ASHLEYS QC ***(18)***.

a. Activate the appropriate conda environment

```
conda activate ashleys
```

b. Run ASHLEYS. Substitute the number of CPUs you wish to use. Replace "/path/to/ashleys-qc/" with the location of your installation files for ASHLEYS.

```
ashleys.py -j numCPU features -f ./ -w 5000000 2000000 1000000 800000
600000 400000 200000 -o ./features.tsv


ashleys.py predict -p ./features.tsv -o ./quality.txt -m
/path/to/ashleys-qc/models/svc_default.pkl
```

c. Retain only libraries that have good quality (Note 6). For example,

```
mkdir bam_poor_quality
for i in $(awk '$3<=0.5 {print $1}' quality.txt)
do
     mv $i $i.bai bam_poor_quality
done
```

3. Now run BreakpointR to identify WC regions in libraries, that is, chromosomal regions for which reads from the two homologs map in opposite orientations to the reference genome. Step 4 can be completed simultaneously with step 3 to save time.
    a. Deactivate the ASHLEYS conda environment, and if needed, activate the environment into which you installed BreakpointR.

```
conda deactivate
```

    b. Write "R" to open a session of R and run BreakpointR. Change the number of CPUs to use (numCPU) as desired. Write ?breakpointr for more options.

```
library(breakpointR)
breakpointr(inputfolder = "./", outputfolder = "./BPR_output/",
windowsize = 2000000, binMethod = 'size', pairedEndReads = TRUE,
pair2frgm = FALSE, min.mapq = 10, filtAlt = TRUE, background = 0.1,
minReads = 50, numCPU=4)
```

    c. Now, extract the WC regions as a tab-separated file. Note that exported WC regions may contain structural variants (SVs), within which StrandPhaseR is unlikely to phase SNVs correctly (Note 7). Write ?exportRegions for more options.

```
exportRegions(datapath = "./BPR_output/data", file = "wc_regions.txt",
collapseInversions = FALSE, minRegionSize = 5000000, state = 'wc')
```

4. We need a list of SNVs to phase (Note 8). If you already have a VCF file containing good-quality SNVs (e.g., from GIAB for HG005; see step 1 of the Materials section), or if you are able to call SNVs using Illumina whole genome sequencing (WGS) or long read data, skip to step 5. Otherwise, while Strand-seq data typically has low coverage, for many applications it is sufficient to call SNVs by combining all Strand-seq libraries for a given sample. Here, we apply BBTools ***(16)***. We use both good- and poor-quality libraries to maximise combined coverage. Substitute the path to your copy of the reference genome.

```
ls ./*.bam bam_poor_quality/*.bam > samples.list


callvariants.sh list=samples.list
ref=/path/to/reference/fasta/GRCh38.fasta out=strand_seq_snps.vcf
ploidy=2 calldel=f callins=f
```

5. To phase the SNVs, run StrandPhaseR. Open R and run the command below. If one wishes to investigate large SVs such as heterozygous deletions (typically >10 kb) on the UCSC Genome Browser ***(19)***, we recommend setting splitPhasedReads=TRUE. This will create BED files with Strand-seq reads assigned to either haplotype 1 or haplotype 2 (step 7c; but see Note 7). Change the number of CPUs (numCPU) as desired, as well as the name of the VCF file containing SNV positions. Write ?strandPhaseR for more options.

```
library(StrandPhaseR)
library(BSgenome.Hsapiens.UCSC.hg38)
```

```
library(parallel)


chromosomes <-
c('chr1','chr2','chr3','chr4','chr5','chr6','chr7','chr8','chr9','chr10'
,'chr11','chr12','chr13','chr14','chr15','chr16','chr17','chr18','chr19'
,'chr20','chr21','chr22')


strandPhaseR(inputfolder = "./", outputfolder="./SPR_output", positions
= "strand_seq_snps.vcf",
WCregions = "wc_regions.txt", pairedEndReads = TRUE, num.iterations = 3,
numCPU=4, exportVCF="HG005", bsGenome='BSgenome.Hsapiens.UCSC.hg38',
chromosomes=chromosomes)
```

6. StrandPhaseR produces 5 subfolders and a configuration file as output.
    a. The directory VCFfiles contains a VCF file of phased SNVs for each chromosome. In the standard notation, 0|1 and 1|0 represent phased heterozygous genotypes (on different haplotypes). Incomplete genotypes are written 0|., 1|., .|0, and .|1. StrandPhaseR reports only putative bialellic heterozygous SNVs, so 0|. is effectively the same as 0|1, 1|. is the same as 1|0, etc. (Note 9). It may be desirable to combine the VCF files for all chromosomes into a single phased VCF file:

```
bcftools concat --threads 4 SPR_output/VCFfiles/chr*.vcf >
./strandseq_phased_SNVs.vcf
```

    b. The directory Phased contains two types of files. Files named like chr14_phased_hap1.txt give information about the quality of individual alleles at SNVs on each haplotype: the number of Strand-seq reads used to phase

each SNV ("cov"), a measure of the similarity of the single cell haplotype to the consensus haplotype ("score"), and the entropy ("ent"). Files named like chr14_phasedFiles_hap2.txt determine whether plus-oriented reads belong to haplotype 1 or haplotype 2 for the WC regions provided to StrandPhaseR (and conversely, whether minus reads belong to haplotype 2 or haplotype 1).

c. The directory data contains .RData files, one for each chromosome, which store the SNV matrices that are sorted to optimize the assignment of SNVs to haplotypes as part of a maximum likelihood approach ***(8)***.

d. When splitPhasedReads=TRUE, the directory browserfiles contains haplotype-specific BED files that can be loaded into the UCSC Genome Browser for viewing.

e. When compareSingleCells=TRUE, the directory SingleCellHaps contains single cell loss of heterozygosity (LOH) analyses. In our experience, single cell LOH is uncommon, and this functionality is rarely used.

f. The configuration file StrandPhaseR.config can be used to reproduce a StrandPhaseR run as follows:

```
strandPhaseR(inputfolder = "./", outputfolder="./SPR_output", positions
= "strand_seq_snps.vcf",
WCregions = "wc_regions.txt", configfile = "StrandPhaseR.config")
```

7. Phased SNVs may be valuable or interesting in themselves, but a number of further analyses are also possible. We briefly illustrate three of them:

    a. To compensate for the shallow coverage of most Strand-seq datasets and phase missing SNVs, you can apply WhatsHap to fill in the global Strand-seq haplotypes with dense local haplotypes using short- or long-read data. To phase indels add the flag –indels.

```
whatshap phase -o all_phased.vcf --reference=/path/to/reference.fasta
all_unphased_SNVs.vcf additional_WGS.bam strandseq_phased_SNVs.vcf
```

b. It is also possible to tag short and long sequence reads by haplotype. The command below tags BAM reads with HP:i:1 or HP:i:2 according to their haplotype, and these tags can then be used to separate reads into haplotype-specific BAM files using samtools (version 1.12 or greater). However, only reads that cover a phased heterozygous SNV can be tagged in this way. Another limitation is that reads fully contained within SVs may be tagged incorrectly or not at all (see Note 7). For example, a heterozygous deletion may be mistaken for a homozygous deletion in haplotype-separated data, because reads inside a heterozygous deletion will not be tagged for lack of phased heterozygous SNVs (assuming the SNVs were called using samples that contained the deletion).

```
bgzip strandseq_phased_SNVs.vcf
tabix -p vcf strandseq_phased_SNVs.vcf.gz
whatshap haplotag -o haplotagged.bam --reference
/path/to/reference.fasta strandseq_phased_SNVs.vcf.gz data_to_split.bam
--ignore-read-groups --skip-missing-contigs
samtools view -bh -d HP:1 haplotagged.bam > haplotagged.hap1_only.bam
```

c. The UCSC Genome Browser BED files that StrandPhaseR produces with the option splitPhasedReads=TRUE, which contain reads from phased WC regions separated into two haplotypes based on read orientation, can be used to view known heterozygous deletions. The deletions appear as read-poor regions on one haplotype but not the other (Figure 4). Not all read-poor regions are deletions, however, because heterozygous inversions can mimic

heterozygous deletions in the BED files when the SNVs they contain are all mistakenly assigned to one haplotype (Note 7). In some cases, paired-normal type analyses can help detect and resolve SVs in reads separated by haplotype. For example, when SNVs can be phased using Strand-seq for a normal (wildtype) sample from an individual, complex SVs in a paired tumour sample can be more reliably phased because the SVs themselves do not interfere with StrandPhaseR's SNV phasing step.

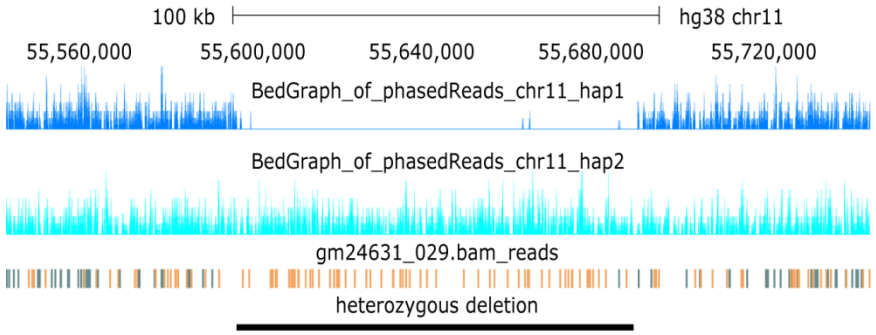

Figure 4. A 93 kb heterozygous deletion in HG005 viewed on the UCSC Genome Browser. The two haplotype-separated StrandPhaseR BED files (blue) with reads from all libraries show that only one haplotype lacks reads within the deletion. The single-library BreakpointR BED file, in which orange and teal reads map with different orientations, shows the deletion as a stretch of unidirectional reads within a WC region

4. Notes

1. Rather than installing the correct versions of R, samtools, etc., individually, it may be simpler to use the package management system Miniconda3 (docs.conda.io/en/latest/miniconda.html). It is required for the installation of ASHLEYS, but you can also use it to install the other required programs more easily, for example

```
conda create -c bioconda -c conda-forge -c agbiome -n spr r-base=4.1.0
samtools=1.14 bwa-mem2 bbtools=37* parallel whatshap
```

The conda environment containing these packages can then be activated using

```
conda activate spr
```

2. If difficulties arise installing any of the R packages or their dependencies, check whether they can be installed with conda more easily. For example, devtools, zoo, and even BreakpointR are available from conda, and they can be installed into the environment described in Note 1:

```
conda install -c conda-forge r-devtools
conda install -c r r-zoo
conda install -c bioconda bioconductor-genomicranges
```

If the installation fails because of dependency conflicts, try removing the entire conda environment and then re-creating it with the R package you need to add. For example:

```
conda remove -n spr --all
conda create -c bioconda -c conda-forge -c agbiome -c r -n spr r-
base=4.1.0 samtools=1.14 bwa-mem2 bbtools=37* parallel whatshap r-zoo
```

3. We installed the most recent commit for StrandPhaseR on GitHub (as of January 2022; github.com/daewoooo/StrandPhaseR). If you wish to use exactly the same version, substitute this command:

```
devtools::install_github("daewoooo/StrandPhaseR",
ref="3c8d905bf0269b93f4da1e1ca7934887a32db024")
```

4. FASTQ files other than the HG005 libraries used in this chapter may contain adapter sequences that must be removed. To check whether this is necessary, run FastQC (www.bioinformatics.babraham.ac.uk/projects/fastqc/) and check the "Adapter

Content" and "Overrepresented Sequences" fields in the HTML files produced. If adapter-like sequences appear there, substitute them into the following Cutadapt command ***(20)*** and run it for each pair of FASTQ files. The flag "-b" denotes the read 1 adapter and the flag "-B" denotes the read 2 adapter, and they can be repeated if multiple adapter sequences are present.

```
cutadapt \
        -b
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACNNNNNNNNATCTCGTATGCCGTCTTCTGCTTG \
        -B
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTNNNNNNGTGTAGATCTCGGTGGTCGCCGTATCATT \
        -o output_file.trimmed_R1_001.fastq.gz" \
        -p output_file.trimmed_R2_001.fastq.gz" \
        input_file.R1_001.fastq.gz" \
        input_file.R2_001.fastq.gz" \
        -m 30 \
        -q 15
```

Cutadapt and FastQC can be installed with Conda:

```
conda install -c bioconda cutadapt
conda install -c bioconda fastqc
```

5. All the steps of this analysis pipeline can be combined using the snakemake pipeline manager. For educational purposes we present all the analysis steps separately to better understand each individual step and modify and adjust the analysis to one's personal preference and needs.

6. ASHLEYS assigns libraries a quality score between 0 and 1, where higher values represent better libraries ***(18)***. The creators recommend that QC should be verified manually for libraries with scores between 0.3 and 0.7, for example, by examining BreakpointR ideograms. Although users of StrandPhaseR would ideally follow this recommendation, it may be difficult for those who are not accustomed to evaluating the quality of Strand-seq libraries. Instead, it may be more practical to omit manual verification and use all libraries that surpass some quality cutoff.

   We compared phasing using libraries with ASHLEYS quality scores above 0.5 or 0.7 for the HG005 libraries. With the 0.5 cutoff, StrandPhaseR was able to phase 25% more SNVs than with the 0.7 cutoff, with similar switch and switch/flip error rates relative to the trio-phased GIAB SNVs (switch error 1.38% vs.1.39% with the 0.7 cutoff; switch/flip error 0.97% for both cutoffs; calculated with WhatsHap across all intersection blocks and averaged over all chromosomes). However, the block-wise Hamming distance was slightly lower with the 0.7 cutoff (1.22% vs. 1.62%; averaged over all chromosomes).

   Based on these results, we believe that an ASHLEYS quality score cutoff of 0.5 is sufficient for many applications, although we encourage users who are familiar with manually verifying Strand-seq library quality to do so. We note that ASHLEYS QC tends to give high quality scores to rare WGS-like libraries in which Strand-seq failed (data not shown). In such libraries, all chromosomes (rather than roughly half) appear to be WC regions with reads in both orientations, and the orientation of reads is random with respect to the phase of the alleles they cover. Such libraries occasionally score above 0.7, but they score above 0.5 more frequently and may sometimes be used for phasing when the 0.5 cutoff is applied. However, they can easily be excluded manually by examining BreakpointR ideograms and checking for libraries in which all diploid chromosomes have reads in both orientations (i.e., both

orange and teal) along their full length.

7. SVs such as inversions disrupt the relationship between read orientation and haplotype that StrandPhaseR exploits, while copy number variants break the assumption that SNV alleles can be partitioned into exactly two haplotypes. For example, homozygous inversions within WC regions will result in a complete switch in haplotype, while heterozygous inversions will cause nearly all alleles to be jumbled together on the non-inverted haplotype. Similarly, a duplication can yield a WC region in which reads with one orientation are roughly 2x more common than expected and the SNV haplotype of the duplication is mixed with one of the two original SNV haplotypes. In general, it is best to distrust StrandPhaseR's SNV phasing inside SVs. However, SNVs inside inversions can be phased locally when WC regions are defined that capture only the inverted regions, that is, when all region start and end coordinates match inversion breakpoints. The local haplotypes within the inversions can in theory be merged with global chromosome-length haplotypes for non-inverted regions by accounting for the (phased) genotype of each inversion.

8. It is important to be sure that few of the heterozygous genotypes in your VCF file are errors. For example, when an SNV is covered by only one Strand-seq read (from one homolog), a reference genotype mis-called as a heterozygote may cause the user to assign the spurious alternate allele to the other homolog (Note 9). The VCF files output by StrandPhaseR can be used in two ways to check the quality of SNVs. First, StrandPhaseR calculates a quality score for each phased allele (Q1, Q2) where higher values indicate more reliable SNVs. Second, since StrandPhaseR only operates on SNVs with heterozygous genotypes in the input VCF, you can check how often putatively heterozygous SNVs are called homozygous by StrandPhaseR. This is a very inexact measure of SNV quality, but in general we expect that at most 7%—and usually nearer 1%—of SNVs without missing data (without ".") are not 0|1

or 1|0 in StrandPhaseR output.

```
grep -v '^#' ./strandseq_phased_SNVs.vcf | cut -f10 | cut -f1 -d: | grep -v '\.' | sort | uniq -c
```

If BBTools is used to call SNVs from Strand-seq data directly, a variety of variant-calling cutoffs can be altered to improve SNV quality. Write "callvariants.sh" to explore them. In a VCF file, quality metrics such as QUAL (SNV quality), GQ (genotype quality), and DP (coverage) are often given that can help evaluate heterozygous SNVs. If SNVs were called with another data type, such as Illumina WGS, there are a multitude of options to assess or improve SNV quality beyond the scope of this chapter.

9. When you are confident that the heterozygous SNVs input to StrandPhaseR truly are heterozygous, it is reasonable to fill in the missing allele in genotypes like 0|., .|1, etc., as follows:

```
sed -e  's/\.|0/1|0/;s/1|\./1|0/;s/\.|1/0|1/;s/0|\./0|1/' ./strandseq_phased_SNVs.vcf  > ./strandseq_phased_SNVs.sub.vcf
```

However, this will cause problems if there are any instances of .|0, .|1. 1|., or 0|. in the file that should not be replaced (e.g., in the header).

In the current version of StrandPhaseR on the devel branch of the GitHub repository (commit 9e986b445a6339d200301e798b14f2027c530856), this can be done more elegantly by setting assume.biallelic=TRUE.

Acknowledgements

Work in the Lansdorp laboratory is funded by a Program Project Grant (#1074) from the Terry Fox Research Institute, a Project Grant (#PJT-159787) from the Canadian Institutes of Health Research, and a grant (#40044) from the Canadian Foundation for Innovation and the Government of British Columbia.

Conflict of interest

None declared

References

1. Falconer E, Hills M, Naumann U, et al (2012) DNA template strand sequencing of single-cells maps genomic rearrangements at high resolution. Nat Methods 9:1107–1112
2. Sanders AD, Falconer E, Hills M, et al (2017) Single-cell template strand sequencing by Strand-seq enables the characterization of individual homologs. Nat Protoc 12:1151–1176
3. Porubský D, Sanders AD, Wietmarschen N van, et al (2016) Direct chromosome-length haplotyping by single-cell sequencing. Genome Res 26:1565–1574
4. Porubský D (2017), Haplotype resolved genomes: Computational challenges and applications. Dissertation, University of Groningen
5. Wietmarschen N van and Lansdorp PM (2016) Bromodeoxyuridine does not contribute to sister chromatid exchange events in normal or Bloom syndrome cells. Nucleic Acids Res 44:6787–6793
6. Porubsky D, Sanders AD, Taudt A, et al (2020) breakpointR: an R/Bioconductor package to localize strand state changes in Strand-seq data. Bioinformatics 36:1260–1261
7. Hanlon VCT, Chan DD, Hamadeh Z, et al Construction of Strand-seq libraries in open

nanoliter arrays. Cell Rep Meth (in press)

8. Porubsky D, Garg S, Sanders AD, et al (2017) Dense and accurate whole-chromosome haplotyping of individual genomes. Nat Commun 8:1293
9. Wagner J, Olson ND, Harris L, et al (2021), Benchmarking challenging small variants with linked and long reads, https://www.biorxiv.org/content/10.1101/2020.07.24.212712.abstract
10. Ebert P, Audano PA, Zhu Q, et al (2021) Haplotype-resolved diverse human genomes and integrated analysis of structural variation. Science 372:eabf7117
11. Lin J-H, Chen L-C, Yu S-Q, et al (2021), LongPhase: an ultra-fast chromosome-scale phasing algorithm for small and large variants, https://www.biorxiv.org/content/10.1101/2021.09.09.459623v1
12. Weisenfeld NI, Kumar V, Shah P, et al (2017) Direct determination of diploid genome sequences. Genome Res 27:757–767
13. Selvaraj S, R Dixon J, Bansal V, et al (2013) Whole-genome haplotype reconstruction using proximity-ligation and shotgun sequencing. Nat Biotechnol 31:1111–1118
14. Li H, Handsaker B, Wysoker A, et al (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25:2078–2079
15. Li H (2013), Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM, http://arxiv.org/abs/1303.3997
16. Bushnell B BBTools software package, sourceforge.net/projects/bbmap/
17. Martin M, Patterson M, Garg S, et al (2016), WhatsHap: fast and accurate read-based phasing, https://www.biorxiv.org/content/10.1101/085050v2
18. Gros C, Sanders AD, Korbel JO, et al (2021) ASHLEYS: automated quality control for single-cell Strand-seq data. Bioinformatics 37:3356–3357
19. Kent WJ, Sugnet CW, Furey TS, et al (2002) The human genome browser at UCSC. Genome Res 12:996–1006
20. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal 17:10–12