Chapter 15

Next-Generation Analysis of Trypanosomatid Genome Stability and Instability

Emma M. Briggs, Catarina A. Marques, Joao Reis-Cunha, Jennifer Black, Samantha Campbell, Jeziel Damasceno, Daniella Bartholomeu, Kathryn Crouch, and Richard McCulloch

Abstract

Understanding the rate and patterns of genome variation is becoming ever more amenable to whole-genome analysis through advances in DNA sequencing, which may, at least in some circumstances, have supplanted more localized analyses by cellular and genetic approaches. Whole-genome analyses can utilize both short- and long-read sequence technologies. Here we describe how sequence generated by these approaches has been used in trypanosomatids to examine DNA replication dynamics, the accumulation of modified histone H2A due to genome damage, and evaluation of genome variation, focusing on ploidy change.

Key words Next-generation sequencing, DNA replication, MFA-seq, ChIP-seq, DNA damage, Single nucleotide polymorphisms, Copy number variation, Ploidy

1 Introduction

The genome is the crucible of life for cellular organisms. Accurate copying and transmission of DNA content from parent to offspring is needed for the propagation of cellular life. However, the genetic content cannot remain unchanged, since alterations in sequence are needed for adaptation to changing or hostile environments, and are the basis for selection by evolution. Normally these content changes accumulate slowly and often occur in a limited way, such as through single base changes or small insertions and deletions, subtly altering the protein and RNA products that are encoded. Other changes are more dramatic, either because they happen at high rates or because they occur through larger scale changes in the genome. These dramatic changes are not frequently found as examples of evolution but instead are normally genetically programmed reactions that reflect specific aspects of organism biology and

Paul A. M. Michels et al. (eds.), *Trypanosomatids: Methods and Protocols*, Methods in Molecular Biology, vol. 2116, https://doi.org/10.1007/978-1-0716-0294-2_15, © The Author(s) 2020

development. Prominent examples of such adaptive change are immunoglobulin gene rearrangements during B- and T-cell development [1, 2], mating type switching during yeast life cycles [3], and genome reconstruction in a range of ciliates [4–6]. In all cases, understanding of the timing, rate and mechanism of such dramatic change relative to the slower accrual of evolution-directing mutations is greatly aided by mapping the nature and location of the changes. Such questions are hugely easier to address, and more comprehensively evaluated, using next generation sequencing approaches that study such biology on a genome-wide scale.

Trypanosomatids—parasites within the wider order Kinetoplastae—provide a rich source of adaptive change, including frequent changes in the content and location of genes encoding the critical variant surface glycoprotein coat that underlies immune evasion by antigenic variation in *Trypanosoma brucei* [7], and the remarkable levels of genome-wide mosaic aneuploidy and gene amplification that are seen during *Leishmania* growth and dissemination [8, 9]. All such processes are amenable to analysis through next generation sequencing. Here, we provide details of three broad approaches that have been applied to date to understand the reactions that underlie stable transmission and directed variation in trypanosomatid genome content. First, we explain how the initiation and progression of DNA replication has been mapped by Marker Frequency Analysis (MFA-seq) [10, 11]; elsewhere, we have discussed how this approach compares with other strategies to map replication initiation, either using next generation sequencing or not [12]. Second, we explain how sites of DNA damage may be mapped in trypanosomatids [13], exploiting chromatin immunoprecipitation of histone γH2A, a phosphorylated variant of core histone H2A, on residue Thr130, which is generated in response to a range of DNA insults [14, 15]. Third, whole-genome sequencing, to date using Illumina methodology, can be exploited to map patterns of single nucleotide polymorphism (SNP) accumulation during growth [16], to determine patterns of chromosome and gene copy number variation (CNV) [17–19], and to predict structural changes, such as translocations [16, 20]; here, as an example, we describe the evaluation of chromosome CNV (CCNV) in *Leishmania major*, using freely available files and datasets. We have described each of the three approaches such that they can be followed individually, rather than relying on cross-checking on materials and protocols between the approaches.

2 Materials

2.1 MFA-seq Analysis of Replication Dynamics

2.1.1 Cell Preparation and Staining with Propidium Iodide

1. 1× PBS supplemented with 5 mM of EDTA.
2. 100% methanol (at 4 °C) or 1% formaldehyde (methanol-free) diluted in 1× PBS (*see* **Note 1**).
3. Propidium iodide (Sigma Aldrich).
4. RNase A (Sigma Aldrich).
5. Triton X-100 (Promega).
6. BD Falcon™ 12 × 75 mm (5 ml) tube with a 35 μm nylon mesh cell strainer cap (BD Biosciences, Cat. No. 352235) or other type of tube compatible with the FACS system being used.

2.1.2 Cell Sorting, DNA Extraction, and NGS

1. Fluorescence-activated cell sorting (FACS) system (e.g., BD FACSAria™ Cell Sorter).
2. Lysis Buffer: 1 M NaCl, 10 mM EDTA, 50 mM Tris–HCl pH 8.0, 0.5% SDS, 0.4 mg/ml Proteinase K, and 0.8 μg/ml of glycogen [21].

2.1.3 Genomic DNA Extraction and Sequencing

1. DNeasy Blood and Tissue DNA extraction kit (Qiagen).
2. Nextera® XT DNA Sample Preparation kit (Illumina) or Tru-Seq® DNA Sample Preparation kit (Illumina).
3. Illumina MiSeq 250 bp paired-end sequencing system or Illumina HiSeq 100 bp paired-end sequencing system.

2.1.4 Marker Frequency Analysis

All analysis can be run on a computer running UNIX or Linux with version 2.7 or higher of Python. The following **software** needs to be installed before MFA-seq analysis can be run.

1. FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/).
2. fastq-mcf (https://github.com/ExpressionAnalysis/ea-utils/blob/wiki/FastqMcf.md).
3. Bowtie2 (http://bowtie-bio.sourceforge.net/bowtie2/index.shtml).
4. Samtools (http://www.htslib.org).
5. Bedtools (https://bedtools.readthedocs.io/en/latest/).
6. pysam package (https://pypi.org/project/pysam/) (version 0.8.3 recommended).
7. mfaseq.py (https://github.com/CampbellSam/MFAseq).
8. ggplot2 (https://ggplot2.tidyverse.org) and R package (https://www.r-project.org), or Prism (GraphPad software Inc.).

The following files must be obtained prior to the analysis:

1. adapters.fa—.fa file containing the adapter sequences used in the library preparation and sequencing processes.
2. refgenome.fa—fasta file containing the reference genome to which the sequencing data are going to be aligned to. This can be obtained from TriTrypDB.org (Downloads → Data Files → Current_Release).

2.2 ChIP-seq Analysis of γH2A Distribution

2.2.1 Chromatin Immunoprecipitation

1. ChIP-IT Express Enzymatic Kit (Active Motif®).
2. Anti-γH2A antibody.
3. Formaldehyde (methanol-free).
4. Serum-free media, as appropriate for trypanosomatids.
5. Dounce homogenizer.
6. Magnetic rack.

2.2.2 Library Preparation

1. TrueSeq ChIP Library Preparation kit (Illumina).

2.2.3 qPCR

1. SYBR™ Select Master Mix (ThermoFisher).

2.2.4 Data Analysis

All analyses can be carried out on a computer running UNIX or Linux with the following software installed:

1. FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/).
2. TrimGalore (https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/).
3. Bowtie2 (http://bowtie-bio.sourceforge.net/bowtie2/index.shtml).
4. Samtools (http://www.htslib.org).
5. DeepTools (https://deeptools.readthedocs.io/en/develop/).

2.3 Next Generation Analysis of Instability

2.3.1 Hardware

1. A computer running UNIX, Linux or is suitable for all the procedures detailed here (*see* **Note 23**).

2.3.2 Datasets

1. To perform chromosomal copy number variation (CCNV) evaluations, a chromosome-level assembly of the desired reference genome, as well as whole-genome sequencing reads preferentially in FASTQ format with at least 20× of coverage is needed. Lower genome coverages could be used; however, they can compromise the ploidy analysis. The advantage of using FASTQ instead of fasta reads is the base quality information, which can be used to more accurately identify SNPs and genomic alterations. Although not required in all ploidy

estimation methodologies, the use of a General Feature Format (GFF) file containing the genome coordinates of all annotated genes could be important to perform CCNV in complex genomes or to polish the ploidy estimation.

Genomic reads generated by Illumina (https://www.illumina.com/documents/products/illumina_sequencing_introduction.pdf) are suggested due to their high depth and base quality; however, this analysis can also be performed with reads from other technologies as 454 [22] and Ion torrent (https://www.thermofisher.com/br/en/home/brands/ion-torrent.html) [23]. The use of long-reads such as generated by the PacBio Sequel/RSII (https://www.pacb.com/smrt-science/smrt-sequencing/) and Oxford Nanopore (https://nanoporetech.com/) to perform ploidy estimations is possible, but this analysis would need a different toolset from the one described in this chapter.

2. Chromosome-level assembly of the *Leishmania major* Friedlin reference genome (version 41), available at TriTrypDB (http://tritrypdb.org/tritrypdb/). Version 41 was used for the analysis described here, but new versions can also be used:

   http://tritrypdb.org/common/downloads/Release_41/LmajorFriedlin/fasta/data/TriTrypDB-41_LmajorFriedlin_Genome.fasta.

3. General Feature Format (GFF) file containing the genome coordinates of all annotated genomic features (*see* **Note 1**), also available at TriTrypDB: http://tritrypdb.org/common/downloads/Release_41/LmajorFriedlin/gff/data/TriTrypDB-41_LmajorFriedlin.gff.

4. Files with whole genome short sequencing reads (100 bases) from a *Leishmania major* isolate. The file we will use here is in a FASTQ format and can be downloaded from the Sequence Read Archive (SRA) of the National Center for Biotechnology Information (NCBI): https://www.ncbi.nlm.nih.gov/sra/?term=SRR6369659 (*see* **Notes 24–28**).

2.3.3 Software (See **Note 29**)

1. *SRAtoolkit*. Package used to easily download and process reads from the NCBI SRA database.

   Download link: https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/.

   Documentation link: https://www.ncbi.nlm.nih.gov/books/NBK242621/.

2. *FASTQC*. Program used to evaluate the quality of FASTQ reads.

   Download link: https://www.bioinformatics.babraham.ac.uk/projects/download.html#fastqc.

Documentation link: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

3. *Trimmomatic*. Program used to evaluate and remove low quality reads and adapter sequences. Download link: http://www.usadellab.org/cms/?page=trimmomatic.
   Documentation link: http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf.
4. *BWA*. Software used to map reads in a low-divergent genome.
   Download link: https://sourceforge.net/projects/bio-bwa/.
   Documentation link: http://bio-bwa.sourceforge.net/bwa.shtml.
5. *SAMtools*. SAMtools is a group of tools that allow the user to manipulate files in SAM or BAM formats.
   Download link: http://www.htslib.org/download/.
   Documentation link: http://www.htslib.org/doc/samtools.html.
6. *BCFtools*. A set of tools that manipulate Variant Call Format (VCF) files.
   Download link: https://samtools.github.io/bcftools/.
   Documentation link: https://samtools.github.io/bcftools/bcftools.html.
7. *BEDtools*. A group of tools to evaluate and compare genome alignment and annotation features.
   Download link: https://bedtools.readthedocs.io/en/latest/content/installation.html.
   Documentation link: https://bedtools.readthedocs.io/en/latest/content/bedtools-suite.html.
8. *GATK*. A group of tools with a focus on variant discovery and genotyping.
   Download link: https://software.broadinstitute.org/gatk/download/.
   Documentation link: https://software.broadinstitute.org/gatk/documentation/.
9. *R*. R is a free software environment for statistical computing and graphics.
   Download link: https://cran.r-project.org/mirrors.html.
   Documentation link: https://www.r-project.org/.
10. *R library "ggplot"*.
    Documentation link: https://www.rdocumentation.org/packages/ggplot2/versions/3.1.0/topics/ggplot.
11. *Picard Tools*. Command line tools for manipulating high-throughput sequencing data.
    Download link: https://broadinstitute.github.io/picard/.

Documentation link: https://broadinstitute.github.io/picard/command-line-overview.html.

12. *vcflib*. A C++ library for parsing and manipulation VCF files. Download and documentation link: https://github.com/vcflib/vcflib.

3 Methods

3.1 Mapping Replication Initiation by MFA-seq

All steps are done at room temperature, unless stated otherwise. All volumes are for the processing of 1 × 10^8 cells (*see* **Note 2**). The fixing protocols are different for *T. brucei* procyclic form cells and bloodstream form cells (*see* **Note 1**).

3.1.1 Cell Preparation and Staining with Propidium Iodide—T. brucei Procyclic Cell Forms and L. major/L. mexicana Promastigote Cells

1. From a log-phase culture (1 × 10^7 cells/ml *T. brucei* procyclic form cells; 5 × 10^6 cells/ml *L. major* or *L. mexicana* promastigote form cells), collect 1 × 10^8 cells by centrifuging for 10 min at 1000 × *g*.
2. Wash the pellet in 10 ml of 1× PBS supplemented with 5 mM EDTA (*see* **Note 3**), and centrifuge again for 10 min at 1000 × *g*.
3. Resuspend the cells in 1.2 ml of 1× PBS supplemented with 5 mM EDTA, and add 2.8 ml of ice-cold 100% methanol, in a drop-wise fashion while vortexing gently; the final fixing solution will be of 70% methanol, and the concentration of 2.5 × 10^7 cells/ml (*see* **Note 4**).
4. Wrap the tube in aluminum foil, and store at 4 °C (from overnight to up to 3 weeks) until the sorting.
5. Centrifuge the cells for 10 min at 1000 × *g*, at 4 °C.
6. Wash the pellet in 1 ml of 1× PBS supplemented with 5 mM EDTA, and centrifuge again for 10 min at 1000 × *g*, at 4 °C.
7. Resuspend the pellet in 4 ml of 1× PBS supplemented with 5 mM EDTA, 10 μg/ml of propidium iodide, and 10 μg/ml of RNase A.
8. Incubate for 45 min at 37 °C, protected from light.
9. Transfer the cells to a 5 ml BD Falcon™ tube through the 35 μm nylon mesh cell strainer cap.
10. Keep the cells on ice and protected from light.

3.1.2 Cell Preparation and Staining with Propidium Iodide—T. brucei Bloodstream Form Cells

1. From a log-phase culture (1 × 10^6 cells/ml *T. brucei* bloodstream form cells), collect 1 × 10^8 cells by centrifuging for 10 min at 1000 × *g*.
2. Wash the pellet in 10 ml of 1× PBS supplemented with 5 mM EDTA (*see* **Note 3**), and centrifuge again for 10 min at 1000 × *g*.

3. Resuspend the pellet first in 500 μl of 1× PBS, and then add 9.5 ml of 1% formaldehyde in 1× PBS.
4. Incubate the cells for 10 min at room temperature.
5. Centrifuge the cells for 10 min at 1000 × *g*.
6. Wash the pellet in 10 ml of 1× PBS, and then centrifuge for 10 min at 1000 × *g*.
7. Resuspend the cells in 4 ml of 1× PBS, so to have a final concentration of 2.5×10^7 cells/ml (*see* **Note 4**).
8. Wrap the tube in aluminum foil, and store at 4 °C (from overnight to up to 3 weeks) until the sorting.
9. Centrifuge the cells for 10 min at 1000 × *g*.
10. Resuspend the pellet, first in 1 ml of 0.01% Triton X-100 diluted in 1× PBS, and then add the remaining 9 ml of 0.01% Triton X-100 diluted in 1× PBS; incubate for 30 min at room temperature.
11. Centrifuge the cells for 10 min at 700 × *g* (*see* **Note 5**).
12. Wash the cells in 10 ml of 1× PBS, and then centrifuge for 10 min at 1000 × *g*.
13. Resuspend the pellet first in 4 ml of 1× PBS with 10 μg/ml of propidium iodide, and 100 μg/ml of RNase A.
14. Incubate for 1 h at 37 °C, protected from light.
15. Transfer the cells to the 5 ml BD Falcon™ tube through the 35 μm nylon mesh cell strainer cap.
16. Keep the cells on ice and protected from light.

3.1.3 Cell Sorting into G1, S, and G2/M Phase Subpopulations

1. Immediately before introducing the sample into the sorting machine, gently vortex and refilter the sample through the cell strainer cap.
2. Run the samples through the FACS machine—the settings (*see* **Note 6**) will vary between systems and different FACS facilities will have different protocols of use in place—contact them first before running the experiment. Propidium iodide is excited by the 488 nm or 561 nm lasers and detected by the 695/40 nm or 610/20 nm detectors (respectively).
3. On the FACS machine software, create the following plots: side scatter area (SSC-A) vs forward scatter area (FSC-A), and PE area (PE-A) vs PE width (PE-W).
4. Initially, run ~ 50,000 events to get an idea of the quality of the sample and to set the gates for the sorting. First, the SSC vs FSC scatter plot will inform on the shape of the cells, while the PE-A vs PE-W scatter will allow you to draw a gate to include only singlet events (excluding agglomerates of cells). From this gate, generate a histogram of cell count vs PE-A. This will allow

you to see the "typical" cell cycle profile. On this graph, set the gates to sort cells in G1, S (can be divided into early S and late S populations), and G2/M phase populations.

5. Sort the samples into the different populations (*see* **Note 7**) into collection tubes (*see* **Note 8**) containing 200 μl of lysis buffer; the collection chamber was kept at 4 °C (*see* **Note 9**). If various collection tubes are needed, keep them on ice until the next step.
6. After the sorting, homogenize the sorted cells and lysis solution, and incubate for 2 h at 55 °C (*see* **Note 10**).
7. The lysate can be stored at −20 °C, for later gDNA extraction (*see* **Note 11**), or used straight for gDNA extraction.

3.1.4 gDNA Extraction from the G1, S and G2/M Phase Subpopulations

With the exception of the lysis one, all steps are as described in the instructions of the Qiagen DNeasy Blood and Tissue kit.

1. Thaw frozen lysates at 37 °C and, in the case of multiple sorting sessions, pool per cell cycle stage; take note of the total volume per sample.
2. To each sample, add 100% ethanol, in a volume that corresponds to one-third of the sample's volume, and vortex thoroughly.
3. Transfer 800 μl onto a DNeasy Mini spin column (Qiagen).
4. Centrifuge for 1 min at 6000 × *g* and discard the follow-through.
5. Repeat **steps 3** and **4** until all the lysate has been passed through the column.
6. Wash the column with 500 μl of buffer AW1 (Qiagen).
7. Centrifuge for 1 min at 6000 × *g* and discard the flow-through.
8. Add 500 μl of buffer AW2 (Qiagen) to the column.
9. Centrifuge for 3 min at 20,000 × *g* and discard the flow-through.
10. Transfer the column to a 1.5 ml Eppendorf and add 50 μl of Buffer AE (Qiagen).
11. Centrifuge for 1 min at 6000 × *g*.
12. Store the gDNA at −20 °C, until needed for library prep.

3.1.5 Sequencing Library Preparation

Library preparation and sequencing were performed at facilities such as the Glasgow Polyomics at the University of Glasgow, or Eurofins Genomics (Germany). Seek advice on which kit and sequencing system to use, as these evolve rapidly and more efficient ones might be available at the time of your experiment. In our work we have used the following:

1. DNA libraries were prepared with the Nextera® XT DNA Sample Preparation kit (Illumina) or the TruSeq® DNA Sample Preparation kit (Illumina), according to the manufacturer instructions.
2. The samples were multiplexed and sequenced using either the Illumina MiSeq paired-end 250 bp sequencing system or the Illumina HiSeq paired-end sequencing system, according to the manufacturer instructions.

3.1.6 Marker Frequency Analysis

1. Assess the quality of the sequenced data (.fastq files) using the FastQC software package, a quality control tool. Type "FastQC" on the command line/terminal; if installed, it opens a graphical interface in which the files are uploaded and analyzed. The results are shown in a graph showing quality score versus position in read. Decide on a threshold—in our studies we have used a threshold score of 20.
2. Back in the command line/terminal, use the fastq-mcf tool to simultaneously trim the sequencing reads according to the defined threshold (in this case, 20), and remove the sequences of the adaptors used for the library preparation and sequencing processes.

```
# Type fastq-mcf -h to open the help information on
this tool; it will provide information on the
commands used.
fastq-mcf -h

# -q allows to define the quality threshold; -w
window size for the quality trimming; adapters.fa
refers to the file that has the adapter sequences; -o
output file; only the files for the bloodstream
form early S (BSF_ES_L001_R1.fastq and
BSF_ES_L001_R2.fastq) are shown as an example; there
are two files per sample as the sequencing was
paired-end. This will generate files containing the
trimmed sequences (e.g.
BSF_ES_L001_R1_TRIMMED.fastq).
 fastq-mcf -q 20 -w 5 adapters.fa
$BSF_ES_L001_R1.fastq BSF_ES_L001_R2.fastq -o
BSF_ES_L001_R1_TRIMMED.fastq -o
BSF_ES_L001_R2_TRIMMED.fastq
```

3. Align the trimmed files onto the reference genome using bowtie2 and prepare the output for downstream analysis by using samtools to convert the output to binary format and sort it by coordinate. For convenience, this can be achieved by piping the output of bowtie2 directly to samtools, thus preventing the generation of a large intermediate file.

```
 # First need to index the reference genome.
 bowtie2-build refgenome.fa indexed_refgenome

# Align the trimmed sequences to the indexed
reference genome. Use the --very-sensitive-local
option – this does not attempt to map 100% of the
read, it will try to map as much of it as it can,
while tolerating the not complete mapping of its
ends; -p leads bowtie2 to run a defined number of
parallel search threads, with each of these running
on a different processor or core – this will depend
on the computer or server/cluster service being used
(in this example is 4); -x will tell bowtie2 where
to find the indexed reference genome; -1 and -2 are
used in the case of paired-end reads, these inform
bowtie2 where to find the forward (-1) and reverse
(-2) reads; bowtie2 will generate the .sam file
containing the actual alignment, as well as a .log
file, which is a report of the alignment, including
statistics and information on the overall alignment
rate. Pipe the output directly to samtools to
convert the output to binary BAM format and sort it.
bowtie2 -p 4 --very-sensitive-local -x
indexed_refgenome -1 BSF_ES_L001_R1_TRIMMED.fastq -2
BSF_ES_L001_R2_TRIMMED.fastq 2> BSF_ES.log |
samtools view -bS - | samtools sort - -o BSF_ES.bam
```

4. Index the BAM file using samtools. The index is required for many downstream analysis steps.

```
 # Create an index for each alignment file using
SAMtools index
samtools index input.bam
```

5. Apply custom MFAseq Python script to perform marker frequency analysis. The script used to perform the MFAseq analysis is available at https://github.com/CampbellSam/MFAseq. Python version 2.7 or higher is required to run this script and the pysam package (version 0.8.3 recommended) is also a dependency that must be installed prior to running the analysis. Running the mfaseq.py script on the command line with no input will provide usage instructions that are also available in the script header. Two alignment files are the only required input but the window size and output file can also be specified as input parameters if necessary. This script will produce output in wiggle format (.wig) that can then be viewed as a custom track on EuPathDB using the GenomeBrowser tool or plotted for visualization using ggplot in R or Python. Details to alter the appearance of the EuPathDB track are also included within

the script and will normally be printed as standard out when the output file is generated. The .wig data can also be imported to excel and plotted in Excel or Prism (GraphPad). Bigwig output can be visualized on other genome viewers, such as IGV.

```
 # The Python script will calculate a read depth
ratio between the two sets of alignments for a
specified window size along each chromosome and
output these values as a list in .wig format. The
default window size is 2500bp and the ratio is
calculated as first alignment file/second alignment
file.
mfaseq.py --file1 alignmentFile1.srt.bam --file2
alignmentFile2.srt.bam --window [window size in
bases] --wig [output file name - defaults to
standard out]
```

3.2 Localization of DNA Damage by γH2A ChIP-seq

For the ChIP procedure described in Subheadings 3.2.1–3.2.5, all buffers and reagents are supplied in the ChIP-IT Express Enzymatic Kit (Active Motif®), with the exceptions of appropriate serum-free medium, formaldehyde, and an anti-γH2A antibody.

3.2.1 Chromatin Extraction and Enzymatic Shearing

1. Harvest 1×10^8 parasites via centrifugation at 1200 × *g* for 10 min, at 4 °C. Remove media from cell pellet.
2. Resuspend parasites in 10 ml Fixation Solution consisting of 10% formaldehyde in appropriate medium (serum-free). Incubate with agitation at room temperature for 5 min (*see* **Note 12**).
3. Add 1 ml of 10× Glycine to halt fixation.
4. Centrifuge at 1200 × *g* for 10 min, at 4 °C, to pellet parasites. Pour off supernatant.
5. Resuspend in 10 ml 1× Glycine and incubate with agitation for 5 min at room temperature. Centrifuge as before to pellet parasites.
6. Resuspend parasites in 10 ml ice-cold 1× PBS and pellet again via centrifugation at 1200 × *g* for 10 min, at 4 °C. Remove all PBS by pouring and pipetting. At this stage the pellet may be stored at −80 °C.
7. Resuspend cells in 1 ml ice-cold Lysis Buffer supplemented with 5 μl protease inhibitor cocktail (PIC) and 5 μl phenylmethylsulfonyl fluoride (PMSF). Pipette and vortex gently to resuspend. Transfer to a 1.5 ml Eppendorf and incubate the on ice for 30 min. During this time prepare the Enzymatic Shearing Cocktail Solution by adding 1 μl of the Enzyme Shearing Cocktail to 99 μl 50% glycerol.

8. Transfer samples in the Lysis Buffer to a Dounce homogenizer and Dounce with ten strokes on ice. Ensure cells have lysed by checking 10 μl in a hemocytometer under a phase contrast microscope. Continue to Dounce until cells have lysed and nuclei are released (*see* **Note 13**).
9. Transfer lysed cells to a 1.5 ml Eppendorf. Centrifuge at 2400 × *g* for 10 min, at 4 °C, to isolate the nuclei. Carefully remove and discard the supernatant.
10. Resuspend the nuclei in 350 μl Digestion Buffer (supplemented with 1.75 μl 100 mM PMSF and 1.75 μl PIC). Incubate at 37 °C for 5 min.
11. Add 17 μl of prepared Enzymatic Shearing Cocktail solution to each sample and incubate at 37 °C for 5 min (*see* **Note 14**).
12. To stop the shearing reaction, add 7 μl ice-cold 0.5 M EDTA. Incubate on ice for 10 min.
13. Centrifuge the sheared chromatin at 18,000 × *g* for 10 min, at 4 °C.
14. Transfer the supernatant, containing sheared chromatin, into 1.5 ml Eppendorf tubes as 50 μl aliquots. Chromatin can be used immediately for downstream steps or stored at −80 °C.

3.2.2 DNA Clean Up, Quantification and Shearing Check

1. If frozen, thaw one 50 μl aliquot on ice. Add 150 μl dH_2O to the sample, then 10 μl 5 M NaCl.
2. Incubate at 65 °C for 4–16 h to reverse the cross-links.
3. Add 1 μl RNase A and incubate at 37 °C for 15 min.
4. Add 10 μl Proteinase K and incubate at 42 °C for 1.5 h.
5. (a) Quantify the sample (*see* **Note 15**). Multiple the resulting concentration by 4.42 to account for dilution of the sample in the above steps and calculate the DNA content of prepared chromatin. Concentrations >15 ng/μl are expected.

   (b) Assess shearing efficiency. Add 4 μl of 6× loading dye to 16 μl of sample and load 5 and 10 μl to wells of a 1% TAE agarose gel. Run at 100 V for 1 h to observe fragment sizes. If optimal, shearing should result in 200–1500 bp bands.

3.2.3 Chromatin-Immunoprecipitation (ChIP)

For immunoprecipitation of γH2A we used 11 μl of antibody (produced in house).

1. If frozen, thaw chromatin on ice. Transfer 10 μl of chromatin to a 1.5 ml Eppendorf and store at −20 °C. This will serve as the Input sample and will be used in downstream steps.

2. Set up ChIP reaction(s) as per the table below (*see* **Note 16**). **It is important to add the antibody last**.

Reagent	Volume
Protein G magnetic beads	25 μl
ChIP buffer 1	20 μl
Sheared chromatin[a]	61–150 μl
Protease inhibitor cocktail	2 μl
dH_2O	Up to final volume 200 μl
Antibody (1–3 μg)	*X* μl
Total volume	*200 μl*

[a] As little as 1 μg of chromatin can be used successfully depending upon the target (*see* **Note 17**)

3. Incubate on an end-to-end rotator at 4 °C overnight.
4. Spin tubes briefly to collect liquid from inside the cap.
5. Place tubes on a magnetic rack and allow beads to gather to the side of the tubes.
6. Carefully remove the supernatant and discard.
7. Remove tubes from the magnetic rack and add 800 μl of ChIP Buffer 1. Fully resuspend the beads by pipetting up and down to wash (*see* **Note 18**).
8. Replace tubes on the magnetic rack. Wash as before with 800 μl ChIP Buffer 2.
9. Perform a second wash with 800 μl ChIP Buffer 2. After this final wash step, remove as much supernatant as possible. Use a 200 μl pipette if necessary.
10. To elute, resuspend the washed beads in 50 μl Elution Buffer AM2.
11. Incubate for 15 min on an end-to-end rotator, at room temperature.
12. Briefly spin tubes to collect liquid from inside the cap.
13. Add 50 μl Reverse Cross-linking Buffer to the beads and mix by pipetting up and down.
14. Place tubes on the magnetic rack and allow the beads to pellet. Transfer the supernatant, which contains the eluted chromatin, into a fresh 1.5 ml Eppendorf. This is the Elute sample.
15. Retrieve the Input sample of chromatin and thaw on ice. Add 88 μl of ChIP Buffer 2 and 2 μl 5 M NaCl to the Input chromatin only.

16. Incubate both the ChIP Elute and Input samples at 65 °C for 2.5 h (*see* **Note 19**). The samples can be stored at −20 °C at this point if necessary.
17. Return the tubes to room temperature and spin briefly to remove any liquid from the cap. Add 1 μl Proteinase K and mix well.
18. Incubate at 37 °C for 1 h. During this step place the Proteinase K Stop Solution at room temperature for at least 30 min.
19. Return the samples to room temperature and add 2 μl Proteinase K Stop Solution. Briefly spin tubes. DNA can now be used immediately for downstream analysis or stored at −20 °C.

3.2.4 ChIP-qPCR

DNA acquired from ChIP reactions can be used in qPCR analysis to compare the presence of a specific DNA sequence in Eluted samples relative to Input, and hence infer if binding of the target (i.e., γH2A) takes place at that sequence. Primer efficiency must be ascertained before use.

1. Dilute Input DNA 1:10 in dH_2O (*see* **Note 20**). qPCR needs to be performed with both Input and Eluted samples for each primer pair.
2. For each reaction combine 1 μl DNA (diluted Input or Elute), 400 nM each primer, 1× SYBR Select Master Mix, and H_2O to a final volume of 20 μl. We recommend performing each reaction in triplicate, as well as performing an H_2O negative control for each primer pair.
3. Perform qPCR with the following conditions: 50 °C for 2 min and 95 °C for 2 min, followed by 40 cycles of 95 °C for 15 s, 59 °C for 15 s, and 72 °C for 1 min. Take fluorescence intensity measurements during the extension step (72 °C for 1 min) of each cycle.
4. Calculate the average C_T for each condition from the three replicate values.
5. Calculate the percentage of Input sequence in the Elute sample using the table below.

	Raw average Input C_T (10%)	**Adjust Input to 100%**	**Raw average Elute C_T**	**Percentage input**
Calculation	Average of three C_T values	Ct input—3.322[a]	Average of three C_T values	$100 \times 2^{(\text{adjusted input} - \text{Elute})}$
Example	28.5	25.178	30.6	2.33%

[a] The number of cycles to subtract is calculated from the dilution factor used (i.e., for a 1:10 dilution, log2 of 10 is 3.322)

3.2.5 ChIP-seq

For library preparation we recommend the TrueSeq ChIP Library Preparation kit from Illumina. The following steps then must be followed for both the Input and Elute sequencing files.

1. Assess read quality using FASTQC. This is done in the same manner as for MFA-seq (*see* Subheading 3.1.6).
2. Trim reads to remove Illumina adaptor sequences and based with quality scores <20.

```
# This can be done using Trim Galore, which will by
default remove Illumina adapter sequences and bases
with a Phred score < 20. Trim_galore also has a
database of adaptors used in commercial library prep
kits and will trim these if they are found. Custom
adaptors can be supplied at the command line.
 # For single-end reads
 trim_galore <file>

 #For paired-end reads
 trim_galore -paired <reads1File> <reads2File>
```

3. Map to the reference genome using Bowtie2 in "very-sensitive" mode and prepare BAM files for downstream analysis. This can be performed in the same manner as for MFA-seq (*see* Subheading 3.1.6) and will first require a reference genome. If single end reads are used, the command can be altered as follows:

```
 bowtie2 -p 4 --very-sensitive-local -x
indexed_refgenome -U <readsFile> 2> BSF_ES.log |
samtools view -bS - | samtools sort - -o
<output.bam>
```

4. In order to avoid mapping artefacts, remove reads with a MapQ value <0 using SAMtools (*see* **Note 21**).

```
 # This is done with the SAMTools view function
 Samtools view -h -q 1 <input.bam> > <output.bam>

 # Create an index for each alignment file using
SAMtools index
 samtools index input.bam
```

5. To obtain the resulting ChIP signal across the genome, the read-depth of Elute samples is calculated relative to the corresponding Input sample. SES normalisation is recommended [24]. This can be done using the bamCompare tool from Deeptools (*see* **Note 22**).

```
 #Use bamCompare to calculate signal enrichment
normalized to Input
bamCompare -b1 <Elute.bam> -b2 <Input.bam> --
scaleFactorsMethod SES -o output.bw
```

6. For further analysis, the resulting normalized signal enrichment file (in bigwig format) can be directly used with Deeptools computeMatrix, plotHeatmap, and plotProfile tools. This can be used to plot ChIP signal across specific regions of interest (e.g., coding regions), which can be supplied as a bed format file of coordinates. Bigwig output can also be visualized on genome viewers, such as IGV.

3.3 Next Generation Sequencing to Examine Genome Variation

3.3.1 Download Data Sets

In the command line (*see* **Note 30**), use wget and fastqdump to download the appropriated data files (*see* **Note 31**):

```
#For the genome reference file:
>wget <link-to-genome-file>
Example command:
>wget
http://tritrypdb.org/common/downloads/release41
/LmajorFriedlin/fasta/data/TriTrypDB-
41_LmajorFriedlin_Genome.fasta.

#For the GFF file:
>wget <link-to-GFF-file>
Example command:
> wget
http://tritrypdb.org/common/downloads/release-
41/LmajorFriedlin/gff/data/TriTrypDB-
41_LmajorFriedlin.gff.

# Download the whole genome sequencing reads:
> fastq-dump.2.8.1 --split-files <Read.file.ID>
Example command:
> fastq-dump.2.8.1 --split-files SRR6369659
```

3.3.2 Run FastQC to Evaluate the Read Library Quality

FastQC evaluates the quality of the read libraries and generate several report files. Among them, the ∗.html file contains the summary of the results (Fig. 1a, b). It is desirable that the mean read quality is higher than phred 30, meaning one expected error in 1000 bases (*see* **Note 32**).

```
# For the file containing one group of the read pair
> fastqc <read.libray1> <read.libray2>
Example command:
> fastqc SRR6369659_1-paired.fastq
SRR6369659_2-paired.fastq
```

3.3.3 Run Trimmomatic

Trimmomatic will exclude low quality reads, remove positions with low quality at the extremities of the quality reads and adaptors sequences.

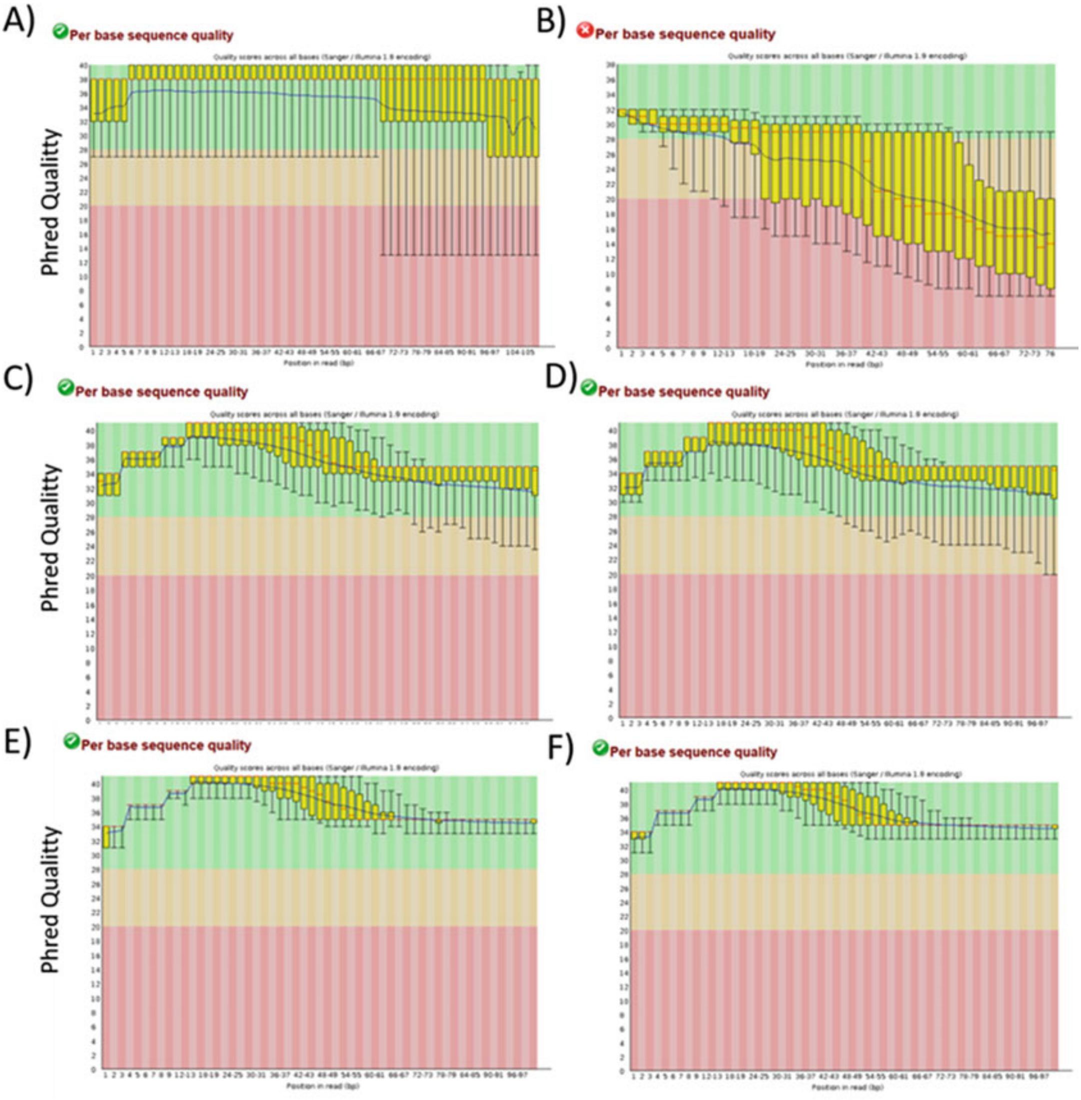


Fig. 1 FastQC read libraries base quality. In this image the *X* axis denotes the phred base quality and the *Y* axis represents the read position. (**a**) Read library with a mean quality higher than 30. (**b**) Read library with a mean quality lower than 30. Read libraries (**c**) SRR6369659_1.fastq and (**d**) SRR6369659_2.fastq before the trimming step. Read libraries (**e**) SRR6369659_1.fastq and (**f**) SRR6369659_2.fastq after the trimming step

```
#Trimmomatic runs as follow:
> java -jar trimmomatic-0.33.jar PE -phred33 <Read-
file-1.fastq> <Read-file-2.fastq>
<Paired_output_1.fastq> <Unpaired_output_1.fastq>
<Paired_output_2.fastq> <Unpaired_output_2.fastq>
ILLUMINACLIP:<Adapters fasta file>:<seed
mismatches>:<palindrome clip threshold>:<simple clip
threshold> LEADING:<quality> TRAILING:<quality>
SLIDINGWINDOW:<window size>:<required quality>
```

```
MINLEN:<size>
Example command:
> java -jar Trimmomatic-0.33/trimmomatic-
0.33.jar PE -phred33 SRR6369659_1.fastq
SRR6369659_2.fastq SRR6369659_1-paired.fastq
SRR6369659_1-UNpaired.fastq SRR6369659_2
-paired.fastq SRR6369659_2-UNpaired.fastq
ILLUMINACLIP:all_adapters:2:30:10 LEADING:25
TRAILING:25 SLIDINGWINDOW:5:30 MINLEN:50
```

Trimmomatic can be run using single-end or paired-end read libraries. The above example represents the command line by using paired-end reads, which requires two input files, `Read-file-1.fastq` and input file 2 is `Read-file-2.fastq`. Trimmomatic will generate four output files. The output files `Paired_output_1.fastq` and `Paired_output_2.fastq` contain the reads that have a corresponding high quality read pair with each other. On the other hand, the output files `Unpaired_output_1.fastq and Unpaired_output_2.fastq` contain the reads that did not have a high quality pair with each other.

At the end of the process, Trimmomatic will generate a summary of the process, as in the example below:

```
Input Read Pairs: 21125144 Both Surviving: 15021842
(71.11%) Forward Only Surviving: 1980418 (9.37%)
Reverse Only Surviving: 1406608 (6.66%) Dropped:
2716276 (12.86%)
```

This means that from the `21125144` reads present in the input files, `15021842 (71.11%)` were of high quality in both pair-end read files, `1980418 (9.37%)` were of high quality only in the first read library, `1406608 (6.66%)` were of high quality only in the second read library and `2716276 (12.86%)` presented low quality in both libraries, and were excluded. A comparison of the quality of read libraries before and after the trimming step can be seen in Fig. 1.

3.3.4 Mapping the Reads to the Reference Genome

BWA-MEM maps the filtered reads to the reference genome, generating an alignment file in Sequence Alignment/Map (SAM) format.

```
#Index the Reference genome
> bwa index <Reference-genome.fasta>
Example command:
> bwa index TriTrypDB-
41_LmajorFriedlin_Genome.fasta

#Align reads to the reference genome using BWA-mem:
> bwa mem -M <reference.genome.fasta>
```

```
<Paired_output_1.fastq> <Paired_output_2.fastq> >
Mapped-file.sam
Example command:
> bwa mem -M TriTrypDB-
41_LmajorFriedlin_Genome.fasta SRR6369659_1-
paired.fastq SRR6369659_2-paired.fastq >
SRR6369659-Mapped-file.sam
```

If necessary, the user could use all the "paired" and "unpaired" reads in this mapping analysis to improve the depth of coverage. To that end, it is possible to combine all reads in a single file, with the command:

```
> cat <Paired_output_1.fastq>
<Unpaired_output_1.fastq> <Paired_output_2.fastq>
<Unpaired_output_2.fastq> >
<allreads_file_name.fastq>

> bwa mem -M <reference.genome.fasta>
<allreads_file_name.fastq> > <All-reads-Mapped-
file.sam>
```

The samtools flagstat command generates a report (Mapped-file-info) with information on the number and proportion of mapped reads and number and proportion of properly paired mapped reads.

```
#Check alignment results using SAMtools flagstat:
> samtools flagstat <Mapped-file.sam> > <Mapped-
file-info>
Example command:
> samtools flagstat SRR6369659-Mapped-file.sam
> SRR6369659-Mapped-file-info
```

The flagstat results in the "SRR6369659-Mapped-file-info" can be seen below:

```
30055150 + 0 in total (QC-passed reads + QC-failed
reads)
11466 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
25150804 + 0 mapped (83.68% : N/A)
30043684 + 0 paired in sequencing
15021842 + 0 read1
15021842 + 0 read2
25081734 + 0 properly paired (83.48%: N/A)
25133580 + 0 with itself and mate mapped
5758 + 0 singletons (0.02%: N/A)
29784 + 0 with mate mapped to a different chr
22208 + 0 with mate mapped to a different chr
(mapQ>=5)
```

In summary, flagstat result showed that from the 30055150 reads in the input files, 25150804(83.68%) mapped in the reference genome TriTrypDB-41_LmajorFriedlin_Genome.fasta, where 25081734(83.48%) of the reads were properly paired (*see* **Note 33**).

3.3.5 Filter Alignment File by Mapping Quality and Convert .sam File to .bam

```
# Use SAMtools to filter the alignment file and
convert to the binary bam file.
> samtools view -bh <Mapped-file.sam> -q <30> -o
<Mapped-file-q30.bam>
Example command:
> samtools view -bh SRR6369659-Mapped-file.sam
-q30 -o SRR6369659-Mapped-file-q30.bam
```

Samtools view flag "-b" converts the SAM to a BAM compact file, while the flag "-q 30" filter the alignment file to only report reads that presented a mapping quality higher than Phred 30. Finally, the flag "-h" generates a header for the BAM file.

3.3.6 Sort .bam File

Samtools sort sorts alignments based on leftmost coordinates. This sorting is required by the majority of the downstream programs that work with alignment files.

```
# Sort the bam alignment file
> samtools sort <Mapped-file-q30.bam> -o <Mapped-
file-q30-sorted.bam>
Example command:
> samtools sort SRR6369659-Mapped-file-q30.bam
-o SRR6369659-Mapped-file-q30-sorted.bam
```

Generate the read depth coverage of each position in each chromosome

```
#Estimate the read depth of coverage in each genomic
position
> bedtools genomecov -ibam <Mapped-file-q30-
sorted.bam> -g <reference.genome.fasta> -d >
<Genome-coverage.bed>
Example command:
> bedtools genomecov -ibam SRR6369659-Mapped-
file-q30-sorted.bam -g TriTrypDB-
41_LmajorFriedlin_Genome.fasta -d > SRR6369659-
coverage.bed
```

The flag "-d" BEDTools will generate a file reporting the depth of coverage in each position of each chromosome. This file can be used to estimate the mean genome coverage as well as the mean coverage for each chromosome.

3.3.7 Estimate the Mean Genome Coverage

```
#Count the number of nucleotides in the reference
genome
> wc -l <Genome-coverage.bed> | awk '{print $1}' >
<Genome-positions>
Example command:
> wc -l SRR6369659-coverage.bed | awk '{print
$1}' > SRR6369659-Genome-positions
```

This command will count and save, in the "`Genome-positions`" file, the number of lines in the file "`Genome-coverage.bed`", which is equivalent to the number of nucleotides in the evaluated genome. In the example, the "`SRR6369659-Genome-positions`" contains the number "`32855095`".

```
#Estimate the sum of the read depth coverage (RDC)
of all genomic positions:
cat <Genome-coverage.bed> | awk '{print totalsize+=
$3}' | tail -n1 > <Genome-RDC>
Example command:
awk '{print totalsize += $3}' SRR6369659-
coverage.bed | tail -n1 > SRR6369659-Genome-RDC
```

This command will sum the RDC of each position in the genome. In the example, the "`SRR6369659-Genome-positions`" contains the number "`2144501882`". This value can vary if different mapper programs or versions are used.

The genome mean RDC can be obtained by dividing "`Genome-RDC`" by the "`Genome-positions`". In the example, by dividing `2144501882` by `32855095`, the mean genome coverage of ~ 65×.

3.3.8 Estimate the Chromosomal Somy by RDC

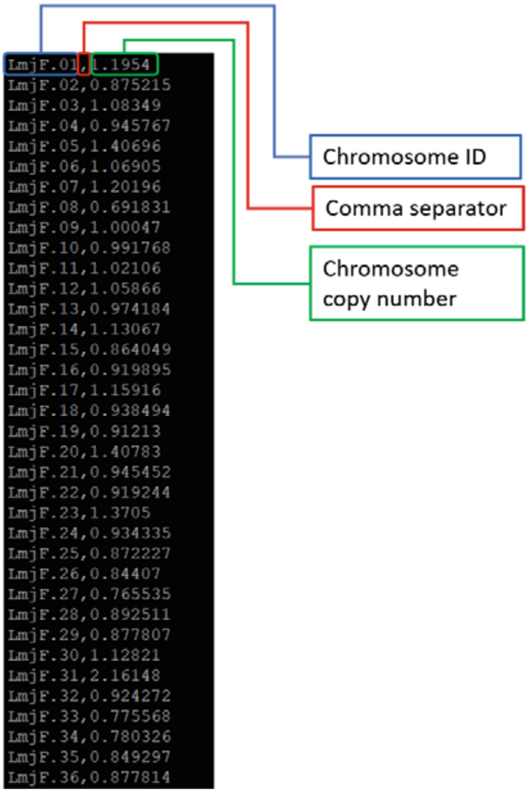
```
#Generate a list with all chromosome names:
> cat <Genome-coverage.bed> | awk '{print $1}' |
sort -u > <Chromosome-IDs>
Example command:
> cat SRR6369659-coverage.bed | awk '{print
$1}' | sort -u > SRR6369659-Chromosome-IDs

# Generate a table containing the normalized
chromosome copy number based on the ratio between
the mean genome RDC and the mean genome RDC.
> for i in $(cat <Chromosome-IDs>); do awk -v i="$i"
'$1 == i {print $0}' <Genome-coverage.bed> | awk -v
i="$i" ' {sum += $3} {NR} END {print
i","(sum/NR)/<genome_RDC>}' >> <CCNV-Table.csv>;
done
Example command:
> for i in $(cat SRR6369659-Chromosome-IDs); do
awk -v i="$i" '$1 == i {print $0}' SRR6369659-
```

```
coverage.bed | awk -v i="$i" ' {sum += $3} {NR}
END {print i","(sum/NR)/65}' >>
SRR6369659_CCNV_Table.csv; done
```

In this command line, the "genome_RDC" corresponds to the mean genome coverage, estimated in the 3.3.6 step, which is "65" in the example command.

The output CCNV-Table.csv file contains two columns separated by a coma (","), where the first column contains the chromosome ID and the second one contains the chromosomal copy number estimation. In this CCNV estimation, values of "0.5," "1," and "2" denotes that the chromosome has, respectively, "0.5", "1," or "2" copies per haploid genome. This means that, if the studied genome is mainly diploid, a value of "1" in this estimation represent two chromosomal copies (one per haploid genome), whereas a value of "1.5" represent three copies and a value of "2" represent four copies. An example of the table generated can be seen in Fig. 2.



```
LmjF.01,1.1954
LmjF.02,0.875215
LmjF.03,1.08349
LmjF.04,0.945767
LmjF.05,1.40696
LmjF.06,1.06905
LmjF.07,1.20196
LmjF.08,0.691831
LmjF.09,1.00047
LmjF.10,0.991768
LmjF.11,1.02106
LmjF.12,1.05866
LmjF.13,0.974184
LmjF.14,1.13067
LmjF.15,0.864049
LmjF.16,0.919895
LmjF.17,1.15916
LmjF.18,0.938494
LmjF.19,0.91213
LmjF.20,1.40783
LmjF.21,0.945452
LmjF.22,0.919244
LmjF.23,1.3705
LmjF.24,0.934335
LmjF.25,0.872227
LmjF.26,0.84407
LmjF.27,0.765535
LmjF.28,0.892511
LmjF.29,0.877807
LmjF.30,1.12821
LmjF.31,2.16148
LmjF.32,0.924272
LmjF.33,0.775568
LmjF.34,0.780326
LmjF.35,0.849297
LmjF.36,0.877814
```

Fig. 2 Chromosome copy number estimation table

3.3.9 Generate a CCNV Plot

The CCNV plot will be generated in "R".

```
#Install R library "ggplot2"
>install.packages("ggplot2")
#Load the R library "ggplot2"
>library(ggplot2)
#Open the file CCNV-Table.csv
>Table <- read.table(file = "<CCNV-Table.csv>", sep
= ",")
#Generate the plot using ggplot
> ggplot(Table, aes(x=V1, y=V2)) + geom_bar(stat =
"identity") + labs(x = "Chromosomes", y = "Copies
per haploid genome") +
theme(axis.text.x=element_text(angle=90)) +
ggsave("<CCNV_plot1.png>", width = 10, height = 7
plot= last_plot(), dpi = 300)
Example command:
>library(ggplot2)
>Table <- read.table(file =
"SRR6369659_CCNV_Table.csv", sep = ",")
>ggplot(Table, aes(x=V1, y=V2)) + geom_bar(stat
= "identity") + labs(x = "Chromosomes", y =
"Copies per haploid genome") +
theme(axis.text.x=element_text(angle=90)) +
ggsave("SRR6369659_CCNV-CCNV_plot1.png", width
= 10, height = 7 plot= last_plot(), dpi = 300)
```

In this plot, each bar corresponds to one chromosome, where the height of the bar corresponds to the number of copies of that chromosome for each haploid genome (Fig. 3).

3.3.10 Restrict the Somy Estimation to Coding Sequences (CDSs) Regions

The chromosomal somy estimation could be restricted to CDSs. This optional step removes the biased impact of non-CDS repetitive regions, as well as the bias from "N" blocks in the reference genome, improving the somy estimation. Hence, this step is highly recommended in repetitive, "N" rich genome drafts. The CCNV estimation using only CDSs-filtered regions can be seen in Fig. 3b.

```
#Filter only the CDS regions from the GFF file and
sort the annotation file
>awk '$3=="CDS"{print $0}' <File.gff> | sort -k1,1 -
k4,4n > <File-CDS.gff>
Example command:
>awk '$3=="CDS"{print $0}' TriTrypDB
-41_LmajorFriedlin.gff | sort -k1,1 -k4,4n >
TriTrypDB-41_LmajorFriedlin-CDS.gff

#Merge redundant CDS regions using bedtools merge
> bedtools merge -i <File-CDS.gff> > <File-CDS-
```

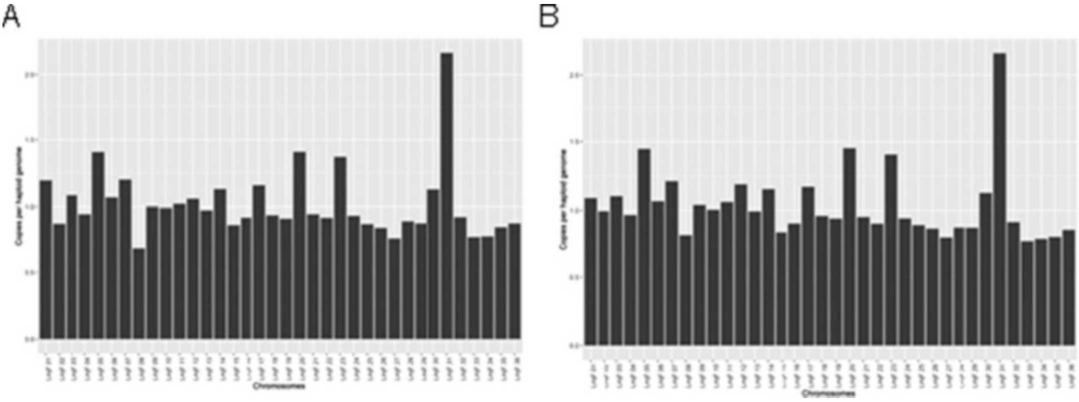


Fig. 3 Chromosome copy number estimation by RDC. In this image, each chromosome is represented by a black bar, where the height of the bar represents the chromosomal copy number per haploid genome. (**a**) Chromosome copy number estimation using the whole chromosomal sequence. (**b**) Chromosome copy number estimation using only coding sequences

```
merged.gff>
Example Command:
>bedtools merge -i TriTrypDB-41_LmajorFriedlin-
CDS.gff > TriTrypDB-41_LmajorFriedlin-CDS-merged.gff

#Generate a Genome-coverage.bed compatible with
bedtools intersect:
> awk '{print $1"\t"$2"\t"$2"\t"$3}' <Genome-
coverage.bed> > <Genome-coverage.bed-form>
Example command:
> awk '{print $1"\t"$2"\t"$2"\t"$3}'
SRR6369659-coverage.bed > SRR6369659-
coverage.bed-form

#Filter the Genome-coverage.bed-form file using the
CDS coordinates with
the program bedtools intersect
> bedtools intersect -a <Genome-coverage.bed-for> -b
<File-CDS-merged.gff> | awk '{print $1"\t"$2"\t"$4}'
> <GFF-filtered-CDS.bed>
Example command:
> bedtools intersect -a SRR6369659-
coverage.bed-form -b TriTrypDB-
41_LmajorFriedlin-CDS-merged.gff | awk '{print
$1"\t"$2"\t"$4}' > SRR6369659-GFF-filtered-
CDS.bed

# Count the number of nucleotides in CDSs in the
filtered reference genome:
> wc -l <GFF-filtered-CDS.bed> | awk '{print $1}' >
```

```
<CDS-Genome-positions>
Example command:
> wc -l SRR6369659-GFF-filtered-CDS.bed | awk
’{print $1}’ > SRR6369659-CDS-Genome-positions

#Estimate the sum of the RDC of all CDSs genomic
positions:
cat <GFF-filtered-CDS.bed> | awk ’{print totalsize+=
$3}’ | tail -n1 > <CDS-Genome-RDC>
Example command:
awk ’{print totalsize += $3}’ SRR6369659-GFF-
filtered-CDS.bed | tail -n1 > SRR6369659-CDS-
Genome-RDC

#Estimate the genome coverage based on CDS regions:
Divide the value in “CDS-Genome-RDC” for the value
in “SRR6369659-CDS-Genome-positions”
Example command:
Divide the value in SRR6369659-CDS-Genome-RDC
(1131730860) for the value in SRR6369659-CDS-
Genome-positions (16255649), obtaining ~69.62,
which represent the genome coverage based on
CDSs regions

#Generate a list with all chromosome names:
> cat <GFF-filtered-CDS.bed> | awk ’{print $1}’ |
sort -u > <Chromosome-IDs>
Example command:
> cat SRR6369659-GFF-filtered-CDS.bed | awk
’{print $1}’ | sort -u > SRR6369659-GFF-
filtered-Chromosome-IDs

# Generate a table containing the normalized
chromosome copy number based on the ration between
the mean genome RDC and the mean genome RDC.
> for i in $(cat <Chromosome-IDs>); do awk -v i="$i"
’$1 == i {print $0}’ <GFF-filtered-CDS.bed> | awk -v
i="$i" ’ {sum += $3} {NR} END {print
i","(sum/NR)/<genome_RDC>}’ >> <CCNV-Table.csv>;
done
Example command:
> for i in $(cat SRR6369659-GFF-filtered-
Chromosome-IDs); do awk -v i="$i" ’$1 == i
{print $0}’ SRR6369659-GFF-filtered-CDS.bed |
awk -v i="$i" ’ {sum += $3} {NR} END {print
i","(sum/NR)/69}’ >> SRR6369659_CCNV_Table-
CDS.csv; done
```

```
#Generate the plot in R using ggplot (Fig. 3b)
> library(ggplot2)
> Table <- read.table(file = "<CCNV-Table-CDS.csv>",
sep = ",")
> ggplot(Table, aes(x=V1, y=V2)) + geom_bar(stat =
"identity") + labs(x = "Chromosomes", y = "Copies
per haploid genome") +
theme(axis.text.x=element_text(angle=90)) +
ggsave("<CCNV_plot-CDS.png>", width = 10, height =
7, plot= last_plot(), dpi = 300)
Example command:
> library(ggplot2)
> Table <- read.table(file =
"SRR6369659_CCNV_Table-CDS.csv", sep = ",")
> ggplot(Table, aes(x=V1, y=V2)) +
geom_bar(stat = "identity") + labs(x =
"Chromosomes", y = "Copies per haploid genome")
+ theme(axis.text.x=element_text(angle=90)) +
ggsave("SRR6369659-CCNV_plot-CDS.png", width =
10, height = 7, plot= last_plot(), dpi = 300)
```

3.3.11 Chromosomal Somy Estimations Based on Allele Frequency Analysis

The chromosomal somy estimations based on allele frequency analysis can be performed using any SNP caller program. In this example we will use GATK. This estimation is based on the proportion of reads that correspond to each allele in heterozygous positions. Initially, generate an index and sequence dictionary required to run GATK and mark PCR duplicated reads.

```
#Format files to run GATK:
Index fasta reference genome with BWA:
>bwa index <reference.genome.fasta>
Index fasta reference using samtools:
Example command:
>bwa index TriTrypDB-
41_LmajorFriedlin_Genome.fasta

>samtools faidx <reference.genome.fasta>
Create a sequence dictionary using Picard
Example command:
>samtools faidx TriTrypDB-
41_LmajorFriedlin_Genome.fasta

>java -jar CreateSequenceDictionary.jar R=
<reference.genome.fasta> O= <reference.genome.dict>
Assigns all the reads in a file to a single new
read-group
>java -jar picard-tools-
1.119/AddOrReplaceReadGroups.jar
```

```
SORT_ORDER=coordinate I=<file.bam> O=$<file-RG.bam>
RGID=<ID> RGSM=<ID> RGPU=unit11 RGLB=Lib1
RGPL=ILLUMINA
Example command:
java -jar picard-tools-
1.119/AddOrReplaceReadGroups.jar
SORT_ORDER=coordinate I=SRR6369659-Mapped-file-
q30-sorted.bam o=SRR6369659-Mapped-file-q30-
sorted-RG.bam RGID=ID RGSM=ID RGPU=unit11
RGLB=Lib1 RGPL=ILLUMINA

Mark duplicated reads originated by PCR:
>java -jar picard-tools-1.119/MarkDuplicates.jar
I=<file-RG.bam> O=<file-RG.DD.bam> M=<file-
MarkDuplicates.metrics.tmp.txt>
Example command:
java -jar picard-tools-1.119/MarkDuplicates.jar
I=SRR6369659-Mapped-file-q30-sorted-RG.bam
O=SRR6369659-Mapped-file-q30-sorted-RG-DD.bam
M=MarkDuplicates.metrics.tmp.txt
```

Next, generate the SNP call file in VCF format using GATK.

```
#Index the bam file with the de-duplicated reads:
>samtools index <file-RG.DD.bam>
Example command:
samtools index SRR6369659-Mapped-file-q30-
sorted-RG-DD.bam

#Re-align reads
>java -jar GenomeAnalysisTK.jar -T
RealignerTargetCreator -R <reference.genome.fasta> -
I <file-RG.DD.bam> -o <file-intervals.tmp.list>
Example command:
java -jar GenomeAnalysisTK.jar -T
RealignerTargetCreator -R TriTrypDB-
41_LmajorFriedlin_Genome.fasta -I SRR6369659-
Mapped-file-q30-sorted-RG-DD.bam -o SRR6369659-
intervals.tmp.list

>java -jar GenomeAnalysisTK.jar -T IndelRealigner -R
<reference.genome.fasta> -I <file-RG.DD.bam> -
targetIntervals <file-intervals.tmp.list> -o <file-
realign.tmp.bam>
Example command:
java -jar GenomeAnalysisTK.jar -T
IndelRealigner -R TriTrypDB-
41_LmajorFriedlin_Genome.fasta -I SRR6369659-
```

```
Mapped-file-q30-sorted-RG-DD.bam -
targetIntervals SRR6369659-intervals.tmp.list -
o SRR6369659-Mapped-file-q30-sorted-RG-DD-
SNPs.bam

#Call SNPS:
java -jar GenomeAnalysisTK.jar -T HaplotypeCaller -R
<reference.genome.fasta> -I <file-realign.tmp.bam> -
ploidy 2 -stand_call_conf 30 -stand_emit_conf 10 -o
<file-SNPs-indels.vcf>
Example command:
java -jar GenomeAnalysisTK.jar -T
HaplotypeCaller -R TriTrypDB-
41_LmajorFriedlin_Genome.fasta -I SRR6369659-
Mapped-file-q30-sorted-RG-DD-SNPs.bam -ploidy 2
-stand_call_conf 30 -stand_emit_conf 10 -o
SRR6369659-SNPs-Indels.vcf

#Filter vcf file (optional):
>vcflib/bin/vcffilter -f "DP > 9" -f "QUAL >10"
<file-SNPs-indels.vcf> > <file-SNPs-filter.tmp.vcf>
Example command:
/media/data/joao/VCFlib.dir/vcflib/bin/vcffilte
r -f "DP > 9" -f "QUAL >10" SRR6369659-SNPs-
Indels.vcf > SRR6369659-SNPs-Indels-filt.vcf

#Recover only SNP calls
>java -jar GenomeAnalysisTK.jar -T SelectVariants -R
<reference.genome.fasta> -V <file-SNPs-
filter.tmp.vcf> -selectType SNP -o <file-only-
SNPs.vcf>
Example command:
java -jar
/home/bioinfo/bin/GenomeAnalysisTK.jar -T
SelectVariants -R ../../TriTrypDB-
41_LmajorFriedlin_Genome.fasta -V SRR6369659-
SNPs-Indels-filt.vcf -selectType SNP -o
SRR6369659-SNPs-filt.vcf
```

After obtaining the VCF file containing all the SNP positions, filter only the heterozygous positions and generate the allele frequency ratio for the whole genome as well as for each chromosome (Figs. 4 and 5):

```
#Convert the VCF output to a table
>java -jar GenomeAnalysisTK.jar -R
<reference.genome.fasta> -T VariantsToTable -V
<file-only-SNPs.vcf> -F CHROM -F POS -F REF -F ALT -
```

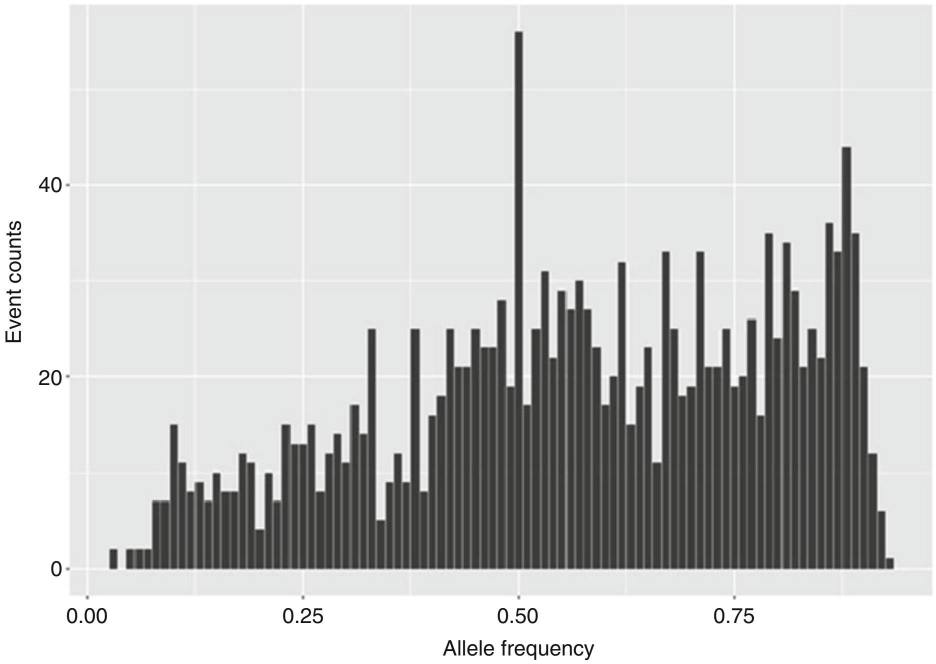


Fig. 4 Genome ploidy estimation based on allele frequency ratios. In this image, the *X* axis denotes the allele frequency ratio of heterozygous positions, while the *Y* axis represents the number of heterozygous positions that present a given allele frequency ratio. The higher peak in 0.5 shows that the majority of the heterozygous positions present a similar RDC in both allele variants, supporting an overall diploid genome

```
GF GT -GF AD -o <file-only-SNPs.table>
Example command:
java -jar GenomeAnalysisTK.jar -R TriTrypDB-
41_LmajorFriedlin_Genome.fasta -T
VariantsToTable -V SRR6369659-SNPs-filt.vcf -F
CHROM -F POS -F REF -F ALT -GF GT -GF AD -o
SRR6369659-SNPs-filt.table

#Remove the header from the table file:
>tail -n +2 <file-only-SNPs.table> > <file-only-
SNPs.NH.table>
Example command:
tail -n +2 SRR6369659-SNPs-filt.table >
SRR6369659-SNPs-filt-NH.table

#Generate the overall genome ploidy based on allele
frequency ratio across the genome with two decimal
values. Obs: The number of decimal values can be
changed to 1 or 3 by respectively replacing the
"%0.2f\n" for "%0.1f\n" or "%0.3f\n",
>awk '{print $6}' <file-only-SNPs.NH.table> | sed
s'/,/\t/' | awk '{if ($1>=5 && $2>=5) print $0}' |
```

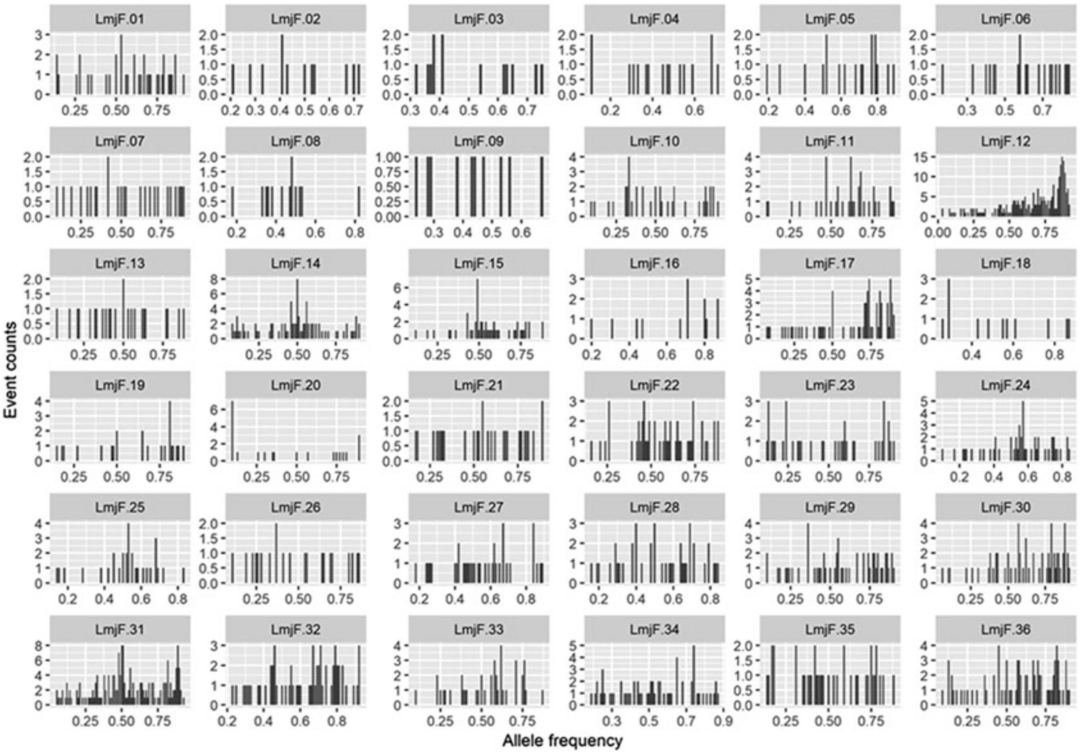


Fig. 5 Chromosomal somy estimation based on allele frequency ratios. In this image, each box corresponds to a chromosome. In each box, the *X* axis denotes the allele frequency ratio of heterozygous positions, while the *Y* axis represents the number of heterozygous positions that present a given allele frequency ratio. As the number of heterozygous SNPs in *Leishmania* is low, the chromosome somy estimation based on allele frequency is compromised

```
awk ’$3 = $1+$2{printf "%0.2f\n",$1/$3}’ | sort |
uniq -c | awk ’{print $2"\t"$1}’ > <whole-genome-
allele-frequency>
Example command:
awk ’{print $6}’ SRR6369659-SNPs-filt-NH.table
| sed s’/,/\t/’ | awk ’{if ($1>=5 && $2>=5)
print $0}’ | awk ’$3 = $1+$2{printf
"%0.2f\n",$1/$3}’ | sort | uniq -c | awk
’{print $2"\t"$1}’ > SRR6369659-whole-genome-
allele-frequency

#Plot the whole genome allele frequency using “R”
(Fig. 4).
library(ggplot2)
Table <- read.table(file = "whole-genome-allele-
frequency", sep = "\t")
ggplot(Table, aes(x=V1, y=V2)) + geom_bar(stat =
"identity") + labs(x = "Allele frequency", y =
"Event counts") + ggsave("Genome-Allele-
frequency.png", width = 10, height = 7,
plot= last_plot(), dpi = 300)
Example command:
library(ggplot2)
Table <- read.table(file = "SRR6369659-whole-
genome-allele-frequency", sep = "\t")
ggplot(Table, aes(x=V1, y=V2)) + geom_bar(stat
= "identity") + labs(x = "Allele frequency", y
= "Event counts") + ggsave("SRR6369659-whole-
genome-allele-frequency.png", width = 10,
height = 7, plot= last_plot(), dpi = 300)

#Generate a chromosomal somy estimation based on the
allele frequency of each chromosome (Fig. 5).
cat <file-only-SNPs.NH.table> | sed s'/,/\t/' | awk
'$8=$6+$7{if ($6>=5 && $7>=5) printf
$1",""%0.2f\n",$6/$8}' | sort | uniq -c | sed
s'/,/\t/' | awk '{print $1"\t"$3"\t"$2}' > <Alelle-
Frequency-chromosomes>
Example command:
cat SRR6369659-SNPs-filt-NH.table | sed
s'/,/\t/' | awk '$8=$6+$7{if ($6>=5 && $7>=5)
printf $1",""%0.2f\n",$6/$8}' | sort | uniq -c
| sed s'/,/\t/' | awk '{print $1"\t"$3"\t"$2}'
> Alelle-Frequency-chromosomes

#Plot the chromosomal somy estimation based on the
allele frequency of each chromosome using "R"
library(ggplot2)
Table <- read.table(file = "<Alelle-Frequency-
chromosomes>", sep = "\t")
ggplot(Table, aes(x=V2, y=V1)) + facet_wrap(.~V3,
scales = "free") + geom_bar(stat = "identity") +
labs(x = "Allele frequency", y = "Event counts") +
ggsave("chromosome-allele-frequency.png", width =
10, height = 7, plot= last_plot(), dpi = 300)

Example command:
library(ggplot2)
Table <- read.table(file = "Alelle-Frequency-
chromosomes", sep = "\t")
ggplot(Table, aes(x=V2, y=V1)) +
facet_wrap(.~V3, scales = "free") +
geom_bar(stat = "identity") + labs(x = "Allele
frequency", y = "Event counts") +
ggsave("SRR6369659-chromosome-allele-
frequency.png", width = 10, height = 7, plot=
last_plot(), dpi = 300)
```

As the low number of heterozygous SNPs compromises the chromosomal somy estimation by allele frequency in *Leishmania*, we have also performed this analysis in a highly heterozygous *T. cruzi* clone, obtained from the Y strain (Fig. 6). *T. cruzi* Y clone 2 genomic read library can be obtained from NCBI SRA (ID: SRX3453758), while the *T. cruzi* CL Brener Esmeraldo-like haplotype reference genome can be obtained from TriTypDB (https://tritrypdb.org/common/downloads/release-42/TcruziCLBrenerEsmeraldo-like/fasta/data/TriTrypDB-42_TcruziCLBrenerEsmeraldo-like_Genome.fasta). The commands used to generate the following plot are the same as the one used to generate the *Leishmania* estimations.

4 Notes

MFA-seq Analysis of Replication Dynamics

1. As a fixing agent, methanol is used for *T. brucei* procyclic cells and *Leishmania* promastigote cells. Insufficient *T. brucei* bloodstream form cells could be retrieved after fixing with methanol; instead, fixation with 1% formaldehyde (methanol-free) allows the retrieval of a sufficient number of cells.
2. The number of cells needed is variable depending upon how many cells are needed for downstream applications; for this protocol we recommend preparing 3×10^8 cells for each sorting session. If other numbers of cells are to be used, adjust the volumes used in the protocol accordingly.
3. Washing in 1× PBS in the presence of EDTA prevents cell aggregation.
4. A final concentration of 2.5×10^7 cells/ml is ideal for the following steps in the protocol; if denser, the cells start to form aggregates.
5. After formaldehyde fixation the cells seem to become more fragile, so the speed used in the centrifugation was reduced from 1000 × *g* to 700 × *g*.
6. Settings such as flow and event threshold rates will vary depending on the machine; this will impact the hours that will be needed to obtain the desirable number of cells (e.g., a machine that sorts a max. of 3000 events per second vs one that does it at 10,000 events per second).
7. The number of cells needed depends on the application. The minimum number of cells we have used is 1×10^6 cells in each of the S phase subpopulations (early and late).
8. The type of tubes will depend on the machine used.

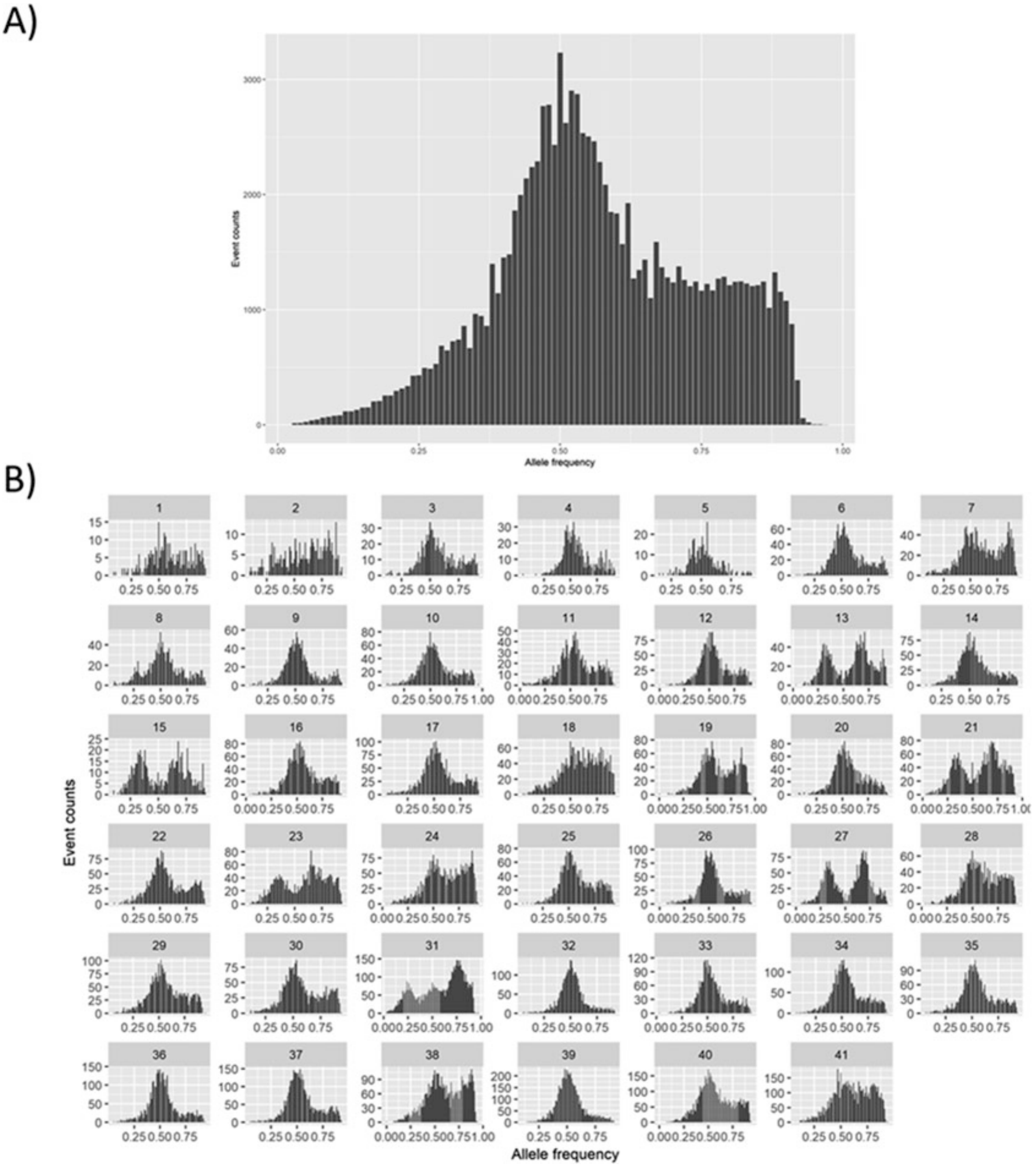


Fig. 6 *T. cruzi* Y clone 2 ploidy and somy estimation based on allele frequency ratios. In this image, the *X* axis denotes the allele frequency ratio of heterozygous positions, while the *Y* axis represents the number of heterozygous positions that present a given allele frequency ratio. (**a**) Whole-genome ploidy. The peak in 0.5 shows that the majority of the heterozygous positions present a similar RDC in both allele variants, supporting an overall diploid genome. (**b**) Chromosomal somy estimation, where each box corresponds to a chromosome. The majority of the chromosomes present a peak in 0.50, suggesting disomy. However, chromosomes 13, 15, 21, 23, and 27 presented peaks in ~0.33 and ~0.66 suggesting trisomy, while the chromosome 31 presented peaks in 0.25, 0.50, and 0.75 suggesting tetrasomy

9. The SDS in the lysis buffer will precipitate at 4 °C, giving it a 'blurry' look; this is ok during collection and remember to mix it thoroughly after sorting and before lysis.
10. The temperature and incubation period allow for the reversal of the formaldehyde crosslinking.
11. If several sorting sessions are needed to obtain the desired number of cells, store the lysate at −20 °C and extract gDNA from the samples of the different sorting sessions at the same time (e.g., if three sessions were run to obtain enough cells in S phase, pool the extracts from the three sessions and extract the gDNA using a single column).

Localization of DNA Damage by γH2A ChIP-seq

12. Do not exceed fixation time as cells will be difficult to lyse.
13. It may be helpful to observe cells before lysis for comparison. If cells are successfully lysed numerous nuclei should appear as black dots and cell bodies will appear "ghostly". It is important that the Dounce homogenizer is tight fitting for optimal lysis.
14. Shearing has been optimized for *T. brucei* chromatin. Incubation for 5 min results in fragments between 200 and 1500 bp in length. Shearing time may require optimization for other genomes.
15. Quantification has been carried out using a Qubit. If using a spectrometer for quantification, the DNA must be cleaned up by phenol–chloroform prior to quantification. Column purification is not recommended as protein may clog the column.
16. When performing multiple samples to compare, that is, treatment vs control, use the same quantity of chromatin (ng) per IP reaction.
17. Depending on the target, more chromatin may need to be added to the ChIP reaction in order to achieve a high enough concentration of DNA in the eluted sample for downstream analysis. If vastly more chromatin is required, the number of cells used for chromatin extraction can be increased. However, the volumes of Fixation Solution, Glycine, 1× PBS, Lysis Buffer (as well as the supplements PIC and PMSF), Digestion Buffer (plus PIC and PMSF), Enzymatic Shearing Cocktail and 0.5 M EDTA must all be scaled accordingly. For example, double all reagents when the protocol is started with 2×10^8 parasites.
18. When washing the samples, it is important to prevent the beads from drying; therefore, perform wash steps one tube at a time.
19. When large amounts of chromatin have been used, and so the samples contain higher concentrations of proteins, this step can be extended.

20. If little DNA is available for analysis, input DNA can be diluted further and eluted DNA can also be diluted. However, these must be corrected for accurately when calculating Input percentage.
21. This is especially important when analyzing regions of highly similar sequences such as *T. brucei* VSG genes.
22. When targeting γH2A, it is likely that some nonspecific binding to the unphosphorylated H2A histone occurs. Therefore, it may be useful to compare ChIP signal enrichment between control and treatment samples (e.g., wild-type and mutant). This can also be done via Deeptools using the bigwigCompare tool.

Next Generation Sequencing to Examine Genome Variation

23. The examples shown here were executed on an Ubuntu 10.10 server with a 2.8-GHz CPU. For 32 million reads, at least 4 GB of RAM is required. More RAM may need to if more deeply sequenced samples are used.
24. Although not required for all ploidy estimation methodologies, the use of a GFF file can increase accuracy when performing CCNV analysis in complex genomes. This optional step removes the biased impact of non-CDS repetitive regions, as well as the bias from "N" blocks in the reference genome, improving the somy estimation.
25. The Illumina platform usually yields reads with the deepest and highest base quality. However, reads from other platforms, such as 454 and Ion torrent, can also be used.
26. Long-reads, such as generated by the PacBio Sequel/RSII and Oxford Nanopore platforms, are also suitable for ploidy estimations. The .bam file can be generated using BlasR (https://github.com/PacificBiosciences/blasr), and the following steps described in this chapter can be used.
27. It is preferable to use FASTQ format, rather than fasta format, because it encodes not only the sequence information itself, but also the base quality information. This is essential for accurate detection of SNPs and other types of structural alterations.
28. Coverage of at least 20× is desirable. Using lower genome coverages can compromise sensitivity and accuracy of any structural variation analysis.
29. All software used here are compatible with the Linux operational systems, such as Fedora and Ubuntu. Some of them are also compatible with macOS; however, all of the examples were performed on Linux.

30. A guide on how to work with bash in command line can be seen in: https://www.cs.wcupa.edu/rkline/linux/bash-basics.html. A beginner guide can be seen here: https://www.lifewire.com/guide-to-bash-part-1-hello-world-2202041 and here https://www.bash.academy/.

31. In our example, we are using a paired-end read library from the *L. major* genome. Thus, when downloading it using fastq-dump, the use of "--split-files" flag is mandatory. By doing so, reads from each pair will be split into two different files. When using single-end read libraries, the aforementioned flag is not needed.

32. It is not advisable to use read libraries with a mean read quality lower than 20. Such libraries will reduce the SNP call precision and compromise the accuracy of downstream analysis.

33. We suggest that read libraries in which more than 50% of the read were dropped should not be used in CCNV analysis. As the proportion of excluded reads is not necessarily similar to all chromosomes, this could lead to detection of false variation in chromosome copy number.

Acknowledgement

This work was supported by the BBSRC [BB/K006495/1, BB/M028909/1, BB/N016165/1], the European Commission [RECREPEMLE], and the Wellcome Trust [206815, 083485, 089172]. The Wellcome Centre for Integrative Parasitology is supported by core funding from the Wellcome Trust [104111].

References

1. Roth DB (2014) V(D)J recombination: mechanism, errors, and fidelity. Microbiol Spectr 2. https://doi.org/10.1128/microbiolspec.MDNA3-0041-2014
2. Hwang JK, Alt FW, Yeap LS (2015) Related mechanisms of antibody somatic hypermutation and class switch recombination. Microbiol Spectr 3:MDNA3-0037-2014
3. Lee CS, Haber JE (2015) Mating-type gene switching in Saccharomyces cerevisiae. Microbiol Spectr 3:MDNA3-0013-2014
4. Yao MC, Chao JL, Cheng CY (2014) Programmed genome rearrangements in tetrahymena. Microbiol Spectr 2. https://doi.org/10.1128/microbiolspec.MDNA3-0012-2014
5. Yerlici VT, Landweber LF (2014) Programmed genome rearrangements in the ciliate oxytricha. Microbiol Spectr 2. https://doi.org/10.1128/microbiolspec.MDNA3-0025-2014
6. Betermier M, Duharcourt S (2014) Programmed rearrangement in ciliates: paramecium. Microbiol Spectr 2. https://doi.org/10.1128/microbiolspec.MDNA3-0035-2014
7. McCulloch R, Morrison LJ, Hall JP (2015) DNA recombination strategies during antigenic variation in the African trypanosome. Microbiol Spectr 3:MDNA3-0016-2014
8. Dumetz F, Imamura H, Sanders M, Seblova V, Myskova J, Pescher P, Vanaerschot M, Meehan CJ, Cuypers B, De Muylder G et al (2017) Modulation of aneuploidy in Leishmania donovani during adaptation to different in vitro and in vivo environments and its impact on gene expression. MBio 8:e00599-17
9. Lachaud L, Bourgeois N, Kuk N, Morelle C, Crobu L, Merlin G, Bastien P, Pages M, Sterkers Y (2014) Constitutive mosaic aneuploidy

is a unique genetic feature widespread in the Leishmania genus. Microbes Infect 16:61–66

10. Tiengwe C, Marcello L, Farr H, Dickens N, Kelly S, Swiderski M, Vaughan D, Gull K, Barry JD, Bell SD et al (2012) Genome-wide analysis reveals extensive functional interaction between DNA replication initiation and transcription in the genome of *Trypanosoma brucei*. Cell Rep 2:185–197
11. Marques CA, Dickens NJ, Paape D, Campbell SJ, McCulloch R (2015) Genome-wide mapping reveals single-origin chromosome replication in Leishmania, a eukaryotic microbe. Genome Biol 16:230
12. Marques CA, McCulloch R (2018) Conservation and variation in strategies for DNA replication of kinetoplastid nuclear genomes. Curr Genomics 19:98–109
13. Briggs E, Crouch K, Lemgruber L, Lapsley C, McCulloch R (2018) Ribonuclease H1-targeted R-loops in surface antigen gene expression sites can direct trypanosome immune evasion. PLoS Genet 14:e1007729
14. Glover L, Horn D (2012) Trypanosomal histone gammaH2A and the DNA damage response. Mol Biochem Parasitol 183:78–83
15. Stortz JA, Serafim TD, Alsford S, Wilkes J, Fernandez-Cortes F, Hamilton G, Briggs E, Lemgruber L, Horn D, Mottram JC et al (2017) Genome-wide and protein kinase-focused RNAi screens reveal conserved and novel damage response pathways in Trypanosoma brucei. PLoS Pathog 13:e1006477
16. Damasceno JD, Obonaga R, Silva GLA, Reis-Cunha JL, Duncan SM, Bartholomeu DC, Mottram JC, McCulloch R, Tosi LRO (2018) Conditional genome engineering reveals canonical and divergent roles for the Hus1 component of the 9-1-1 complex in the maintenance of the plastic genome of Leishmania. Nucleic Acids Res 46:11835–11846
17. Reis-Cunha JL, Rodrigues-Luiz GF, Valdivia HO, Baptista RP, Mendes TAO, de Morais GL, Guedes R, Macedo AM, Bern C, Gilman RH et al (2015) Chromosomal copy number variation reveals differential levels of genomic plasticity in distinct Trypanosoma cruzi strains. BMC Genomics 16:499
18. Almeida LV, Coqueiro-Dos-Santos A, Rodriguez-Luiz GF, McCulloch R, Bartholomeu DC, Reis-Cunha JL (2018) Chromosomal copy number variation analysis by next generation sequencing confirms ploidy stability in Trypanosoma brucei subspecies. Microb Genom 4. https://doi.org/10.1099/mgen.0.000223
19. Rogers MB, Hilley JD, Dickens NJ, Wilkes J, Bates PA, Depledge DP, Harris D, Her Y, Herzyk P, Imamura H et al (2011) Chromosome and gene copy number variation allow major structural change between species and strains of Leishmania. Genome Res 21:2129–2142
20. Laffitte MC, Leprohon P, Hainse M, Legare D, Masson JY, Ouellette M (2016) Chromosomal translocations in the parasite Leishmania by a MRE11/RAD50-independent microhomology-mediated end joining mechanism. PLoS Genet 12:e1006117
21. Azuara V (2006) Profiling of DNA replication timing in unsynchronized cell populations. Nat Protoc 1:2171–2177
22. Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen YJ, Chen Z et al (2005) Genome sequencing in microfabricated high-density picolitre reactors. Nature 437:376–380
23. Quail MA, Smith M, Coupland P, Otto TD, Harris SR, Connor TR, Bertoni A, Swerdlow HP, Gu Y (2012) A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. BMC Genomics 13:341
24. Diaz A, Park K, Lim DA, Song JS (2012) Normalization, bias correction, and peak calling for ChIP-seq. Stat Appl Genet Mol Biol 11:9