The genome of a tardigrade -
Horizontal gene transfer or bacterial contamination?
Supplemental material

Felix Bemm[1], Clemens Leonard Weiß[1], Jörg Schultz[2,3], Frank Förster[2,3,*]

[1] Max-Planck-Institute for Developmental Biology, Department Molecular Biology, 72076 Tübingen, Germany

[2] Center for Computational and Theoretical Biology, Campus Nord, University of Würzburg, 97074 Würzburg, Germany

[3] Department for Bioinformatics, Biozentrum, University of Würzburg, 97074 Würzburg, Germany

[*] To whom correspondence should be addressed: frank.foerster@uni-wuerzburg.de

1 Figures

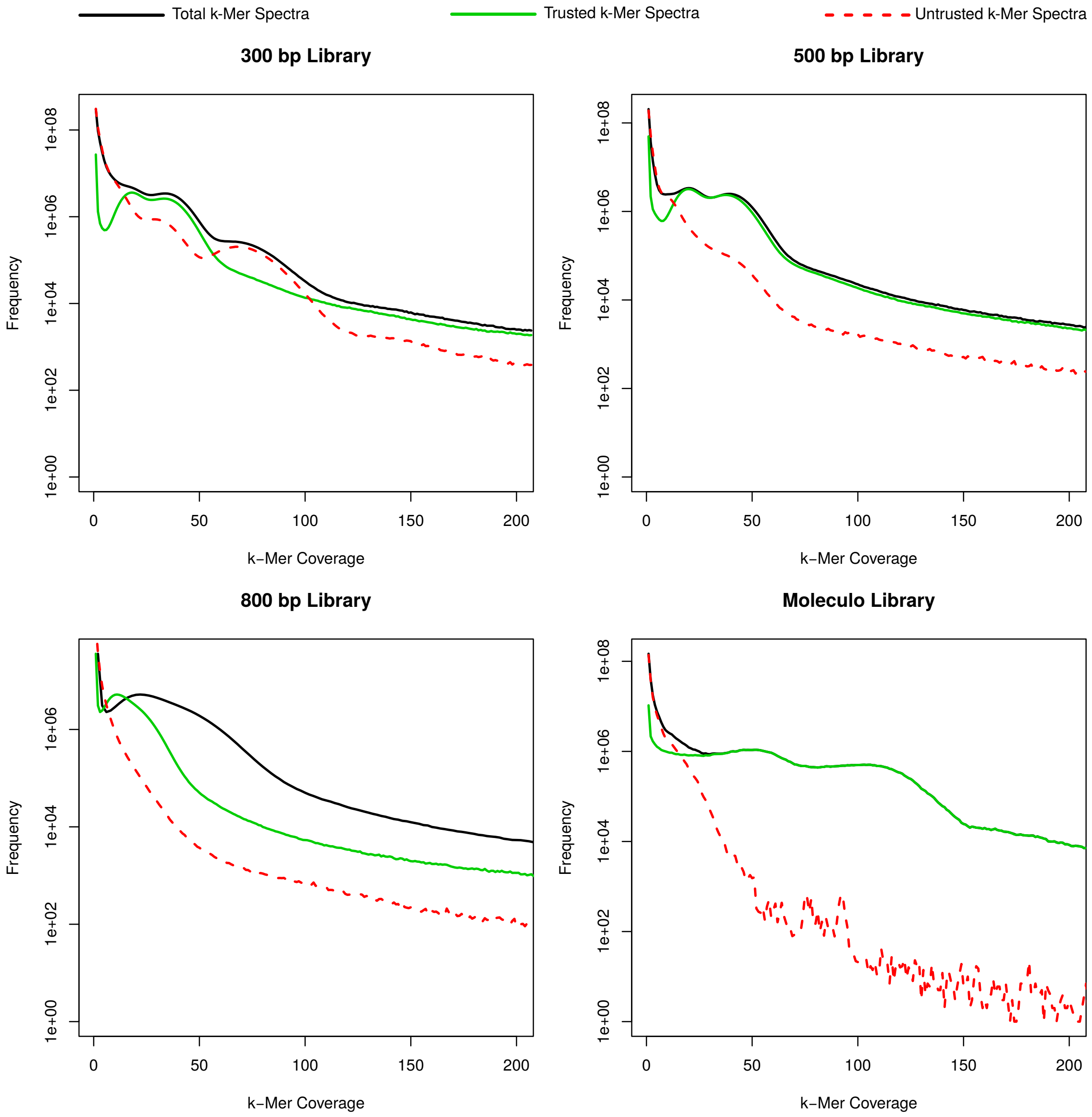

Figure 1: The plots depict the kmer distribution for each library before (black line) and after classification into 'trusted' (green line) and 'untrusted' kmers (red line).

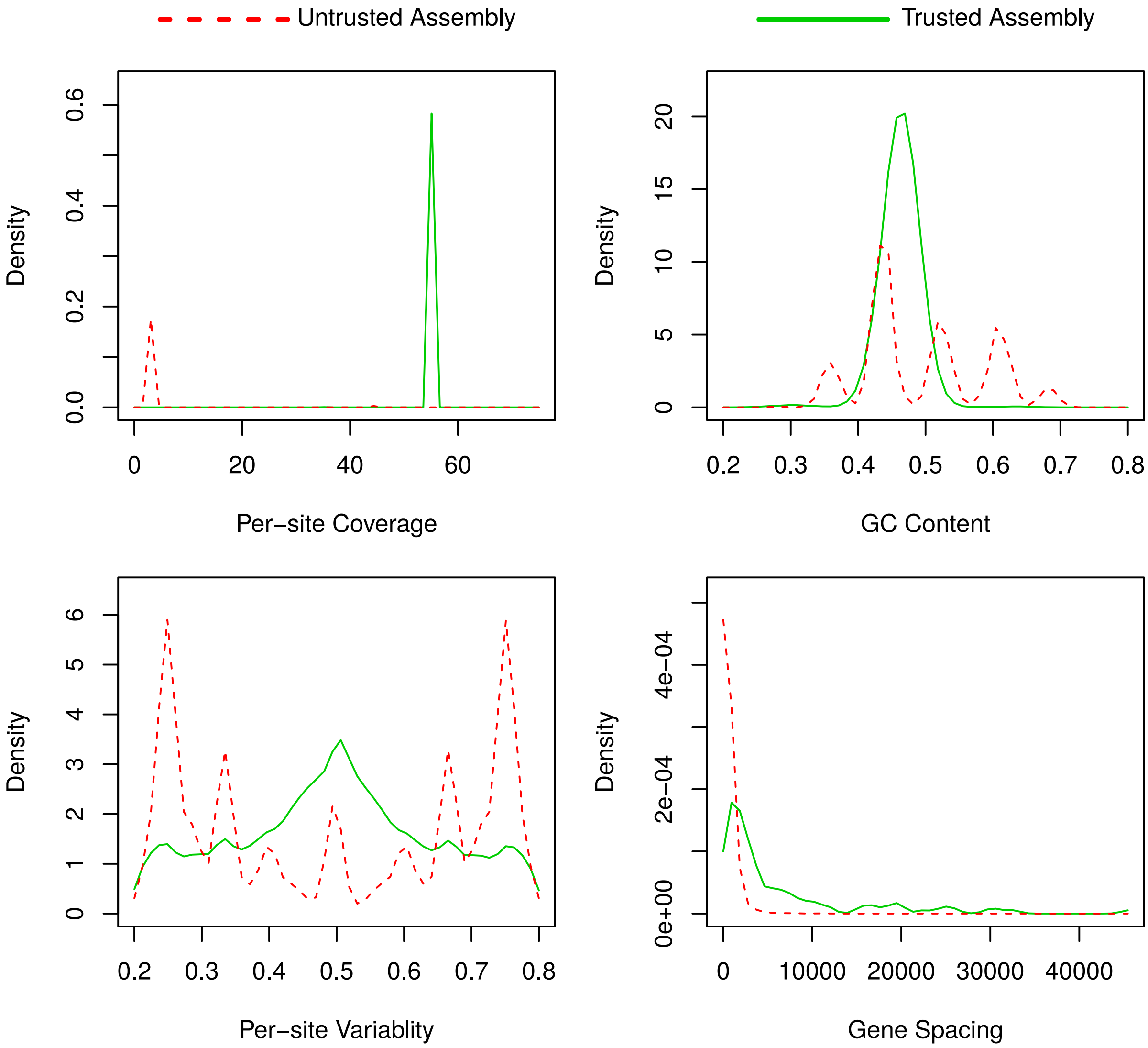

Figure 2: A) Per-site coverage of trusted and untrusted assembly based on mappings of Moleculo reads. Contigs from the untrusted assembly generally don't share the coverage of the trusted, most likley nuclear, genome. B) GC content of trusted and untrusted assembly estimated using sliding window approach. The untrusted assembly contains multiple peaks pointing towards contig subpopulations with different GC content. C) Per-site variability of trusted und untrusted assembly which can serve as ploidy proxy. The untrusted variability spectrum seems distored and contains a multiude of different peaks while the trusted assembly shows a typical diploid spectrum. D) Length distribution of intergenetic regions. Intragenetic regions are significantly larger in the trusted assembly than in the trusted.

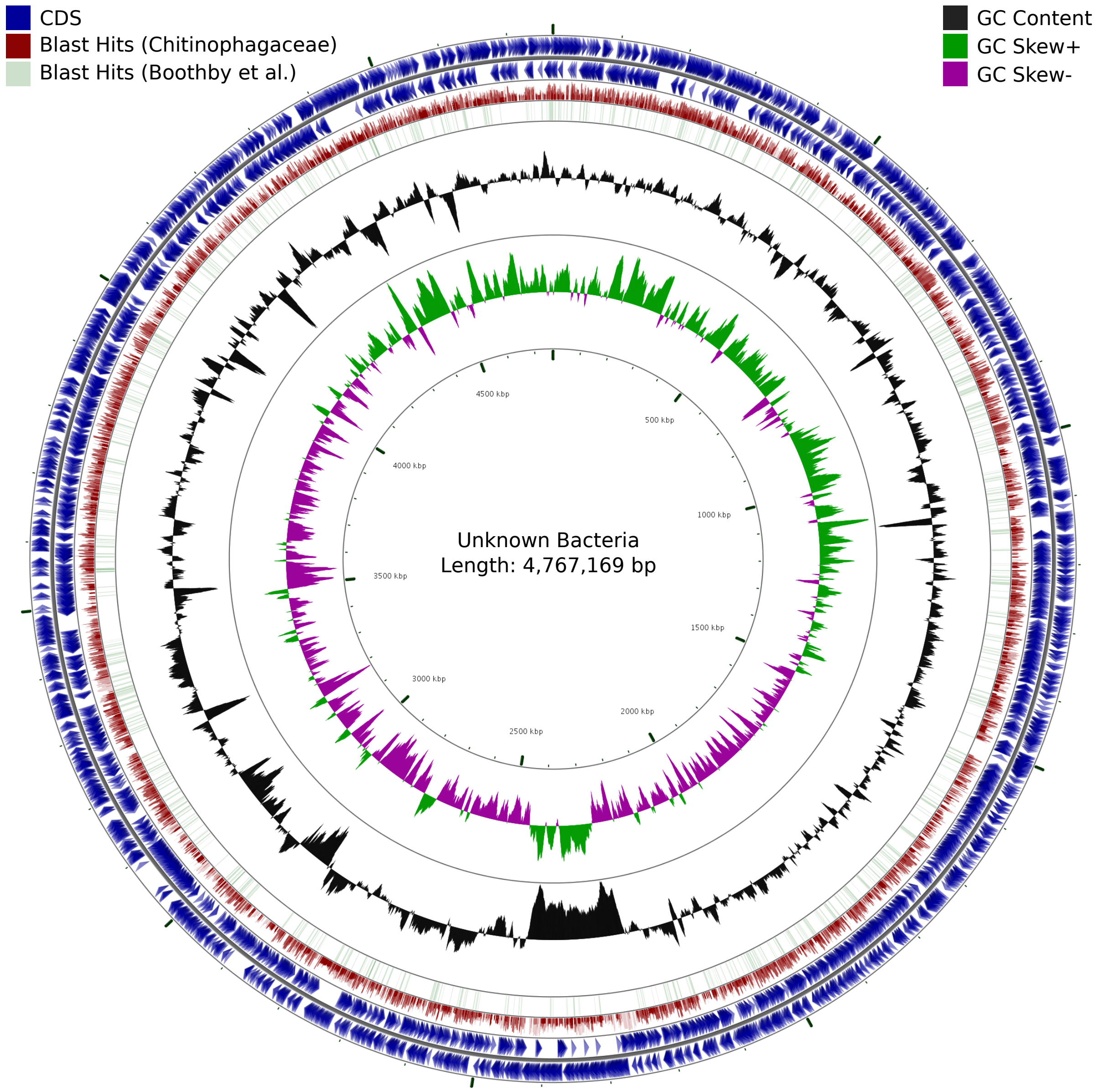

Figure 3: Circular map of an unknown bacterial genome probably belonging to the Chitinophagaceae drawn with CGView. Tracks 1 and 2 (blue) indicate GeneMark-S annotated genes on forward and reverse strand. Track 3 (red) visualizes regions of homology to a set of 30,844 Chitinophagaceae proteins downloaded from UniProtKB. Track 4 (green) shows homology between GeneMark-S predicted proteins and the published protein set of Boothby et al. [3]

2 Methods

GitHub repository

All script files are available from our GitHub repository (https://github.com/greatfireball/hypsibius_genome_revised/).

Data set

We used the data set provided by Boothby et al. [3] and downloaded the data from http://weatherby.genetics.utah.edu/seq_transf/. A complete list of the used input files are given in table 1.

Programs

Table 2: List of all programs including the version numbers and references to publications or websites used for the data processing and analysis

Programname	Version	Reference
Allpath-LG	v 50378	Gnerre et al. [5] and Ribeiro et al. [17]
BEDTools	v 2.20.1	Quinlan and Hall [16]
bioperl	v 1.69.1	Stajich et al. [18]
bowtie2	v 2.2.2	Langmead and Salzberg [8]
bwa	v 0.7.10	Li and Durbin [12, 11]
CGView	v 1.0	Grin and Linke [6]
Falcon	v 0.4.0	https://github.com/PacificBiosciences/falcon
Genemark-S	v 4.3.2	Besemer, Lomsadze, and Borodovsky [1]
Genemark-ET	v 4.29	Lomsadze, Burns, and Borodovsky [14]
Jellyfish	v 2.2.4	Marçais and Kingsford [15]
Perl	v 5.14.2	https://www.perl.org/
samtools	v 1.1	Li et al. [13] and Li [9, 10]
skewer	v 0.1.124	Jiang et al. [7]
‘sm’ R package	v 2.2-5.4	Bowman and Azzalini [4]
Trimmomatic	v 0.3.5	Bolger, Lohse, and Usadel [2]

Trimming of the input data

Short reads were trimmed with skewer.

```
skewer -m pe -q 30 -Q 30 -l 60 -t 64 \
   HD_gen.il_L[358]*00_P1.fastq HD_gen.il_L[358]*00_P2.fastq
```

Long reads were trimmed with Trimmomatic.

```
java -jar trimmomatic-0.35.jar SE -phred33 HD_gen.mo_L[12345]*.fastq \
   HD_gen.mo_L[12345]*.trimmed.fastq \
```

Table 1: Data set used for our analysis including checksums for compressed and decompressed file content.

Filename and location	Modification time	Size in Bytes	MD5 check sum	MD5 check sum decompressed
tg.genome.fsa.gz	2015-11-25T01:34:44Z	72,215,266	b8bd39390ef35dd43d1cda1ca6944d5a	77be374d28b91232c0810cc4d3cd37b9
tg.default.maker.proteins.final.fasta.gz	2015-12-02T23:43:44Z	12,359,873	2de12e5d28d6dba121973db2071565d9	1ad17cfa9e6c26e552fa8048c6ee90af
short_reads/TG-300-SIPE_1_sequence.txt	2015-11-30T21:48:51Z	11,526,955,725	c16b5442c9893b6feaa3aa81a39eefcd	c16b5442c9893b6feaa3aa81a39eefcd
short_reads/TG-300-SIPE_2_sequence.txt.gz	2015-11-30T21:52:41Z	3,920,224,257	3bea43d66d71926fb620966d281598c6	bc8423d4fe4275863e0809445ffd21ce
short_reads/TG-500-SIPE_1_sequence.txt.gz	2015-12-01T05:32:05Z	2,738,243,219	da8b15d388961938584343f8926f7b24	eee7363557ccb1fb0fa75ebe55ae7ee5
short_reads/TG-500-SIPE_2_sequence.txt.gz	2015-12-01T05:35:15Z	2,805,269,168	aa8c2c345484b9464d272e0993d6968b	325d74bbafd9b6019609e2fd33eca260
short_reads/TG-800-SIPE_1_sequence.txt.gz	2015-12-01T05:36:55Z	2,155,735,304	6e9cce1a27000ae2b4f87181a976df92	a85568ef53979c367870eee6390f2ced
short_reads/TG-800-SIPE_2_sequence.txt.gz	2015-12-01T05:37:46Z	2,058,207,374	ccf097cf4f13bb5cbc5a8e002250093d	4a4cc02c2f289d59c300810fb621eb28
moleculo_reads/LR6000049-DNA_A01-LRAAD-01_LongRead.fastq.gz	2015-11-30T17:50:17Z	825,877,986	86e75544f2d6ef5185bae419bbd2a4b2	bace73ed4750b33fc144e56c155454ab
moleculo_reads/LR6000049-DNA_A01-LRAAD-02_LongRead.fastq.gz	2015-11-30T17:51:34Z	835,283,315	4dea3e39a7a25059a6ebbd5588e845b2	cb83c39f9a385f0b4fd1e507cfe40ff1
moleculo_reads/LR6000049-DNA_A01-LRAAD-03_LongRead.fastq.gz	2015-11-30T17:52:51Z	847,867,943	16276b6ef8dea90721eb67ac21d616e6	51d4ce37668684b4aa25e061fb95b4ef
moleculo_reads/LR6000049-DNA_A01-LRAAD-04_LongRead.fastq.gz	2015-11-30T17:56:08Z	859,746,540	3364040445c7377c9323f82d98a2258c	dbe06ec4248199f416bb1d02ff1e65f5
moleculo_reads/LR6000049-DNA_A01-LRAAD-05_LongRead.fastq.gz	2015-11-30T17:56:51Z	854,266,597	7995559df803ef0de0250f1bfac71f1a	98d30f3ceb813d9f53c6df2ed1fa2239

```
   ILLUMINACLIP:adapter.fa:2:30:10 LEADING:30 TRAILING:30 MINLEN:250
```

Estimation of the genomes size

The genome size was estimated by the standalone error correction pipeline of Allpaths-LG.

```
./scripts/genome_size_estimation.sh
```

Counting and Filtering bases on kmers

The kmers of all libraries where counted using the software jellyfish [15]:

```
./scripts/count_kmers.sh
```

The resulting kmer hashes need to be dumped and converted to a hash utilized later during the filtering step. This step and the following required > 200 GB of memory and was performed by the perl script `prepare_filter_fastq_by_valid_kmers.pl`.

```
./scripts/dump_kmers.sh
```

The generated hash was used to filter individual libraries. Therefore, we have written the perl script `filter_fastq_by_valid_kmers_reduced.pl`.

```
./scripts/filter_input_data.sh
```

The filtered data sets are classified as ‘trusted’ or ‘untrusted’ based on the ‘trusted’ kmer content. Reads with at least 95 % ‘trusted’ kmers content are called ‘trusted’ while reads below that threshold are classified as ‘untrusted’.

```
./scripts/extract_classified_sequences.sh
```

Long Read Assembly

Trusted and untrusted Moleculo reads were assembled with Falcon.

```
fc_run.py trusted.falcon.cfg
fc_run.py untrusted.falcon.cfg
```

See configuration files for parameter details.

Assembly Annotation

The trusted assembly was annotated with GeneMark-ES.

```
gmes_petap.pl --sequence HD_gen.trusted.fasta \
   --ES --cores 64
```

The untrusted assembly was annotated with GeneMark-S

```
gmsn.pl --fnn --faa --species HD --gm \
   --name HD HD_gen.unsupported.fasta
```

The largest untrusted sequence was visualized using the CGView Server.

Assembly Comparison

Trusted and untrusted assemblies were compared using GC content, mapping coverage, per-site variability and gene spacing.

GC content The GC content was determined for all contigs ≥ 1 kbp using a sliding window of 1 kbp and a stepsize of 100 bp by the perl script `sliding_window_gc.pl`.

```
mkdir cg
cd cg

for i in ../assemblies/HD*.fasta
do
   ../scripts/sliding_window_gc/sliding_window_gc.pl
      --in "$i" \
      --min-length 1000
      > $(basename "$i").sliding_gc.tsv
done
```

Mapping Coverage The mapping coverage was determined by remapping of the short or longreads onto the assembled contig. For the short read libraries, we used `bowtie2` as mapper. Long read libraries were mapped by `bwa`. The per-base coverage was determined by bedtools.

```
./scripts/determine_mapping_coverage.sh
```

Per-site Variability The per-site variability was calculated by counting bases covering each site of the two assemblies. Each base that occurred at a given site with a minimum frequency of 0.2 was taken into account and a histogram of all these base frequencies was created.

Gene Spacing Length of the intragenetic regions were directly extracted from the GeneMark-S/ES annotation files.

```
gtf2genespacing.pl --gtf HD_gen.supported.gtf
gtf2genespacing.pl --gtf HD_gen.unsupported.gtf
```

All resulting data sets were compared, tested and visualized using the GNU R package 'sm'.

References

[1] J. Besemer, A. Lomsadze, and M. Borodovsky. “GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions.” eng. In: *Nucleic Acids Res* 29.12 (June 2001), pp. 2607–2618.

[2] Anthony M. Bolger, Marc Lohse, and Bjoern Usadel. “Trimmomatic: a flexible trimmer for Illumina sequence data.” eng. In: *Bioinformatics* 30.15 (Aug. 2014), pp. 2114–2120. DOI: 10.1093/bioinformatics/btu170. URL: http://dx.doi.org/10.1093/bioinformatics/btu170.

[3] Thomas C. Boothby et al. “Evidence for extensive horizontal gene transfer from the draft genome of a tardigrade”. In: *Proceedings of the National Academy of Sciences* (2015). DOI: 10.1073/pnas.1510461112. eprint: http://www.pnas.org/content/early/2015/11/18/1510461112.full.pdf. URL: http://www.pnas.org/content/early/2015/11/18/1510461112.abstract.

[4] A. W. Bowman and A. Azzalini. *R package sm: nonparametric smoothing methods (version 2.2-5.4)*. University of Glasgow, UK and Università di Padova, Italia, 2014. URL: %5Curl%7Bhttp://www.stats.gla.ac.uk/~adrian/sm%7D%20%5Curl%7Bhttp://azzalini.stat.unipd.it/Book_sm%7D.

[5] Sante Gnerre et al. “High-quality draft assemblies of mammalian genomes from massively parallel sequence data.” eng. In: *Proc Natl Acad Sci U S A* 108.4 (Jan. 2011), pp. 1513–1518. DOI: 10.1073/pnas.1017351108. URL: http://dx.doi.org/10.1073/pnas.1017351108.

[6] Iwan Grin and Dirk Linke. “GCView: the genomic context viewer for protein homology searches.” eng. In: *Nucleic Acids Res* 39.Web Server issue (July 2011), W353–W356. DOI: 10.1093/nar/gkr364. URL: http://dx.doi.org/10.1093/nar/gkr364.

[7] Hongshan Jiang et al. “Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads.” eng. In: *BMC Bioinformatics* 15 (2014), p. 182. DOI: 10.1186/1471-2105-15-182. URL: http://dx.doi.org/10.1186/1471-2105-15-182.

[8] Ben Langmead and Steven L. Salzberg. “Fast gapped-read alignment with Bowtie 2.” eng. In: *Nat Methods* 9.4 (Apr. 2012), pp. 357–359. DOI: 10.1038/nmeth.1923. URL: http://dx.doi.org/10.1038/nmeth.1923.

[9] Heng Li. “A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data.” eng. In: *Bioinformatics* 27.21 (Nov. 2011), pp. 2987–2993. DOI: 10.1093/bioinformatics/btr509. URL: http://dx.doi.org/10.1093/bioinformatics/btr509.

[10] Heng Li. “Improving SNP discovery by base alignment quality.” eng. In: *Bioinformatics* 27.8 (Apr. 2011), pp. 1157–1158. DOI: 10.1093/bioinformatics/btr076. URL: http://dx.doi.org/10.1093/bioinformatics/btr076.

[11] Heng Li and Richard Durbin. “Fast and accurate long-read alignment with Burrows-Wheeler transform.” eng. In: *Bioinformatics* 26.5 (Mar. 2010), pp. 589–595. DOI: 10.1093/bioinformatics/btp698. URL: http://dx.doi.org/10.1093/bioinformatics/btp698.

[12] Heng Li and Richard Durbin. “Fast and accurate short read alignment with Burrows-Wheeler transform.” eng. In: *Bioinformatics* 25.14 (July 2009), pp. 1754–1760. DOI: 10.1093/bioinformatics/btp324. URL: http://dx.doi.org/10.1093/bioinformatics/btp324.

[13] Heng Li et al. “The Sequence Alignment/Map format and SAMtools.” eng. In: *Bioinformatics* 25.16 (Aug. 2009), pp. 2078–2079.

[14] Alexandre Lomsadze, Paul D. Burns, and Mark Borodovsky. “Integration of mapped RNA-Seq reads into automatic training of eukaryotic gene finding algorithm.” eng. In: *Nucleic Acids Res* 42.15 (Sept. 2014), e119. DOI: 10.1093/nar/gku557. URL: http://dx.doi.org/10.1093/nar/gku557.

[15] Guillaume Marçais and Carl Kingsford. “A fast, lock-free approach for efficient parallel counting of occurrences of k-mers”. In: *Bioinformatics* 27.6 (2011), pp. 764–770. DOI: 10.1093/bioinformatics/btr011. eprint: http://bioinformatics.oxfordjournals.org/content/27/6/764.full.pdf+html. URL: http://bioinformatics.oxfordjournals.org/content/27/6/764.abstract.

[16] Aaron R. Quinlan and Ira M. Hall. “BEDTools: a flexible suite of utilities for comparing genomic features.” eng. In: *Bioinformatics* 26.6 (Mar. 2010), pp. 841–842. DOI: 10.1093/bioinformatics/btq033. URL: http://dx.doi.org/10.1093/bioinformatics/btq033.

[17] Filipe J. Ribeiro et al. “Finished bacterial genomes from shotgun sequence data.” eng. In: *Genome Res* 22.11 (Nov. 2012), pp. 2270–2277. DOI: 10.1101/gr.141515.112. URL: http://dx.doi.org/10.1101/gr.141515.112.

[18] Jason E. Stajich et al. “The Bioperl toolkit: Perl modules for the life sciences.” eng. In: *Genome Res* 12.10 (Oct. 2002), pp. 1611–1618. DOI: 10.1101/gr.361602. URL: http://dx.doi.org/10.1101/gr.361602.