**Supplementary materials**

**De novo assembly and annotation application in discovery of novel pathogenic strain of E. Coli causing Hemolytic Uremic Syndrome (HUS)**

1. **Downloading and inspecting the data**The data were obtained from the NCBI Sequence Read Archive:

wget <https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292678sub_S1_L001_R1_001.fastq.gz>

wget <https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292678sub_S1_L001_R2_001.fastq.gz>

wget <https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292862_S2_L001_R1_001.fastq.gz>

wget <https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292862_S2_L001_R2_001.fastq.gz>

wget <https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292770_S1_L001_R1_001.fastq.gz>

wget <https://d28rh4a8wq0iu5.cloudfront.net/bioinfo/SRR292770_S1_L001_R2_001.fastq.gz>

First, the data were unzipped and the quality of the reads was examined by the FASTQC program.

gunzip <reads>

fastqc -o . ./<reads>

Results of the fastqc analysis are represented in table 1 and on figures 1-3 (only for problematic features).

Table 1 - Fastqc statistics.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Measure | File type | Encoding | Total sequences | Sequences flagged as poor quality | Sequence length | %GC | Problematic features |
| Filename |  |  |  |  |  |  |  |
| SRR292678sub\_S1\_L001\_R1\_001.fastq | Conventional base calls | Sanger / Illumina 1.9 | 5499346 | 0 | 90 | 49 | Per sequence GC content |
| SRR292678sub\_S1\_L001\_R2\_001.fastq | Conventional base calls | Sanger / Illumina 1.9 | 5499346 | 0 | 90 | 49 | Per sequence GC content |
| SRR292770\_S1\_L001\_R1\_001.fastq | Conventional base calls | Sanger / Illumina 1.9 | 5102041 | 0 | 49 | 50 | - |
| SRR292770\_S1\_L001\_R2\_001.fastq | Conventional base calls | Sanger / Illumina 1.9 | 5102041 | 0 | 49 | 49 | - |
| SRR292862\_S2\_L001\_R1\_001.fastq | Conventional base calls | Sanger / Illumina 1.9 | 5102041 | 0 | 49 | 50 | Per sequence GC content |
| SRR292862\_S2\_L001\_R2\_001.fastq | Conventional base calls | Sanger / Illumina 1.9 | 5102041 | 0 | 49 | 49 | - |

Figure 1 - Per sequence GC content of SRR292678sub\_S1\_L001\_R1\_001.fastq

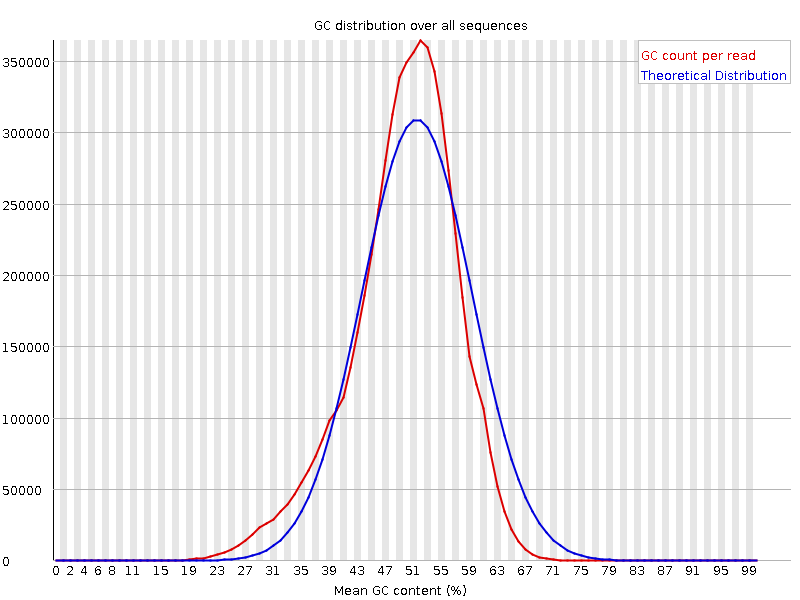


Figure 2 - Per sequence GC content of SRR292678sub\_S1\_L001\_R2\_001.fastq

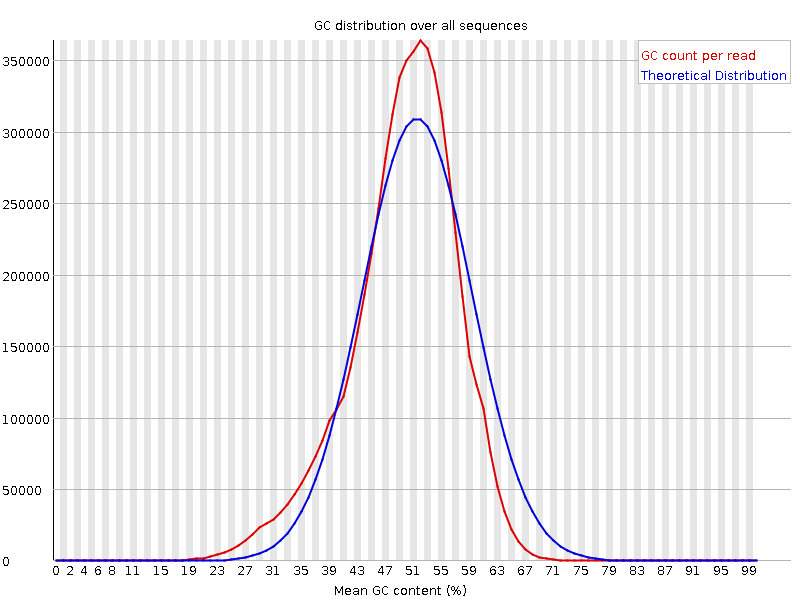
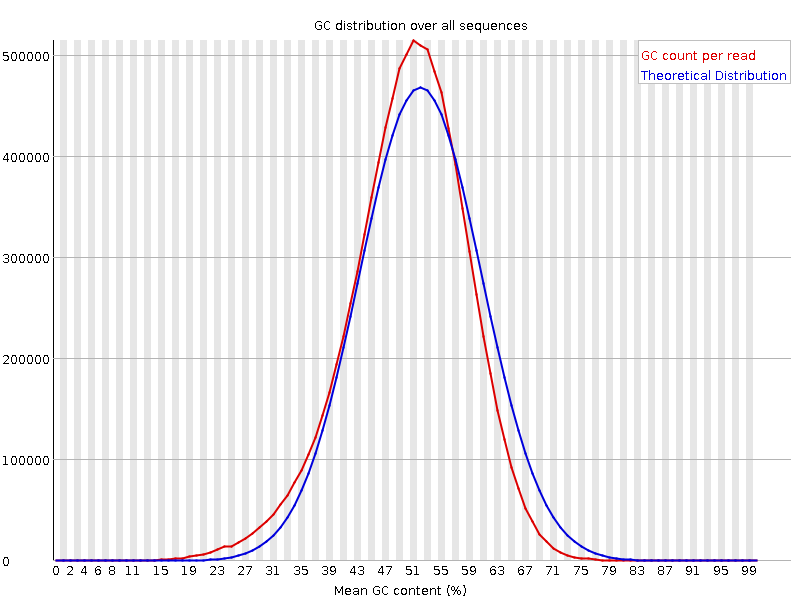


Figure 3 - Per sequence GC content of SRR292862\_S2\_L001\_R1\_001.fastq



1. **Genome assembling and quality assessment**Genome assembling carried out using SPAdes.

Installing SPAdes:

sudo apt install spades

Running in test mode:

/usr/lib/spades/bin/spades.py --test

SPAdes was installed correctly.

Assembling with paired reads only:

/usr/lib/spades/bin/spades.py --pe1-1 SRR292678\_R1.fastq --pe1-2 SRR292678\_R2.fastq -o spades\_output\_1

Quality assessment was performed with the online version of QUAST.

**Statistics without reference contigs**

# contigs 205

# contigs (>= 0 bp) 519

# contigs (>= 1000 bp) 151

# contigs (>= 5000 bp) 84

# contigs (>= 10000 bp) 69

# contigs (>= 25000 bp) 51

# contigs (>= 50000 bp) 32

Largest contig 300784

Total length 5243761

Total length (>= 0 bp) 5315282

Total length (>= 1000 bp) 5204854

Total length (>= 5000 bp) 5039755

Total length (>= 10000 bp) 4936078

Total length (>= 25000 bp) 4640187

Total length (>= 50000 bp) 3936175

N50 105346

N75 51460

L50 15

L75 32

GC (%) 50.53

Mismatches

# N's 0

# N's per 100 kbp 0

**Statistics without reference scaffolds**

# contigs 213

# contigs (>= 0 bp) 501

# contigs (>= 1000 bp) 147

# contigs (>= 5000 bp) 83

# contigs (>= 10000 bp) 67

# contigs (>= 25000 bp) 51

# contigs (>= 50000 bp) 33

Largest contig 300784

Total length 5252701

Total length (>= 0 bp) 5316905

Total length (>= 1000 bp) 5205427

Total length (>= 5000 bp) 5046669

Total length (>= 10000 bp) 4936501

Total length (>= 25000 bp) 4686215

Total length (>= 50000 bp) 4027080

N50 105346

N75 55266

L50 15

L75 32

GC (%) 50.51

Mismatches

# N's 1693

# N's per 100 kbp 32.23

Assembling with mate-paired reads (all three libraries):

/usr/lib/spades/bin/spades.py --pe1-1 SRR292678\_R1.fastq --pe1-2 SRR292678\_R2.fastq --mp1-1 SRR292770\_R1.fastq --mp1-2 SRR292770\_R2.fastq --mp2-1 SRR292862\_R1.fastq --mp2-2 SRR292862\_R2.fastq -o spades\_output\_2

Result: failed with Error code: -9. Not enough RAM. The results were downloaded from:

<https://yadi.sk/d/XHCbTIrvxzN5Y>

Quality assessment for assembling with three libraries:

**Statistics without reference contigs**

# contigs 105

# contigs (>= 0 bp) 369

# contigs (>= 1000 bp) 79

# contigs (>= 5000 bp) 33

# contigs (>= 10000 bp) 30

# contigs (>= 25000 bp) 26

# contigs (>= 50000 bp) 22

Largest contig 698474

Total length 5350156

Total length (>= 0 bp) 5403327

Total length (>= 1000 bp) 5331230

Total length (>= 5000 bp) 5202939

Total length (>= 10000 bp) 5183802

Total length (>= 25000 bp) 5133691

Total length (>= 50000 bp) 4975501

N50 335515

N75 143558

L50 6

L75 13

GC (%) 50.59

Mismatches

# N's 0

# N's per 100 kbp 0

**Statistics without reference scaffolds**

# contigs 90

# contigs (>= 0 bp) 327

# contigs (>= 1000 bp) 54

# contigs (>= 5000 bp) 16

# contigs (>= 10000 bp) 13

# contigs (>= 25000 bp) 10

# contigs (>= 50000 bp) 10

Largest contig 2815616

Total length 5391554

Total length (>= 0 bp) 5437160

Total length (>= 1000 bp) 5365719

Total length (>= 5000 bp) 5258076

Total length (>= 10000 bp) 5238939

Total length (>= 25000 bp) 5200270

Total length (>= 50000 bp) 5200270

N50 2815616

N75 391920

L50 1

L75 4

GC (%) 50.57

Mismatches

# N's 33833

# N's per 100 kbp 627.52

1. **Genome annotation**Genome annotation was performed using Prokka program for rapid bacterial genome annotation.

Firstly, Prokka was downloaded:

sudo apt install prokka

Running prokka on scaffolds.fasta file raised a message:

[15:06:21] This is prokka 1.14.5

[15:06:21] Written by Torsten Seemann <torsten.seemann@gmail.com>

[15:06:21] Homepage is https://github.com/tseemann/prokka

[15:06:21] Local time is Thu Dec 3 15:06:21 2020

[15:06:21] You are jenny

[15:06:21] Operating system is linux

[15:06:21] You have BioPerl 1.7.7

Argument "1.7.7" isn't numeric in numeric lt (<) at /usr/bin/prokka line 259.

[15:06:21] System has 4 cores.

[15:06:21] Option --cpu asked for 8 cores, but system only has 4

[15:06:21] Will use maximum of 4 cores.

[15:06:21] Annotating as >>> Bacteria <<<

[15:06:21] The sequence databases have not been indexed. Please run 'prokka --setupdb' first.

We faced a problem with BioPerl and discussions on github showed that Prokka doesn’t work with version of Perl more than 5.22. We have Perl 5.30. Installing of Perl 5.22 failed, so we used annotation results from:

<http://public.dobzhanskycenter.ru/mrayko/Week3/PROKKA.zip>

Table 2 - Summary for the annotation.

|  |  |
| --- | --- |
| Feature | Number |
| tRNAs | 80 |
| rRNAs | 0 |
| CRISPRs | 1 |
| CDS | 5064 |
| Unique gene codes | 2923 |

1. **Finding the closest relative to E. coli X**The closest relative for our strain was found using 16s ribosomal RNA. We run barrnap tool to find this gene in our assembly:

sudo apt install barrnap

barrnap scaffolds.fasta

We got .gff3 as a result:

NODE\_184\_length\_223\_cov\_0.720238\_ID\_565088 barrnap:0.9 rRNA 95 205 5.6e-18 - . Name=5S\_rRNA;product=5S ribosomal RNA

NODE\_1\_length\_2815616\_cov\_74.3819\_ID\_564387 barrnap:0.9 rRNA 322892 323002 2.2e-11 - . Name=5S\_rRNA;product=5S ribosomal RNA

NODE\_1\_length\_2815616\_cov\_74.3819\_ID\_564387 barrnap:0.9 rRNA 326359 327896 0 - . Name=16S\_rRNA;product=16S ribosomal RNA

NODE\_1\_length\_2815616\_cov\_74.3819\_ID\_564387 barrnap:0.9 rRNA 592162 592272 6.1e-11 - . Name=5S\_rRNA;product=5S ribosomal RNA

NODE\_1\_length\_2815616\_cov\_74.3819\_ID\_564387 barrnap:0.9 rRNA 592407 592517 2.2e-11 - . Name=5S\_rRNA;product=5S ribosomal RNA

NODE\_1\_length\_2815616\_cov\_74.3819\_ID\_564387 barrnap:0.9 rRNA 595966 597503 0 - . Name=16S\_rRNA;product=16S ribosomal RNA

NODE\_1\_length\_2815616\_cov\_74.3819\_ID\_564387 barrnap:0.9 rRNA 2500844 2500954 2.2e-11 - . Name=5S\_rRNA;product=5S ribosomal RNA

NODE\_1\_length\_2815616\_cov\_74.3819\_ID\_564387 barrnap:0.9 rRNA 2504403 2505940 0 - . Name=16S\_rRNA;product=16S ribosomal RNA

NODE\_5\_length\_236041\_cov\_85.7779\_ID\_563492 barrnap:0.9 rRNA 43835 45372 0 + . Name=16S\_rRNA;product=16S ribosomal RNA

NODE\_5\_length\_236041\_cov\_85.7779\_ID\_563492 barrnap:0.9 rRNA 48983 49084 5.5e-10 + . Name=5S\_rRNA;product=5S ribosomal RNA

NODE\_5\_length\_236041\_cov\_85.7779\_ID\_563492 barrnap:0.9 rRNA 85462 86999 0 + . Name=16S\_rRNA;product=16S ribosomal RNA

NODE\_5\_length\_236041\_cov\_85.7779\_ID\_563492 barrnap:0.9 rRNA 90356 90466 2.2e-11 + . Name=5S\_rRNA;product=5S ribosomal RNA

NODE\_6\_length\_209194\_cov\_80.6603\_ID\_563563 barrnap:0.9 rRNA 111955 113492 0 + . Name=16S\_rRNA;product=16S ribosomal RNA

NODE\_6\_length\_209194\_cov\_80.6603\_ID\_563563 barrnap:0.9 rRNA 116849 116959 2.2e-11 + . Name=5S\_rRNA;product=5S ribosomal RNA

NODE\_71\_length\_720\_cov\_1.1218\_ID\_565136 barrnap:0.9 rRNA 314 719 9.8e-23 + . Name=16S\_rRNA;product=16S ribosomal RNA (partial);note=aligned only 25 percent of the 16S ribosomal RNA

NODE\_9\_length\_87669\_cov\_86.9393\_ID\_563775 barrnap:0.9 rRNA 84700 84810 2.2e-11 - . Name=5S\_rRNA;product=5S ribosomal RNA

[barrnap] Done.

In the columns 4-5 there are the start and end of a 16s rRNA gene.

Using very ugly code that I wrote we extracted the sequence of 16s rRNA from scaffolds.fasta and find the closest relative in NCBI database:

with open('scaffolds.fasta') as f:

f.readline()

res = ''

line = ''

while '>' not in line:

res += line

line = f.readline().rstrip()

rna\_16s = res[326358:327895]

with open('16s\_rRNA.txt', 'w') as w:

w.write(rna\_16s)

Parameters for NCBI Nucleotide BLAST:

Database: RefSeq Genome database (refseq\_genomes)

Entrez Query: 1900/01/01:2011/01/01[PDAT] - sequences before 2011

Organism: Escherchia coli (taxid:562)

We found Escherichia coli 55989, complete genome with 100% identity, sequence ID [NC\_011748.1](https://www.ncbi.nlm.nih.gov/nucleotide/NC_011748.1?report=genbank&log$=nuclalign&blast_rank=1&RID=WJ7GBF1J013)

1. **Genome-wide comparison with the reference genome**To visualize the alignment of our assembly on the reference genome we used Mauve. Installation and running the Mauve was a big challenge, so here I show some steps to success (maybe not all the steps are needed):

* get rid of all java versions  
  sudo apt-get purge --auto-remove openjdk\*
* install java version 8  
  sudo apt-get install openjdk-8-jdk
* comment a string in file /etc/java-8-openjdk/accessibility.properties  
  #assistive\_technologies=org.GNOME.Accessibility.AtkWrapper
* download tar.gz file with Mauve and unzip it
* add path to script Mauve  
  JAVA\_CMD=/usr/lib/jvm/java-8-openjdk-amd64/bin/java
* chmod +x ./Mauve.jar
* run

./Mauve

File>Align with progressiveMauve...>Add sequence>reference genome, scaffolds.gbk>Align

Binoculars>Find Product>toxin>stx....>click annotation>GenBank annotation

We found genes of shiga toxins (subunits A and B):

stxA - [3483886-3484845] = 959 bp

stxB - [3483605-3483874] = 269 bp

According to annotation of GenBank stxA is shiga toxin subunit A from Shigella dysenteriae (UniProt Q9FBI2) and stxB is shiga toxin subunit B from Shigella dysenteriae (UniProt Q7BQ98). Those genes were surrounded by bacteriophage genes, so our strain might acquire these shiga-like toxins by horizontal gene transfer.

1. **Antibiotic resistance detection**We used ResFinder to detect the genes responsible for antibiotic resistance.

Table 3 - Antibiotic resistance of E. coli reference strain.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Resistance to: | Gene | | unknown macrolide | mdf(A) | | unknown fluoroquinolone | mdf(A) | | unknown aminoglycoside | mdf(A) | | doxycycline | tet(B) | | unknown phenicol | mdf(A) | | unknown tetracycline | mdf(A) | | unknown rifamycin | mdf(A) | | tetracycline | tet(B) | | minocycline | tet(B) |   Vulnerable to: amikacin, tigecycline, tobramycin, cefepime, chloramphenicol, piperacillin+tazobactam, cefoxitin, ampicillin, ampicillin+clavulanic acid, cefotaxime, ciprofloxacin, colistin, sulfamethoxazole, imipenem, trimethoprim, nalidixic acid, ertapenem, fosfomycin, ceftazidime, temocillin, gentamicin, meropenem, azithromycin. |  |  |  |
| Table 4 - Antibiotic resistance of E. coli X strain. |  |  |  |
| |  |  | | --- | --- | | Resistance to: | Gene | | unknown macrolide | mdf(A) | | unknown fluoroquinolone | mdf(A) | | unknown aminoglycoside | mdf(A) | | doxycycline | tet(B) | | unknown phenicol | mdf(A) | | unknown tetracycline | mdf(A) | | unknown rifamycin | mdf(A) | | tetracycline | tet(B) | | minocycline | tet(B) | | ampicillin | blaCTX-M-15 | | sulfamethoxazole | sul1, sul2 | | trimethoprim | dfrA7 | | ceftriaxone | blaCTX-M-15 | | cephalothin | blaTEM-1B | | piperacillin | blaCTX-M-15 | | amoxicillin | blaCTX-M-15 | | cefepime | blaCTX-M-15 | | ceftazidime | blaCTX-M-15 | | ticarcillin | blaCTX-M-15 | | cefotaxime | blaCTX-M-15 | | aztreonam | blaCTX-M-15 | | streptomycin | aph(6)-Id | |  |  |  |
| 1. **7. Additional - k-mer profile and genome size estimation** 2. Started jellyfish with uncorrected fastq file from pair-read   jellyfish count -m 31 -C -s 10M /Users/evgeniahohlova/Desktop/Bioinf/Practice/project\_3/SRR292678sub\_S1\_L001\_R1\_001.fastq    2. jellyfish histo -o k\_mer\_histo1.histo mer\_counts.jf   2) Then repeat the same action for SPAdes corrected fastq file  jellyfish count -m 31 -C -s 10M /Users/evgeniahohlova/Desktop/Bioinf/Practice/project\_3/spades\_output/corrected/SRR292678sub\_S1\_L001\_R1\_001.00.0\_0.cor.fastq  jellyfish histo -o k\_mer\_histo\_cor.histo mer\_counts\_cor.jf  The algorithm of processing data in RStudio I took from this site: <http://koke.asrc.kanazawa-u.ac.jp/HOWTO/kmer-genomesize.html> |  |  |  |
| Code:  histo <- read.table("/Users/evgeniahohlova/Desktop/Bioinf/Practice/project\_3/k\_mer\_histo1.histo")  plot(histo[5:500,],type="l", xlab = "The nubmer of k-mers", ylab = "Frequency", main = "The k-mer distribution uncorrected data")  points(histo[5:500,])  ## Determine the total number of k-mer analyzed and the peak position  Now, we would calculate the total number of k-mer in the distribution  sum(as.numeric(histo[1:817,1]\*histo[1:817,2]))  total number of k-mer = 329960760  Next, we want to know the peak position. From the graph, we can see it's close to 70.  Thus, we examine the number close to 50 and find the maximum value  histo[50:90,]  peak = 62  the genome size can be estimated as:  sum(as.numeric(histo[1:817,1]\*histo[1:817,2]))/62  genome\_size = 5321948  singleC <- sum(as.numeric(histo[:60,1]\*histo[1:60,2]))/62  plot(1:500,dpois(1:500, 62)\*singleC, type = "l", col=3, lty=2, xlab = "The nubmer of k-mers", ylab = "Frequency", main = "The k-mer distribution uncorrected data")  lines(histo[1:200,],type="l")  # CORRECTED  histo\_cor <- read.table("/Users/evgeniahohlova/Desktop/Bioinf/Practice/project\_3/k\_mer\_histo\_cor.histo")  plot(histo\_cor[5:500,],type="l", xlab = "The nubmer of k-mers", ylab = "Frequency", main = "The k-mer distribution corrected data")  points(histo\_cor[5:500,])  Now, we would calculate the total number of k-mer in the distribution  sum(as.numeric(histo\_cor[1:839,1]\*histo\_cor[1:839,2]))  total number of k-mer = 329934081  Next, we want to know the peak position. From the graph, we can see it's close to 70.  Thus, we examine the number close to 50 and find the maximum value  histo\_cor[50:90,]  peak = 64  the genome size can be estimated as:  sum(as.numeric(histo\_cor[1:817,1]\*histo\_cor[1:817,2]))/64  genome\_size = 5154889  s <- sum(as.numeric(histo\_cor[1:60,1]\*histo\_cor[1:60,2]))/64  plot(1:500, dpois(1:500, 64)\*s, type = "l", col=3, lty=2, xlab = "The nubmer of k-mers", ylab = "Frequency", main = "The k-mer distribution corrected data")  lines(histo\_cor[1:500,],type="l") |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |