DamageProfiler Documentation

Release 1.0

Judith Neukamm, Alexander Peltzer

Apr 26, 2021

Contents:

1 Prerequisites for the installation of DamageProfiler **3**
1.1 Operating System Support . . . 3
1.2 Software Requirements for DamageProfiler . . . 3

2 Installation Instructions for DamageProfiler **5**
2.1 JAR file . . . 5
2.2 Bioconda . . . 5
2.3 Old releases . . . 5

3 General Usage **7**
3.1 How to run . . . 7
3.2 GUI . . . 9
3.3 Log file . . . 10

4 Output Files **11**
4.1 damagePlot.pdf . . . 11
4.2 5pCtoT_freq.txt and 3pGtoA_freq.txt . . . 12
4.3 length_plot.pdf . . . 13
4.4 lgdistribution.txt . . . 13
4.5 edit_distance.pdf . . . 14
4.6 editDistance.txt . . . 16
4.7 misincorporation.txt . . . 16
4.8 5p_freq_misincorporations.txt and 3p_freq_misincorporations.txt . . . 17
4.9 DNA_comp_genome.txt . . . 17
4.10 DNA_composition_sample.txt . . . 18
4.11 dmgprof.json . . . 18
4.12 DamageProfiler.log . . . 18

5 Graphical User Interface **19**
5.1 Load input files . . . 19
5.2 Run configuration . . . 19
5.3 Exploration of results . . . 19
5.4 Metagenomic mapping file . . . 19

6 Runtime Estimation **21**
6.1 How to run . . . 21
6.2 How is the runtime calculated . . . 22

7 Indices and tables 23

sphinx-quickstart on Tue Nov 21 15:42:50 2017. You can adapt this file completely to your liking, but it should at least contain the root *toctree* directive.

This is the main DamageProfiler documentation, where you can find information about the prerequisites, the installation and the usage of this tool.

CHAPTER 1

Prerequisites for the installation of DamageProfiler

1.1 Operating System Support

DamageProfiler has been implemented as a platform-independent tool and can thus be installed and run on Linux, Windows, and MacOS. A Java 11+ platform has to be installed on the workstation used for running the tool.

1.1.1 Linux

It has been successfully tested on Ubuntu 18.04 LTS and 20.04.1 LTS.

1.2 Software Requirements for DamageProfiler

Install a suitable Java 11 (or higher) runtime environment.

CHAPTER 2

Installation Instructions for DamageProfiler

2.1 JAR file

The tool can be downloaded from DamageProfiler's GitHub page. After downloading the JAR file, you can start the application via double click on most operating systems (OSX, Windows, and Linux). If not, please either install Java 11 or higher on your workstation:

```
sudo apt install default-jdk
```

or make the file executable:

```
sudo chmod u+x DamageProfiler-1.0.jar
```

2.2 Bioconda

For easy installation, DamageProfiler is also available as a bioconda package and can be installed with one of the following

On Ubuntu:

```
conda install -c bioconda damageprofiler
conda install -c bioconda/label/cf201901 damageprofiler
```

At the moment, only DamageProfiler version 0.4.9 is available via bioconda.

2.3 Old releases

Old releases can be found at GitHub.

CHAPTER 3

General Usage

DamageProfiler can be used to calculate and visualize damage patterns in ancient DNA. As input, a mapping file (sam, bam, or cram format) is expected. The result is provided in both graphic and text-based representation. DamageProfiler can be used in offline mode, however, idenitfying the species name when running multi-reference mapping files is not possible.

It creates

- damage plots
- fragment length distribution
- edit distances (number of bases that differ between read and reference)
- base frequency table of reference (if reference is specified)
- base frequency table of input file
- table of different base misincorporations and their occurrences

3.1 How to run

```
java -jar DamageProfiler-VERSION.jar [options]
```

Options:

- **-h**

Shows this help page.

- **-version**

Shows the version of DamageProfiler.

- **-i INPUT**

The input sam/bam/cram file.

- **-r REFERENCE**

The reference file (fasta format).

- **-o OUTPUT**

The output folder. Please specify the path to the result folder here. The folder structure will be as following:

- **If neither -s nor -sf are specified:** The results will directly be stored under the output folder specified with -o

  Example:

  -i mapping_file_sample_A.bam -o /path/to/result/directory/mapping_file_sample_A/

  The result files will then be stored in /path/to/result/directory/mapping_file_sample_A/

- **-s is specified:**

  If more than one species is specified, the results are stored in separate folders (per species) under the specified output folder (-o).

  If only one single species is specified, the result will directly be stored under the output folder specified with -o.

  Example:

  -i mapping_file_sample_B.bam -o /home/neukamm/results_damageprofiler/ -s 'NC_002677.1'

  The results will be stored in /home/neukamm/results_damageprofiler/mapping_file_sample_B/NC_002677.1/

  -i mapping_file_sample_B.bam -o /path/to/result/directory/mapping_file_sample_B/ -s 'NC_002677.1,NC_028801.1'

  The results will be stored in /path/to/result/directory/mapping_file_sample_B/NC_002677.1/ and /path/to/result/directory/mapping_file_sample_B/NC_028801.1/ and a summary pdf will be stored in /path/to/result/directory/mapping_file_sample_B/summary.pdf

- **-sf is specified:**

  Species are given as text file, one per line. No quotation marks needed. If more than one species is specified, the results are stored in separate folders (per species) under the specified output folder (-o). If only one single species is specified, the result will directly be stored under the output folder specified with -o.

- **-t THRESHOLD**

Number of bases which are considered for plotting nucleotide misincorporations in the damage plot. Default: 25.

- **-s SPECIES**

Reference sequence name (Reference NAME flag of SAM record). Depending on which database was used for mapping, this is the accession ID of the reference (i.e. NCBI accession ID). Commas within the Reference sequence name are not allowed. The species must be put in quotation marks (e.g. -s 'NC_032001.1|tax|1917232|'), multiple species must be comma separated (e.g. -s 'NC_032001.1|tax|1917232|,NC_031076.1|tax|1838137|').

- **-sf SPECIES FILE**

List with accession IDs of species for which damage profile has to be calculated. This file is a text file, with one species entry per line. Commas within the Reference sequence name are not allowed.

Example:

-i mapping_file_sample_B.bam -o /home/neukamm/results_damageprofiler/ -sf /path/to/species_file.txt

and the content of species_file.txt would look like this:

```
NC_002677.1
NC_028801.1
NC_023501.3
NC_035395.1
```

- **-l LENGTH**

Number of bases which are considered for frequency computations. Default: 100.

- **-title TITLE**

Title used for all plots. Default: input filename.

- **-yaxis_dp_max MAX_VALUE**

Maximal y-axis value that is visualized in the damage plot.

- **-color_c_t COLOR_C_T**

Color for the line representing the C to T misincoporation frequency in the damage plot. The colour should be given as hex colour code (i.e. for magenta, set #ff00ff).

- **-color_g_a COLOR_G_A**

Color for the line representing the G to A misincoporation frequency in the damage plot. The colour should be given as hex colour code (i.e. for magenta, set #ff00ff).

- **-color_instertions COLOR_C_T**

Color for the line representing base insertions in the damage plot. The colour should be given as hex colour code (i.e. for magenta, set #ff00ff).

- **-color_deletions COLOR_DELETIONS**

Color for the line representing base deletions in the dmage plot. The colour should be given as hex colour code (i.e. for magenta, set #ff00ff).

- **-color_other COLOR_OTHER**

Color for the line representing other bases misincorporations in the damage plot. The colour should be given as hex colour code (i.e. for magenta, set #ff00ff).

- **-only_merged**

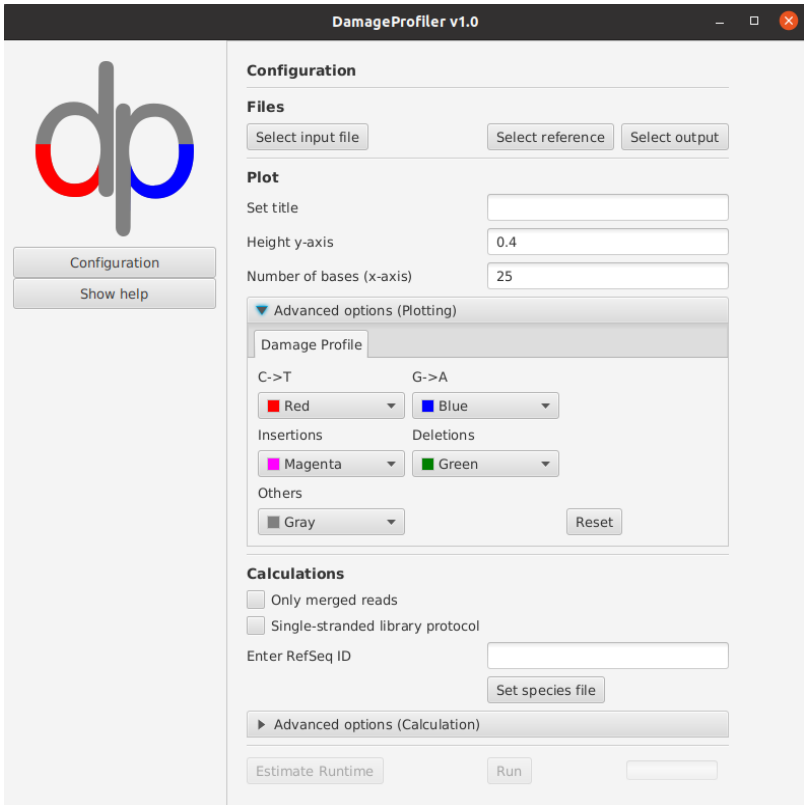
Use only mapped and merged (in case of paired-end sequencing) reads to calculate damage plot instead of using all mapped reads. The SAM/BAM entry must start with 'M_', otherwise it will be skipped. Default: false

- **-sslib**

Single-stranded library protocol was used. Default: false. This option only highlights the C to T base misincorporations in the damage plot.

3.2 GUI

Running the jar file without any parameter starts the GUI to configure the run:



3.3 Log file

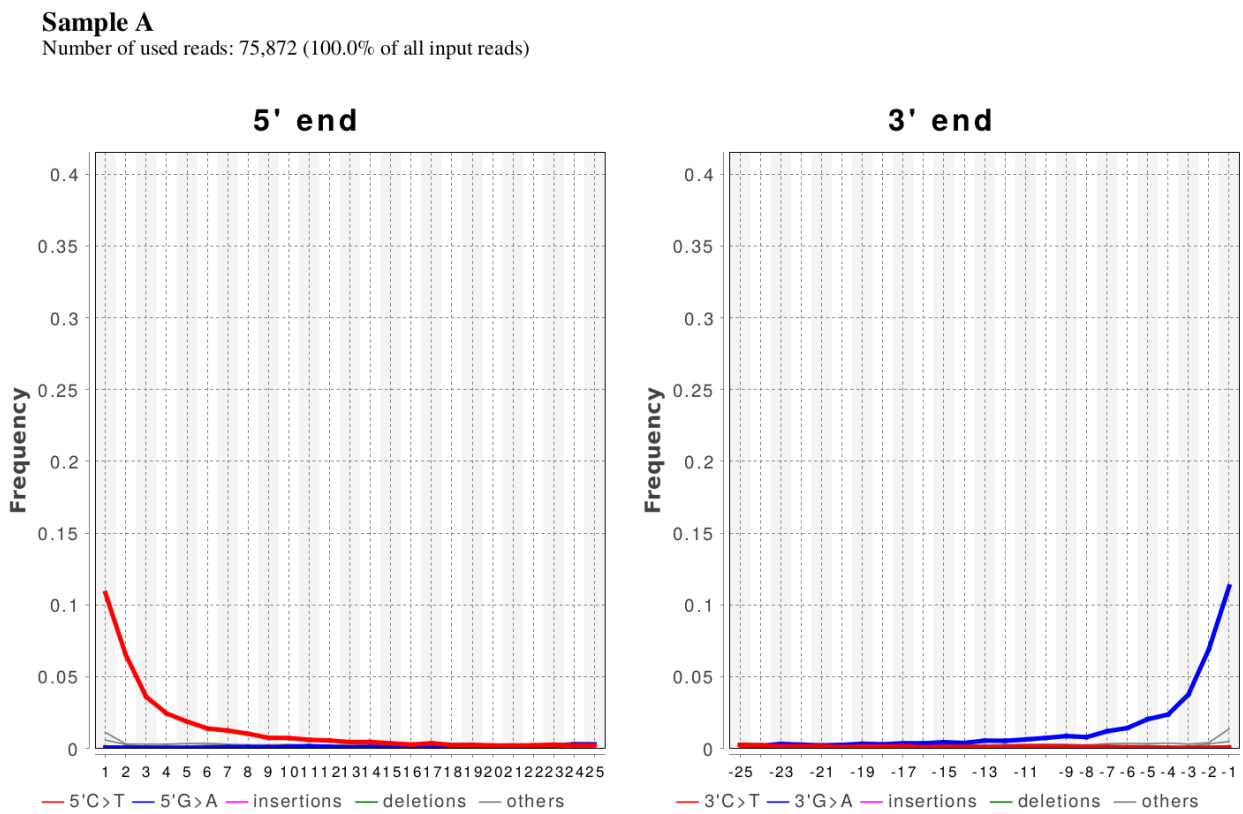
DamageProfiler documents the configuration in a separate log file, which helps you to reproduce your analysis at a later date. The file is saved in the specified result folder.

CHAPTER 4

Output Files

4.1 damagePlot.pdf

The damage plot visualizes the frequency of the particular base misincorporations, deletions, and insertions that occur in the considered reads. The 5' and 3' end of the reads are displayed on the left and right side, respectively. The The x-axis show the position, and the y-axis the frequency. The files *DamagePlot_five_prime.svg* and *DamagePlot_three_prime.svg* contain the visualization as vector graphic for easy further processing.



4.2 5pCtoT_freq.txt and 3pGtoA_freq.txt

These files are tab separated text files, containing the frequency of Cytosine to Thymine and Guanine to Adenine base miscorporation at the 5' and 3'ends, respectively, on which the damage plot is based. The header covers the first three lines, followed by two columns. The first column is the position, starting from the end of the fragment, and the second column contains the frequency of the respective base exchange.

Example *5pCtoT_freq.txt*:

```
# table produced by DamageProfiler
# using mapped file SampleA.bam
# Sample ID: SampleA
pos     5pC>T
1       0.10827902672270852
2       0.06525024039562251
3       0.036067620785707424
4       0.024446388287832053
5       0.018777467039552537
6       ....
```

Example *3pGtoA_freq.txt*:

```
# table produced by DamageProfiler
# using mapped file SampleA.bam
# Sample ID: SampleA
pos     3pG>A
```

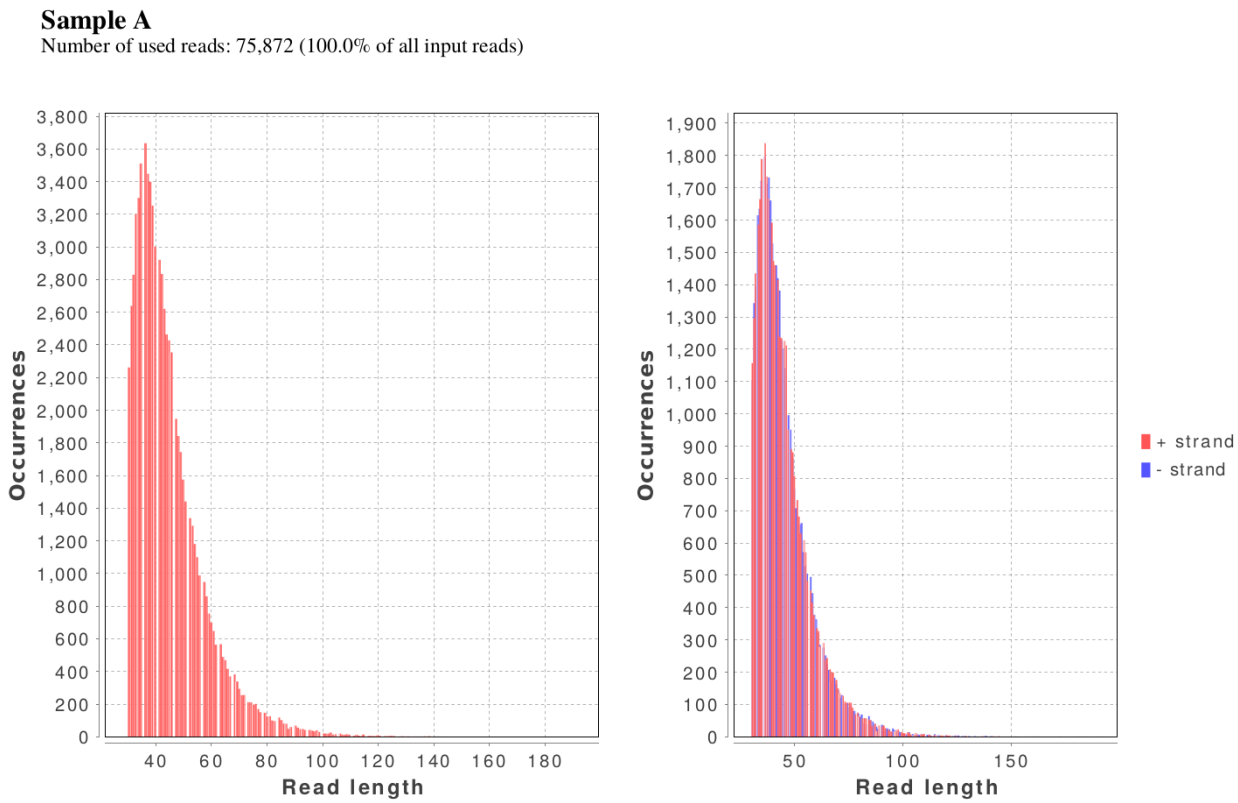
(continues on next page)

(continued from previous page)

```
1       0.11289934178840906
2       0.06908510152863336
3       0.037617996524679474
4       0.023695811903012492
5       0.020417402326950065
6       ....
```

4.3 length_plot.pdf

This figure visualizes the length distribution of all considered reads. The reads length is shown on the x-axis, the number of reads per lentgh on the y-axis. The plot on the left side shows the length histogram of all reads, while the right side separates the reads based on strand orientation. The files *Length_plot_combined_data.svg* and *Length_plot_forward_reverse_separated.svg* provide the plots in svg format.



4.4 lgdistribution.txt

This text file contains a table with read length distributions per strand.

```
# table produced by DamageProfiler
# using mapped file SampleA.bam
# Sample ID: SampleA
# Std: strand of reads
Std     Length  Occurrences
```

(continues on next page)

(continued from previous page)

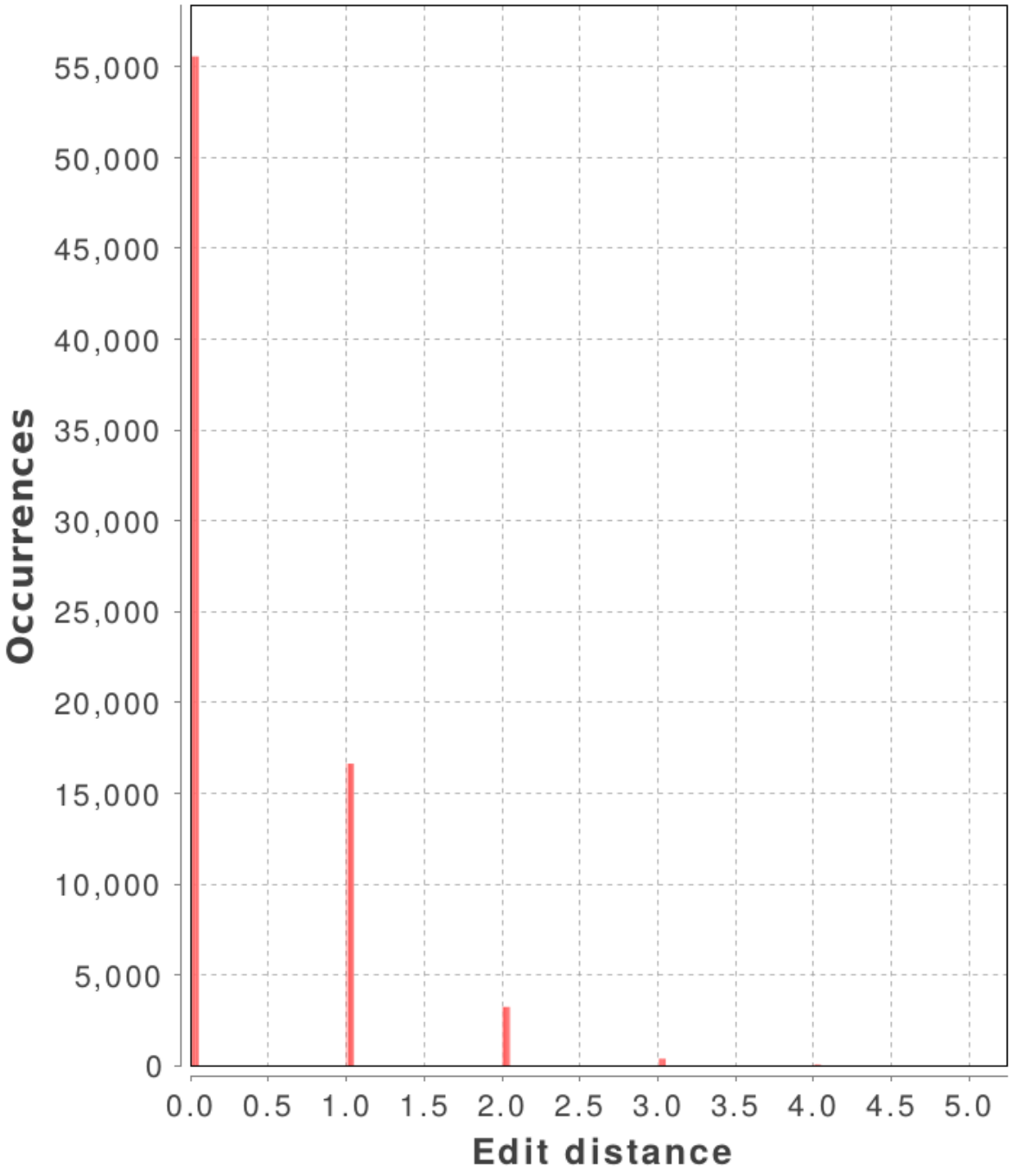
```
+       30.0    1157
+       31.0    1296
+       32.0    1435
...
-       30.0    1105
-       31.0    1343
-       32.0    1395
```

4.5 edit_distance.pdf

A histogram visualizing the number of bases that differ between read and reference. The number of bases (=distance) is shown on the x-axis, the number of reads having this distance (=occurrences) on the y-axis. The file *edit_distance.svg* provides the plot in svg format.

Sample A

Number of used reads: 75,872 (100.0% of all input reads)



4.6 editDistance.txt

This file contains the edit distance distribution of all mapped reads. The edit distance is calculated as the hamming distance between mapped read and the reference.

```
#Edit distances for file: SampleA.bam
    Edit distance   Occurrences
0.0 55569
    1.0     16627
    2.0     3230
    4.0     58
    5.0     9
    3.0     379
```

4.7 misincorporation.txt

This file contains a table with occurrences for each mutations type. The positions are relative positions from the end of the reads.

```
# table produced by DamageProfiler
# using mapped file SampleA.bam
# Sample ID: SampleA
Chr     End     Std     Pos     A       C       G       T       Total   G>A     C>T
↪  A>G     T>C     A>C     A>T     C>G     C>A     T>G     T>A     G>C     G>T     A>
↪-     T>-     C>-     G>-     ->A     ->T     ->C     ->G     S
gi|15826865|ref|NC_002677.1|    3p      +       1       10346.0 8283.0  12587.0 6732.
↪0  37948.0 1401.0  6.0     12.0    12.0    5.0     6.0     46.0    100.0   7.0
↪8.0     2.0     7.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
↪    0.0
gi|15826865|ref|NC_002677.1|    3p      +       2       10329.0 9630.0  11018.0 6971.
↪0  37948.0 775.0   5.0     8.0     7.0     0.0     2.0     33.0    44.0    4.0
↪4.0     1.0     8.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
↪    0.0
gi|15826865|ref|NC_002677.1|    3p      +       3       8692.0  10553.0 10715.0 7988.
↪0  37948.0 419.0   8.0     4.0     9.0     1.0     1.0     17.0    36.0    2.0
↪5.0     0.0     9.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
↪    0.0
gi|15826865|ref|NC_002677.1|    3p      +       4       8959.0  9757.0  10990.0 8242.
↪0  37948.0 259.0   9.0     9.0     9.0     2.0     1.0     3.0     39.0    1.0
↪3.0     0.0     13.0    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
↪    0.0
gi|15826865|ref|NC_002677.1|    3p      +       5       8606.0  10261.0 11081.0 8000.
↪0  37948.0 236.0   6.0     1.0     9.0     0.0     1.0     2.0     34.0    0.0
↪1.0     0.0     19.0    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
↪    0.0
gi|15826865|ref|NC_002677.1|    3p      +       6       8650.0  10351.0 10797.0 8148.
↪0  37946.0 171.0   8.0     2.0     5.0     0.0     0.0     4.0     43.0    4.0
↪3.0     0.0     21.0    0.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0
↪    0.0
gi|15826865|ref|NC_002677.1|    3p      +       7       8573.0  10386.0 10765.0 8221.
↪0  37945.0 132.0   7.0     2.0     1.0     0.0     0.0     1.0     37.0    0.0
↪3.0     0.0     20.0    0.0     0.0     0.0     0.0     1.0     0.0     1.0     0.0
↪    0.0
...
```

4.8 5p_freq_misincorporations.txt and 3p_freq_misincorporations.txt

These files contain the frequencies of all base substitutions per position from the 5' and 3'-ends, respectively.

Example file *5p_freq_misincorporations.txt*:

```
# table produced by DamageProfiler
# using mapped file SampleA.bam
# Sample ID: SampleA
Pos     C>T     G>A     A>C     A>G     A>T     C>A     C>G     G>C     G>T     T>A
↪  T>C     T>G     ->ACGT  ACGT>-
0       0.108279        0.000671        0.000800        0.000640        0.001440
↪  0.000771        0.000270        0.005859        0.011229        0.000428        0.
↪000808        0.000238        0.000000        0.000000
1       0.065250        0.000631        0.000438        0.001168        0.000876
↪  0.000870        0.000321        0.002786        0.003206        0.000047        0.
↪000328        0.000141        0.000013        0.000000
2       0.036068        0.000591        0.000130        0.000972        0.000324
↪  0.001489        0.000192        0.001364        0.003000        0.000057        0.
↪000057        0.000170        0.000013        0.000000
...
```

Example file *3p_freq_misincorporations.txt*:

```
# table produced by DamageProfiler
# using mapped file SampleA.bam
# Sample ID: SampleA
Pos     C>T     G>A     A>C     A>G     A>T     C>A     C>G     G>C     G>T     T>A
↪  T>C     T>G     ->ACGT  ACGT>-
24      0.002608        0.002441        0.000180        0.000240        0.000420
↪  0.002181        0.000095        0.000188        0.002582        0.000119        0.
↪000238        0.000000        0.000013        0.000000
23      0.002354        0.001864        0.000000        0.000427        0.000244
↪  0.002169        0.000185        0.000096        0.002151        0.000118        0.
↪000533        0.000059        0.000000        0.000000
22      0.001550        0.003177        0.000122        0.000183        0.000183
↪  0.002114        0.000000        0.000000        0.002210        0.000061        0.
↪000545        0.000061        0.000000        0.000000
...
```

4.9 DNA_comp_genome.txt

This file contains the basic composition of the sample and parts of the reference to which reads could be mapped.

```
# table produced by DamageProfiler
# using mapped file SampleA.bam
# Sample ID: SampleA
DNA base frequencies Sample
A       C       G       T
0.22213326590555602     0.27659893507234273     0.27791730492742206     0.
↪2233504940946792

DNA base frequencies Reference
A       C       G       T
0.21893033130130574     0.27975782084628925     0.28107944489437814     0.
↪2202324029580269
```

(continues on next page)

(continued from previous page)

4.10 DNA_composition_sample.txt

This files contains the base composition of the reads mapping to the sample per chromosome (Chr), end (End), strand direction (Std) and position (Pos).

```
# table produced by DamageProfiler
# using mapped file SampleA.bam
# Sample ID: SampleA
Chr     End     Std     Pos     A       C       G       T       Total
gi|15826865|ref|NC_002677.1|    3p      +       1       11832   8150    11242   6724 ␣
↪  37948
gi|15826865|ref|NC_002677.1|    3p      +       2       11142   9556    10279   6971 ␣
↪  37948
gi|15826865|ref|NC_002677.1|    3p      +       3       9146    10502   10310   7990 ␣
↪  37948
gi|15826865|ref|NC_002677.1|    3p      +       4       9248    9717    10731   8252 ␣
↪  37948
gi|15826865|ref|NC_002677.1|    3p      +       5       8875    10228   10829   8016 ␣
↪  37948
gi|15826865|ref|NC_002677.1|    3p      +       6       8866    10301   10615   8166 ␣
↪  37948
...
```

4.11 dmgprof.json

The values for the damage profil, the length distribution, and some additional statistics, such as mean, median, and standard deviation of the length distribution are given in json format as well. This is a very common data format for easy data interchange. It is platform independent and usable with many modern programming languages and applications.

4.12 DamageProfiler.log

Each step of the analysis is documented in this file, which facilitates later reproduction of the analysis.

CHAPTER 5

Graphical User Interface

5.1 Load input files

5.2 Run configuration

5.3 Exploration of results

5.4 Metagenomic mapping file

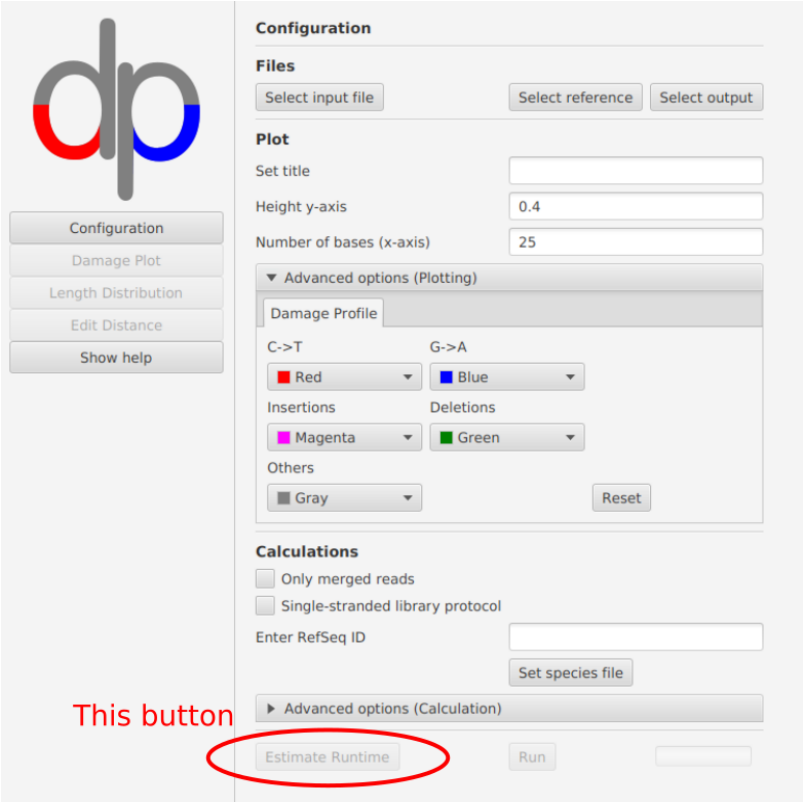
CHAPTER 6

Runtime Estimation

6.1 How to run

The runtime estimation works only when starting DamageProfiler via the *graphical user interface*.

It is possible to estimate the runtime based on the input file size. If all required parameters are set (input file and output directory), the *Estimate Runtime* button will be enabled in the lower part of the GUI.



A window will then open containing information about the file size, the number of records in the input file, and the estimated runtime for processing all read operations. This can either be an actual time span or 'insignificant' if the runtime is less than 1 second. The run can now either be aborted or continued.

6.2 How is the runtime calculated

Coming soon

CHAPTER 7

Indices and tables

- genindex
- modindex
- search