pydamage

Release 0.7

Maxime Borry

Mar 09, 2022

CONTENTS:

Homepage: github.com/maxibor/pydamage

CHAPTER

ONE

PYDAMAGE

Pydamage, is a Python software to automate the process of contig damage identification and estimation. After modelling the ancient DNA damage using the C to T transitions, Pydamage uses a likelihood ratio test to discriminate between truly ancient, and modern contigs originating from sample contamination.

1.1 Installation

1.1.1 With conda (recommended)

```
conda install -c bioconda pydamage
```

1.1.2 With pip

```
pip install pydamage
```

1.1.3 Install from source to use the development version

Using pip

```
pip install git+ssh://git@github.com/maxibor/pydamage.git@dev
```

By cloning in a dedicated conda environment

```
git clone git@github.com:maxibor/pydamage.git
cd pydamage
git checkout dev
conda env create -f environment.yml
conda activate pydamage
pip install -e .
```

1.2 Quick start

```
pydamage --outdir result_directory analyze aligned.bam
```

Note that if you specify `--outdir`, it has to be before the PyDamage subcommand, example: `pydamage --outdir test filter pydamage_results.csv`

1.3 CLI help

Command line interface help message

```
pydamage --help
```

1.4 Documentation

pydamage.readthedocs.io

1.5 Cite

PyDamage has been published in PeerJ: 10.7717/peerj.11845

```
@article{borry_pydamage_2021,
    author = {Borry, Maxime and Hübner, Alexander and Rohrlach, Adam B. and Warinner,␣
↪Christina},
    doi = {10.7717/peerj.11845},
    issn = {2167-8359},
    journal = {PeerJ},
    language = {en},
    month = {July},
    note = {Publisher: PeerJ Inc.},
    pages = {e11845},
    shorttitle = {PyDamage},
    title = {PyDamage: automated ancient damage identification and estimation for␣
↪contigs in ancient DNA de novo assembly},
    url = {https://peerj.com/articles/11845},
    urldate = {2021-07-27},
    volume = {9},
    year = {2021}
}
```

CHAPTER

TWO

API

pydamage.main.**pydamage_analyze**(*bam*, *wlen=30*, *show_al=False*, *process=1*, *outdir=''*, *plot=False*, *verbose=False*, *force=False*, *group=False*)

Runs the pydamage analysis for each reference separately

Parameters

- **bam** (*str*) – Path to alignment (sam/bam/cram) file
- **wlen** (*int*) – window length
- **show_al** (*bool*) – print alignments representations
- **process** (*int*) – Number of processes for parellel computing
- **outdir** (*str*) – Path to output directory
- **verbose** (*bool*) – verbose mode
- **force** (*bool*) – force overwriting of results directory

Returns pandas DataFrame containg pydamage results

Return type pd.DataFrame

CHAPTER

THREE

CLI

To access the help menu:

```
$ pydamage --help
```

The list of arguments of options is detailed below

3.1 pydamage

PyDamage: Damage parameter estimation for ancient DNA
Author: Maxime Borry
Contact: <borry[at]shh.mpg.de>
Homepage & Documentation: github.com/maxibor/pydamage

```
pydamage [OPTIONS] COMMAND [ARGS]...
```

Options

--version
Show the version and exit.

-o, --outdir <outdir>
Output directory

Default pydamage_results

3.1.1 analyze

Run the PyDamage analysis

BAM: path to BAM/SAM/CRAM alignment file. MD tags need to be set.

```
pydamage analyze [OPTIONS] BAM
```

Options

-w, --wlen <wlen>
Window length (in bp) for damage modeling

Default 20

-p, --process <process>
Number of processes for parallel computing

Default 2

-s, --show_al
Display alignments representations

-pl, --plot
Write damage fitting plots to disk

-vv, --verbose
Verbose mode

-f, --force
Force overwriting of results directory

-g, --group
Use entire BAM file as single reference for analyis (ignore reference headers)

Arguments

BAM
Required argument

3.1.2 cite

Get pydamage citation in bibtex format

```
pydamage cite [OPTIONS]
```

3.1.3 filter

Filter PyDamage results on predicted accuracy and qvalue thresholds.

CSV: path to PyDamage result file

```
pydamage filter [OPTIONS] CSV
```

Options

-t, --threshold <threshold>
Predicted accuracy filtering threshold. Set to 0 for finding threshold with kneed method

Default 0.5

Arguments

CSV
Required argument

CHAPTER

FOUR

OUTPUT

Pydamage generates both a tabular and a visual output.

The tabular outputs are comma-separated file (`.csv`) with the following columns, for each analysed reference:

4.1 pydamage_results.csv

- `reference`: name of the reference genome/contig
- `predicted_accuracy`: Predicted accuracy of Pydamage prediction, from the GLM modelling
- `null_model_p0`: parameter `p0` of the null model
- `null_model_p0_stdev`: standard error of the null model paramater `p0`
- `damage_model_p`: parameter `p` of the damage model
- `damage_model_p_stdev`: standard error of the parameter `p` of the damage model
- `damage_model_pmin`: paramater `p_min` of the damage model. *This is the modelled damage baseline*
- `damage_model_pmin_stdev`: standard error of the paramater `p_min` of the damage model
- `damage_model_pmax`: paramater `p_max` of the damage model. *This is the modelled amount of damage on the 5' end.*
- `damage_model_pmax_stdev`: standard error of the paramater `p_max` of the damage model
- `pvalue`: p-value calculated from the likelihood-ratio test-statistic using a chi-squared distribution
- `qvalue`: p-value corrected for multiple testing using Benjamini-Hochberg procedure. *Only computed when multiple references are used*
- `RMSE`: residual mean standard error of the model fit of the damage model
- `nb_reads_aligned`: number of aligned reads
- `coverage`: average coverage along the reference genome
- `CtoT-N`: Proportion of CtoT substitutions observed at position `N` from 5' end
- `GtoA-N`: Proportion of GtoA substitutions observed at position `N` from 5'

4.2 pydamage_filtered_results.csv

Same file as above, but with contigs filtered with `qvalue <= 0.05` and `predicted_accuracy >= threshold` with a user defined filtering threshold (default = 0.5), or determined with the kneedle method.

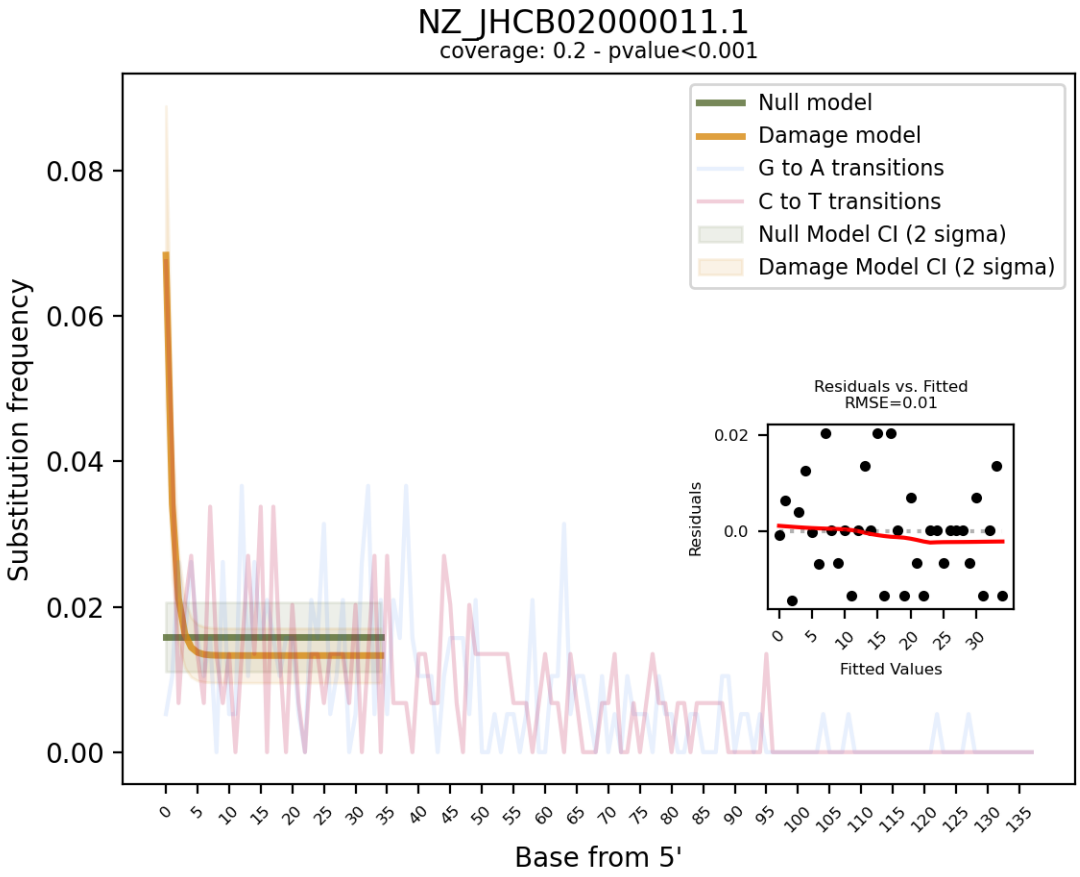
4.3 Plots

The visual output are PNG files, one per reference contig. They show the frequency of observed C to T, and G to A transition at the 5' end of the sequencing data and overlay it with the fitted models for both the null and the damage model, including 95% confidence intervals. Furthermore, it provides a "residuals versus fitted" plot to help evaluate the fit of the pydamage damage model. Finally, the plot contains informtion on the average coverage along the reference and the p-value calculated from the likelihood-ratio test-statistic using a chi-squared distribution.

> The visual output is only produced when using the `--plot` flag

4.4 Example

- **Tabular ouput**
  - pydamage_results.csv
  - pydamage_filtered_results.csv
- **Visual output**


NZ_JHCB02000011.1
coverage: 0.2 - pvalue<0.001
Null model
Damage model
G to A transitions
C to T transitions
Null Model CI (2 sigma)
Damage Model CI (2 sigma)
Substitution frequency
Base from 5'
Residuals vs. Fitted
RMSE=0.01
Residuals
Fitted Values

CHAPTER

FIVE

TROUBLESHOOTING

5.1 My alignment files don't have a MD tag

You can use samtools calmd to set the MD tag

Example:

```
samtools calmd -b alignment.bam reference.fasta > aln.bam
```

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX

Symbols

B

C

M

P