Restriction Site Associated DNA Python-Digested Simulation (RApyDS) Technical Documentation

Last update: November 2019

Contents

1 Technical Specifications

1.1 Software Requirements

- Linux OS
- Python 3+
- Python pip3 (`pip3 install -r requirements.txt`)
- BWA 0.7.12 (http://bio-bwa.sourceforge.net/)
- Firefox (for viewing html files)

1.2 Package Files

```
rapyds/
├── database/
│   └── ...database files
├── docs/
│   └── ...documentation files
├── docs/
│   └── ...enzyme configuration files
├── src/
│   └── ...source files for report
├── templates/
│   └── ...html files for report
├── LICENSE
├── README
├── bwa_aln.sh
├── bwa_index.sh
├── create_histogram.py
├── create_html.py
├── rapyds.py
├── remove_repeat.py
└── tojson.py
```

- **database/** - contains the default restriction enzyme database file
- **docs/** - contains the documentation files and sample output images
- **enzyme/** - contains the enzyme configuration files for the RApyDS run
- **src/** - contains the source css and javascript files for the output report files
- **templates/** - contains the source html files for the output report files
- **LICENSE** - license file

- **README** - quick start readme file
- **bwa_aln.sh** - shell script to run `BWA` alignment of the *reads*
- **bwa_index.sh** - shell script to run `BWA` indexing on the input `FASTA` file
- **create_histogram.py** - python script that plots the RAD Loci density per input sequence, per restriction enzyme
- **create_html.py** - python script that creates the html reports
- **rapyds.py** - main python script
- **remove_repeat.py** - python script that parses the `SAM` file output of the `BWA` program
- **tojson.py** - python script that converts the `csv` files to `json` files

2 Running RApyDS

2.1 Arguments

`-h, --help` - show this help message and exit
`-gc [GC]` - input gc frequency. Value must be between 0 and 1
`-dna [DNA]` - input dna estimated length
`-i [I]` - directory containing the input files
`-pre [PRE]` - prefix of the input files (must match the file name of the sequence, annotation, and/or index files)
`-at [AT]` - what to look for in gene annotation file (ex. gene region, exon, intron, etc) (default: gene)
`-db [DB]` - resteriction enzyme dabatase file.
Format per line: SbfI,CCTGCA|GG
`-re [RE]` - file of list of restriction enzyme to be tested
`-min [MIN]` - minimum fragment size (default: 200)
`-max [MAX]` - maximum fragment size (default: 300)
`-bp [BP]` - base pair read length for FASTQ generation (default: 100)
`-p [P]` - radseq protocol: use ddrad for double digestion (default: orig)
`-o [O]` - output file name (default: report)
`-t [T]` - number of processes (default: 4)

Optional Flags:
`--bwaskip` - skip BWA indexing and alignment
`--clean` - clean files after running

2.2 Input Requirements

- user can only choose between having a directory input (using `-i` with `-pre`) or generating a sequence based on GC frequency (using `-gc` with `-dna`)
- when using directory input, the parameter `-pre <prefix>` is required
- the input directory must contain a sequence file (required), annotation and index files (optional) with file name same as the prefix (ex. `ecoli.fasta` and `ecoli.gff3`). Input sequence files must be in the standard `FASTA` format having the extension of either `.fasta`, `.fna`, or `fna`. While annotation file must in the standard `GFF3` format with extension either be `.gff`, `.gff3`, or `.gtf`.
- user can only choose between `original` and `ddrad` as protocol (`-p`)
- protocol `ddrad` requires a `-re` argument or a list of restriction enzymes to be tested on
- in `original` protocol, if no `-re` argument is given, by default the program uses all the restriction enzymes in the database
- formats for the list of REs and the database are found in the next section.

2.3 File Formats

2.3.1 Database of Restriction Enzymes (-db)

This file will serve as the database of restriction enzymes. Each restriction enzyme must be a line in the file should follow the format: `Enzyme,CUT|SITE`

```
AccI,GT|MKAC
AciI,C|CGC
AclI,AA|CGTT
AfeI,AGC|GCT
AflII,C|TTAAGA
...
```

2.3.2 List of Restriction Enzymes (-re)

Each line of the file must follow the format: `Enzyme` or `Enzyme1 Enzyme2` for ddRAD protocol.

```
AccI
AciI
AclI
AfeI
```

```
AflII
...
```

2.3.3 Sequence File

The input genome file must follow the standard `FASTA` format. Below is an example for the first few lines:

```
>U00096.3 Escherichia coli str. K-12 substr. MG1655, complete
    genome
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC

TTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGACTTAGGTCACTAAATACTTTAACCAA

TATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAAACGCATTAGCACCACC

ATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAAACACAGAAAAAAG

CCCGCACCTGACAGTGCGGGCTTTTTTTTTTCGACCAAAGGTAACGAGGTAACAACCATGCGAGTGTTGAA
```

The program can accomodate multiple `FASTA` in one single file as long as each of the `FASTA` starts with a `>` symbol followed by its *identifier*.

2.3.4 Feature / Annotation File

The feature or annotation file must follow the General Feature Format 3(GFF3). Below is an example for the first few lines:

```
##sequence-region U00096.3 1 4641652
##species https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.
    cgi?id=511145
U00096.3 Genbank region 1 4641652 . + . ID=id-1;Dbxref=taxon
    :511145;Is_circular=true;Name=ANONYMOUS;gbkey=Src;genome=
    chromosome;mol_type=genomic DNA;strain=K-12;substrain=MG1655
U00096.3 Genbank gene 190 255 . + . ID=gene-b0001;Dbxref=EcoGene:
    EG11277;Name=thrL;gbkey=Gene;gene=thrL;gene_biotype=
    protein_coding;gene_synonym=ECK0001,JW4367;locus_tag=b0001
U00096.3 Genbank CDS 190 255 . + 0 ID=cds-AAC73112.1;Parent=gene-
    b0001;Dbxref=ASAP:ABE-0000006,UniProtKB/
```

Similar to the genome file, the program can also accomodate multiple `GFF` in one file provided that the *identifier* in the genome file must match a `sequence-region` in the `GFF` file.

2.4 Common Usage

Original RADSeq with:

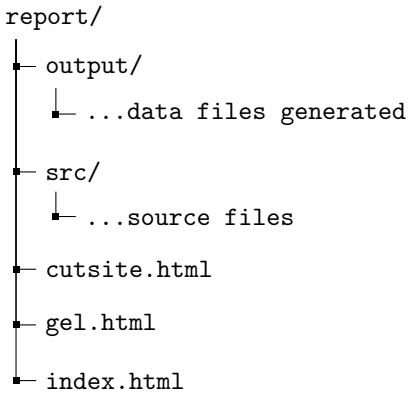
- known or given input sequence file `ecoli.fasta` inside a directory `input_dir`
  `./rapyds.py -i input_dir -pre ecoli [other arguments]`
- known or given input sequence file with custom restriction enzyme list
  `./rapyds.py -i input_dir -pre ecoli -re <path/to/RE_file.txt> [other arguments]`
- unknown sequence but with GC content/frequency
  `./rapyds.py -gc <GC frequency> -dna <sequence length> [other arguments]`

DDRad with known genome:
`./rapyds.py -i input_dir -pre ecoli -p ddrad -re <re_file.txt> [other arguments]`

In case `./rapyds.py` didn't work, an alternative is running using `python rapyds.py`.

3 Output Files

After running RApyDS, it will produce a `.zip` file containing the generated report. The contents of zipped file has the following structure:

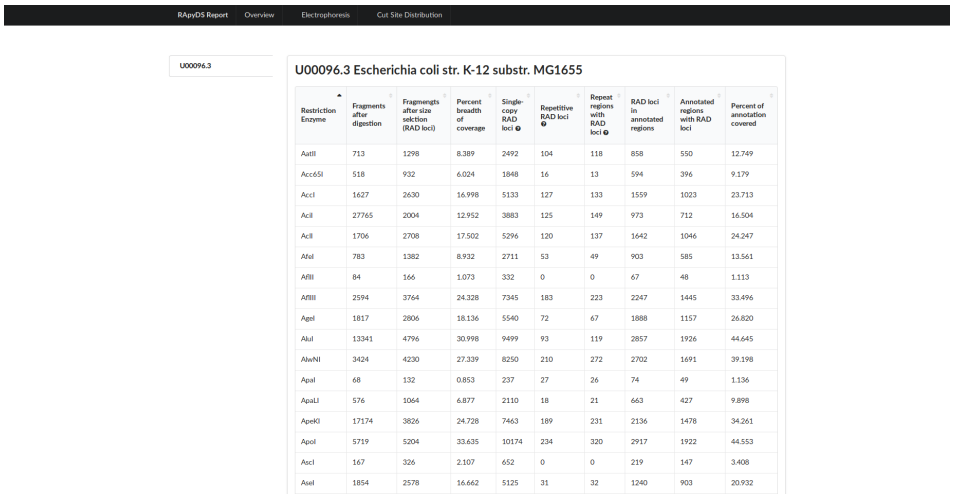


There will be 3 html files:

- `index.html` contains the tabular information about the fragments
- `gel.html` is the electrophoresis simulation
- `cutsite.html` shows the cut site locations

3.1 Overview (index.html)

This html file shows information about the fragments after in silico digestion. Column headers can be clicked to sort the table according to the column's value in increasing or decreasing order. And on the left is a side bar with list of sequence identifier in the input FASTA file.



Restriction Enzyme	Fragments after digestion	Fragmengts after size selction (RAD loci)	Percent breadth of coverage	Single-copy RAD loci	Repetitive RAD loci	Repeat regions with RAD loci	RAD loci in annotated regions	Annotated regions with RAD loci	Percent of annotation covered
AatII	713	1298	8.389	2492	104	118	858	550	12.749
Acc65I	518	932	6.024	1848	16	13	594	396	9.179
AccI	1627	2630	16.998	5133	127	133	1559	1023	23.713
AciI	27765	2004	12.952	3883	125	149	973	712	16.504
AclI	1706	2708	17.502	5296	120	137	1642	1046	24.247
AfeI	783	1382	8.932	2711	53	49	903	585	13.561
AflII	84	166	1.073	332	0	0	67	48	1.113
AflIII	2594	3764	24.328	7345	183	223	2247	1445	33.496
AgeI	1817	2806	18.136	5540	72	67	1888	1157	26.820
AluI	13341	4796	30.998	9499	93	119	2857	1926	44.645
AlwNI	3424	4230	27.339	8250	210	272	2702	1691	39.198
ApaI	68	132	0.853	237	27	26	74	49	1.136
ApaLI	576	1064	6.877	2110	18	21	663	427	9.898
ApeKI	17174	3826	24.728	7463	189	231	2136	1478	34.261
ApoI	5719	5204	33.635	10174	234	320	2917	1922	44.553
AscI	167	326	2.107	652	0	0	219	147	3.408
AseI	1854	2578	16.662	5125	31	32	1240	903	20.932

Figure 1: Overview table of the in silico digestion (original RAD) for E.Coli

3.2 Electrophoresis (gel.html)

One of the two visualisation the program presents is the electrophoresis simulation of the fragments after digestion. This makes use of `d3.js` and `d3-electrophoresis` javascript plug-ins.

The user first selects the genome from the list, input the desired markers, and selects up to five restriction enzymes. The webpage will read from the `json` output of RApyDS program. Loading the webpage might slow down from this part especially if the number of fragments is large. There is another option to further size select the digested fragments using the `start` and `end` input boxes. The page will only display the fragments within the desired size select range.

3.3 Cut Sites (cutsite.html)

The second and last visualisation is the cut site distribution. This also uses `d3.js` javascript plugin.

For this visualisation, the data is from another `json` file also generated by the program which contains the location of the cut sites in the sequence. It is then rendered as a black line in perpendicular to the location in the sequence (in bp) as the x-axis.

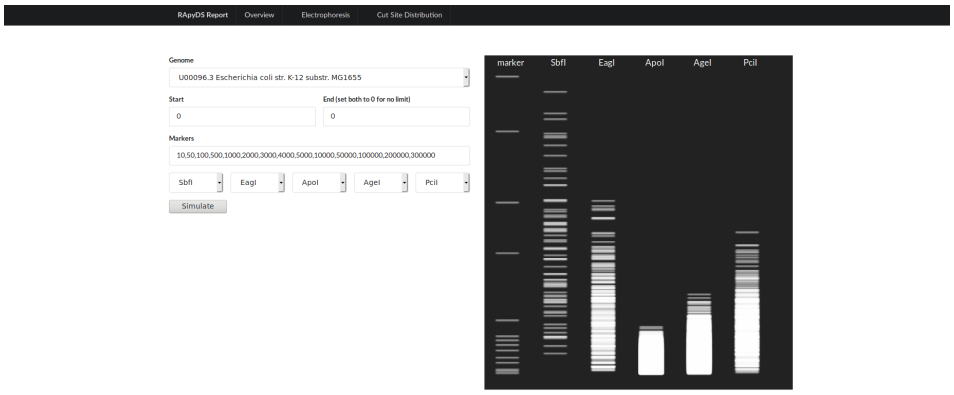


Figure 2: Electrophoresis simulation for E.Coli enzymes SbfI, EagI, ApoI, AgeI, and PciI.

4 RAD Loci Density

A python script named `create_histogram.py` can generate a histogram of the RAD Loci density either with a given bin size or a fixed number of bins.

To generate the graphs with a fixed bin size (ex. 500,000bp):
`python create_histogram.py -binsize 500000 <path/to/report/output/folder>`

To generate the graphs with a fixed number of bins (ex. bins=20):
`python create_histogram.py -nbin 20 <path/to/report/output/folder>`

The `<path/to/report/output/folder>` corresponds to the `output/` folder inside the extracted report files. See Section Output Files.

All generated histgrams will be located inside the `report/output/images/density` folder.

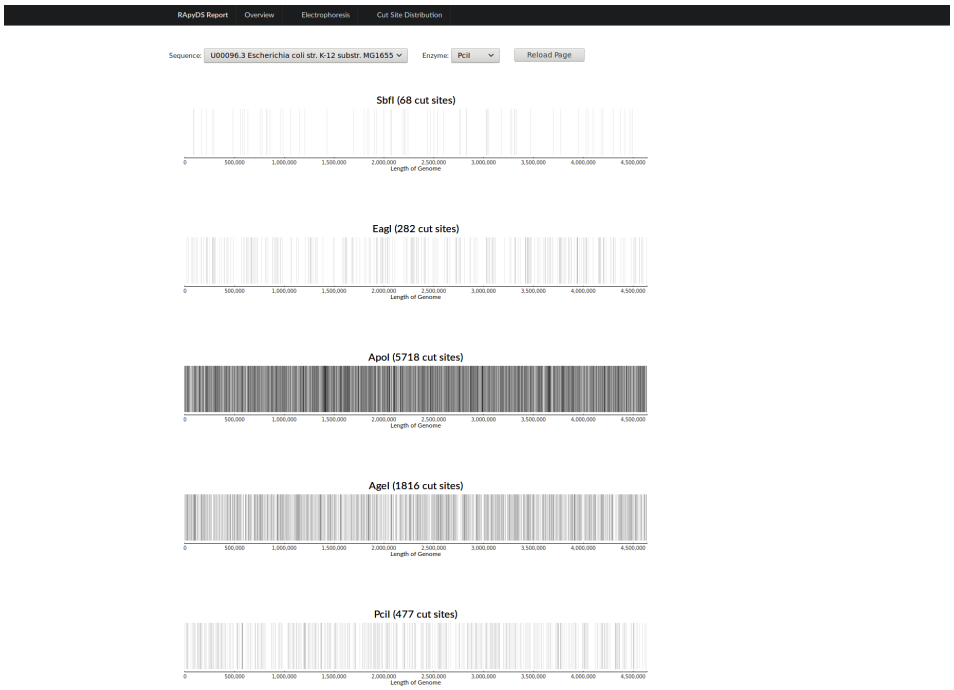


Figure 3: Cut site markers for E.Coli with enzymes SbfI, EagI, ApoI, AgeI, and PciI.

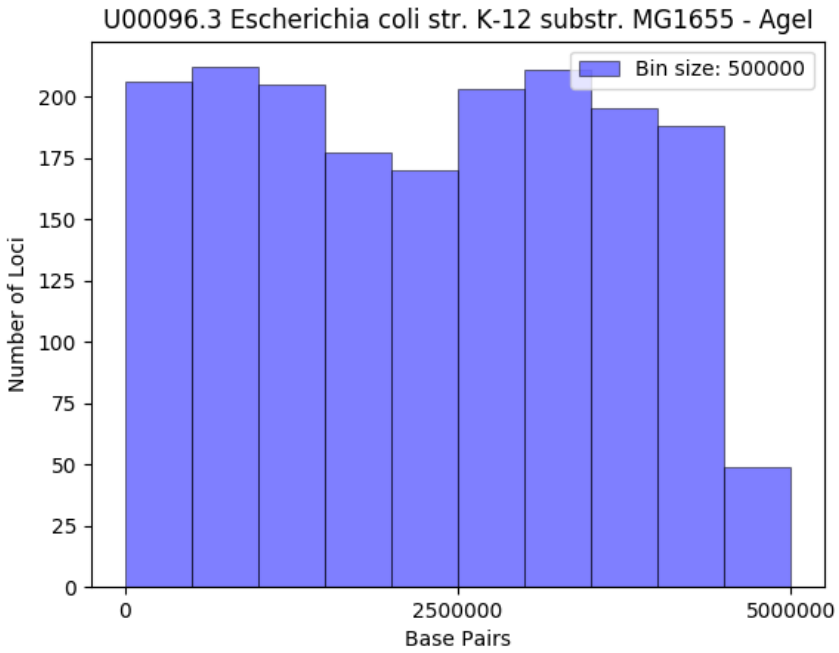


Figure 4: RAD loci density of E.Coli digested by enzyme AgeI with bin size = 500,000bp.

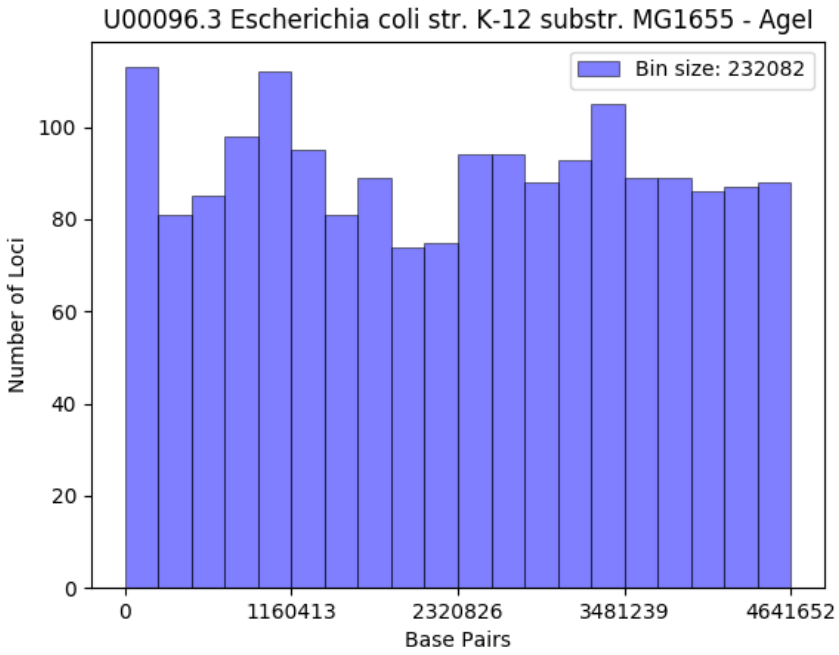


Figure 5: RAD loci density of E.Coli digested by enzyme AgeI with number of bins = 20.