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

Last update: August 2018

Contents

1 Technical Specifications

1.1 Software Requirements

- Linux OS
- Python 2.7 or greater
- numpy (`pip install numpy`)
- BWA 0.7.12 (http://bio-bwa.sourceforge.net/)
- Firefox (for viewing html files)

1.2 Package Files

```
rapyds/
├── data/
│   └── ...contains sample files
├── docs/
│   └── ...documentation files
├── src/
│   └── ...source files for report
├── templates/
│   └── ...html files for report
├── LICENSE
├── README
├── bwa_aln.sh
├── bwa_index.sh
├── create_html.py
├── rapyds.py
├── remove_repeat.py
└── tojson.py
```

- **data/** - contains sample files (`FASTA`, `GFF` and list of REs) and default restriction enzyme database

- **docs/** - contains the documentation files and sample output images
- **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_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
`-i [I]` - input genome sequence file (FASTA)
`-db [DB]` - resteriction enzyme dabatase file.
Format per line: SbfI,CCTGCA|GG
`-re [RE]` - file of list of restriction enzyme to be tested
`-a [A]` - annotation file for genome (GFF)
`-at [AT]` - what to look for in gene annotation file (ex. gene region, exon, intron, etc) (default: gene)
`-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)
`-gc [GC]` - input gc frequency. Value must be between 0 and 1
`-dna [DNA]` - input dna estimated length
`-o [O]` - output file name (default: report)
`-t [T]` - number of processes (default: 4)

Notes:

- user can only choose between `original` and `ddrad` as protocol (`-p`)

- user can only choose between having a genome input or generating a sequence based on GC frequency
- protocol `ddrad` requires a `-re` argument or a list of restriction enzymes
- 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.

2.2 File Formats

2.2.1 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
...
```

2.2.2 List of Restriction Enzymes (-re)

Each line of the file must follow the format: `Enzyme,CUT|SITE`

```
AccI
AciI
AclI
AfeI
AflII
...
```

2.2.3 Genome File

The input genome file must follow the `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*.

2.2.4 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.

2.3 Common Usage

Original RADSeq with:

- known or given genome file (FASTA format)
  `./rapyds.py -i <genome_file.fasta> [other arguments]`
- known or given genome file (FASTA format) with custom restriction enzyme list
  `./rapyds.py -i <genome_file.fasta> -re <restriction_enzymes> [other arguments]`
- unknown genome but with GC content/frequency
  `./rapyds.py -gc <GC frequency> -dna <sequence length> [other arguments]`

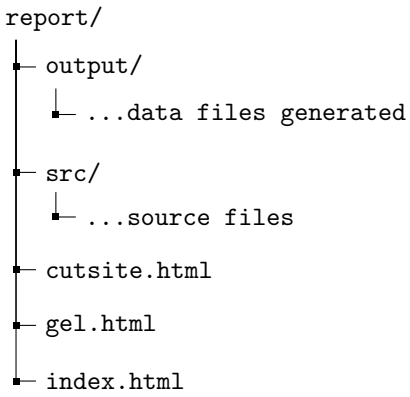
DDRad with known genome:

`./rapyds.py -i <genome_file.fasta> -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 zipped file contains the following structure:



There will be 3 html files:

- `index.html` contains a tabular information about the fragments from in silico digestion
- `gel.html` is the electrophoresis simulation
- `cutsite.html` contains the cut site location simulation

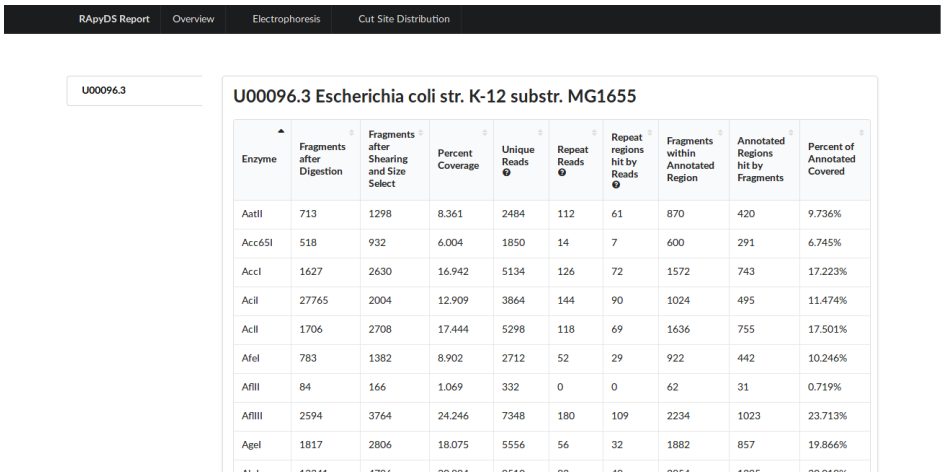
3.1 Overview (index.html)

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.

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 the `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. Webpage load might slow down from this part especially if the number of fragments are large. To address this, the fragments can be further size selected using the `start` and `end` input boxes.



Enzyme	Fragments after Digestion	Fragments after Shearing and Size Select	Percent Coverage	Unique Reads	Repeat Reads	Repeat regions hit by Reads	Fragments within Annotated Region	Annotated Regions hit by Fragments	Percent of Annotated Covered
AatII	713	1298	8.361	2484	112	61	870	420	9.736%
Acc65I	518	932	6.004	1850	14	7	600	291	6.745%
AccI	1627	2630	16.942	5134	126	72	1572	743	17.223%
AciI	27765	2004	12.909	3864	144	90	1024	495	11.474%
AclI	1706	2708	17.444	5298	118	69	1636	755	17.501%
AfeI	783	1382	8.902	2712	52	29	922	442	10.246%
AflII	84	166	1.069	332	0	0	62	31	0.719%
AflIII	2594	3764	24.246	7348	180	109	2234	1023	23.713%
AgeI	1817	2806	18.075	5556	56	32	1882	857	19.866%

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

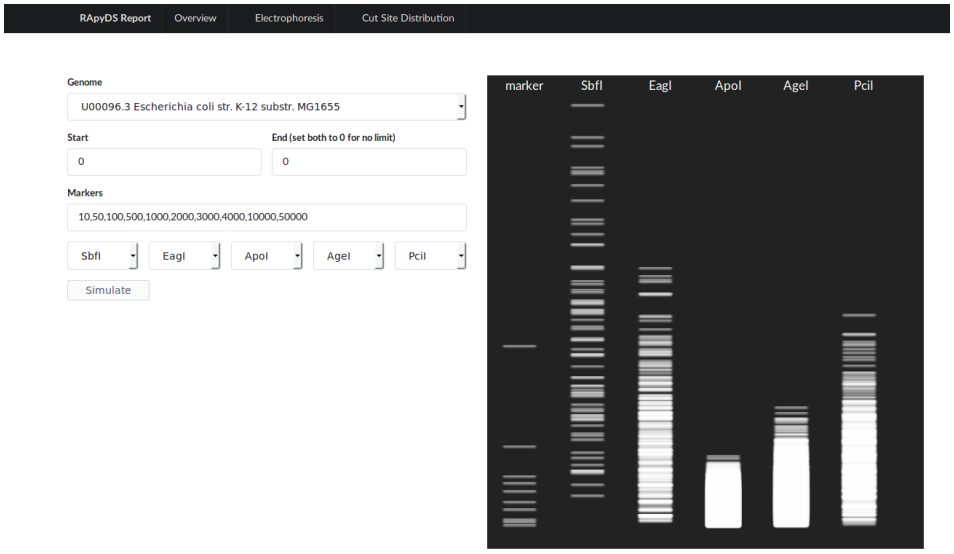


Figure 2: Electrophoresis simulation for E.Coli with enzymes AgeI, ApoI, EagI, PciI, SbfI

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 genome. It is then rendered as a black line in perpendicular to the genome location as the x-axis.

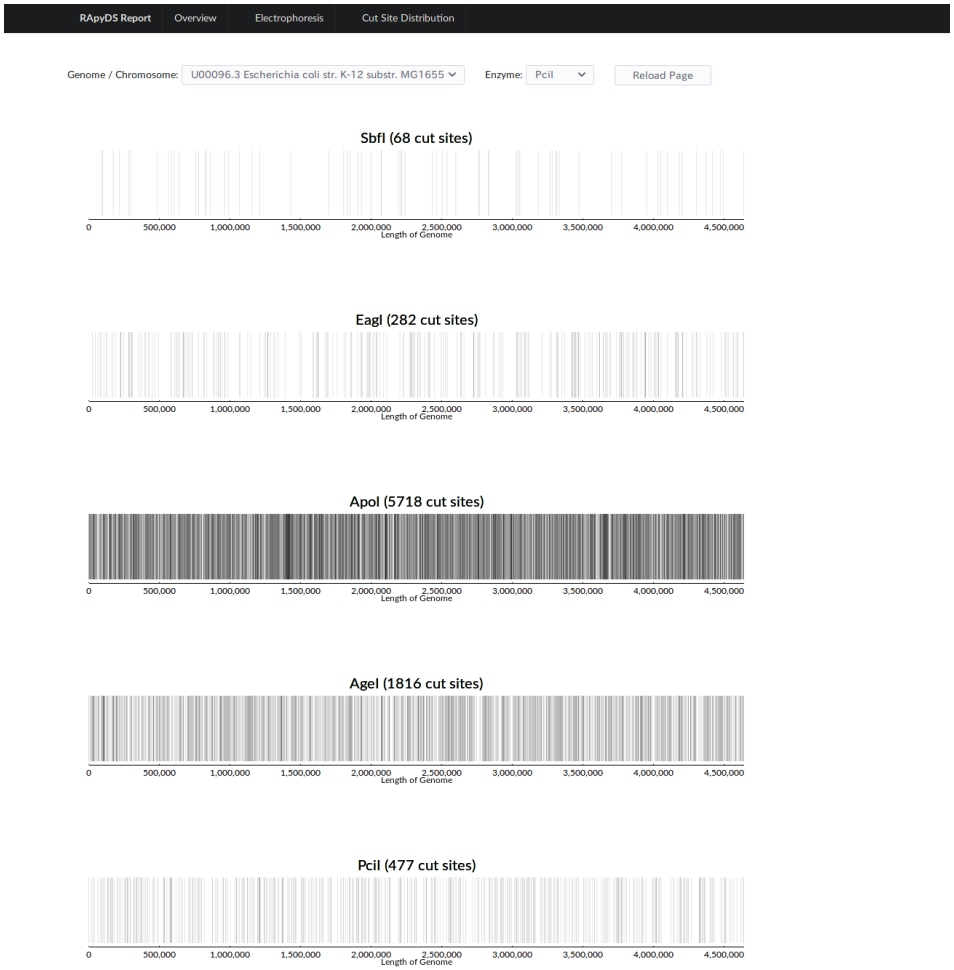


Figure 3: Cut site markers for E.Coli with enzymes AgeI, ApoI, EagI, PciI, SbfI