DensityMap

Table of content

1 Purpose

Nowadays, genomic data visualization became a real challenge. A lot of tools, such as Gbrowse, Jbrowse, Abrowse …, already exists for displaying genomic data of small genomic locus. But only a few try the challenge of the whole chromosome/genome visualization. Phenogram and CviT gives two different answers to this problem but have their own limits. Because we work on repeated sequences, which can be very numerous in some genome (45 % in Human genome, over 90 % Wheat genome), we needed to develop a program fitted to this kind of heavy data. That why DensityMap has been designed to represent the density (number of base pair covered) of one or several features along chromosomes using a serial of windows. The feature correspond to the third column of GFF annotation file (format: http://www.sequenceontology.org/gff3.shtml) used as input file. The program generate SVG pictures which have the advantage produce high quality and re-editable pictures.

This program bring the possibility to represent the content GFF files in a simple manner with way to select the data to plot, automatic procedure of chromosome scaling, and highly configurable output picture with the help of lot of graphics options.

2 Installation

2.1 Requirements

DensityMap is a Perl script that only need one requirement to work, GD::SVG (tested with version 0.33-1). To install GD::SVG libraries on Debian based system, you use apt-get package manager:

```
sudo apt-get install libgd-svg-perl
```

It also use POSIX libraries that are included in linux distributions for ceil and floor functions. As optional requirement you can install Term::ANSIColor libraries for a more colourful output. Download the perl module from cpan:

https://metacpan.org/pod/Term::ANSIColor.

Then install it by following the standard perl module installation procedure:

```
perl Makefile.PL
make
make test
sudo make install
```

To activate the coloured terminal output, uncomment lines 16 and 17 in DensityMap.pl script.

2.2 DensityMap Installation

Download the DensityMap archive or clone repository from github:

https://github.com/sguizard/DensityMap.

The program can be launched directly from the unzipped archive or you can execute the install script for a all users installation:

```
sudo install.sh
```

It will create a directory DensityMap in /usr/local/share and copy all files in it. Then it update the path of colours.txt in DensityMap.pl and create a symbolic link into /usr/local/bin.

3 How to use

Mandatory options

- -i, -input
  - string
  - Name of Gff file(s)
- -re, region_file
  - string
  - Name of bed file
    - This bed file allow to compare specific regions of each sequences ant not to plot the while sequence. It's a tabulated format with 3 columns. The first one contain the sequence name, second one the start of the region to plot and the third one the end of the region to plot .
- -o , -output_img_name
  - string
  - Name of the output image, this extension .svg will be automatically added
- -ty, -type_to_draw
  - string

        - List of type to draw, strand to plot and colour scale to use
        - Format: "Type1=strand;Type2=strand=8"
            - Type (third column of GFF): match, gene, CDS, ...
            - Strand:
                - - → strand -
                - + → strand +
                - both → strand - and strand +
                - fused → Combination of strand - and strand +
                - all → strand - and strand + and fused

<u>Generic options</u>

- -v, -verbose: MORE text dude !!!!
- -h, -help: This help
- -for, -force: Automatically answer yes to picture size validation

<u>Density options</u>

- -c, -colour_scale:
    - integer
    - colour scale to use
    - (Default = 7)
- -sc, -scale_factor:
    - integer
    - window length in bp
    - (Default = 1000)
- -a, -auto_scale_factor:

    - integer
    - Max picture height in pixel
- -ro, -rounding_method:
    - string
    - floor or ceil
    - (Default = floor)
- -gc:
    - integer
    - colour scale to use
    - Create a density map a the GC% along the chromosome, REQUIRE the presence of the fasta sequence in the ##FASTA section of the GFF file

<u>Graphical options</u>

- -ti, -title
    - string
    - Title to print on the picture
- -w, -win_size
    - integer
    - Height of window in pixel
    - Default: 1
- -sh, -show_scale
    - integer
    - Draw Scale, n = num max ticks
    - Default: 50
- -str_w, -str_width

  - integer
  - Strand width in pixel
  - Default: 50
- -str_s, -str_space
  - integer
  - Space between strands in pixel
  - Default: 50
- -sp, -space_chr
  - integer
  - Space between chromosomes in pixel
  - Default: 50
- -lm, -lmargin
  - integer
  - Left margin in pixel
  - Default: 50
- -rm, -rmargin
  - integer
  - Rigth margin in pixel
  - Default: 50
- -tm, -tmargin
  - integer
  - Top margin in pixel
  - Default: 50
- -bm, -bmargin
  - integer

    - Bottom margin in pixel
    - Default: 50
- -ba, background
    - color
    - Fill Background
    - Default: no Background
- -la, label_strand_rotation
    - integer
    - Rotation degree of strand label
    - Default: 0
- ft_f, ft_family:
    - string
    - Font to use for text
    - Default: “Helvetica”
- ft_s, ft_size:
    - integer
    - Size of the font
    - Default: 16

Running the program:

To be valid, the GFF setted in input must respect the format given on the following website: http://www.sequenceontology.org/gff3.shtml. But it also need to contain for each chromosome the header sequence-region.

Example of valid file:

```
##sequence-region 2L 1 23513712
2L     RefSeq gene   7529   9484   .      +      .      ID=gene2671
2L     RefSeq gene   9839   21376  .      -      .      ID=gene2672
…
2L     RM     LTR    23512506      23512653      809    +      .      Target=DM297_I-int 3285
3432
```

```
##sequence-region 2R 1 25286936
2R      RefSeq gene    432219 705848 .      +      .      ID=gene6156
…
##FASTA
>seq1
ALONGSEQ...
```

You need to set the mandatory options to create a basic image of your data:

```
DensityMap.pl -i 2R.gff3 -ty "LTR=fused" -o 2R
```

This program is also executed with default graphical options (see above list) and will compute the size of output picture and ask you if want to continue or stop the execution. As you can see, the produced picture will be over 25 000 pixels height.

To get a reasonable picture height, two options are available to you. First you define a window size to use to compute the density of feature using the -sc option. Or you can let the program choose for you and define a maximum picture height not to exceed with the option -a.

You can try to use the same command with the option -a 3000 to obtain a picture with a height not exceeding 3000 pixels and add the -ba white option to have a white background on the picture.

```
DensityMap.pl -i 2R.gff3 -ty "LTR=fused" -o 2R -a 3000 -ba white
```

The program will prompt the window size (scale factor) and ask you to validate or not the picture height. The produced picture size is 150 px by 2653 px and so can be easily viewable (Figure 1). The default colour scale goes from blue to red which represent nucleotide windows with low density of feature (LTR) and nucleotide windows with high density of feature.

You can several chromosomes by pooling the GFF files on the -g option. You can also display information for different type by defining them in the -ty option. With help of the graphical options you can add a scale a title to the picture (Figure 2).

```
DensityMap.pl -i dmel.gff3 -o dmel -ty "LTR=fused;LINE=fused" -ba white -sc
20000 -sh 100 -title "LTR and LINE retrotransposon in Dmel genome"
```

This way you obtain a density map of the whole genome of *Drosophilia Melanogaster* describing the density of LINE and LTR retrotransposons. As you can see on the produced picture, the LTR retrotransposons shows highest density than LINE and share a similar distribution with high concentration at extremities of each chromosome and large areas devoid of transposable elements, except for chromosome 4 and X which are quasi devoid of transposable elements.

DensityMap can be also used for studying low density features like Rolling Circle transposons. To do this you can use another colour scale, like the number 9. It have designed to set a blank pixel for windows with a density between 0 and 1 % of density and a red pixel for windows with a density over 1 %. You also modify the way the density is rounded. By default, densities are rounded by using floor method, so if the density under 1 %, it will be rounded to 0 %. If you set the rounding method (-ro) to ceil, the value will rounded to 1 %. So if you need to maximize the visibility of low density features you should use the ceil method (Figure 3).

```
DensityMap.pl -i dmel.gff3 -o RC -ty "RC=fused" -ba white -sc 20000 -sh 100
-title "Rolling Circle transposons in Dmel genome" -c 9 -ro ceil
```

If you known the position of centromere inside chromosomes you can add their coordinates in the GFF file with the type centromere. So they will be displayed on chromosomes as shown **as example** on figure 3 chromosome 3R.

You can modify all margin of the picture (bottom, top, left, right), space between the strands and the with of strands.

Customizing colours scheme:

DensityMap is delivered with 10 colour scale for representing the density. You may not find the suitable colour scale for your representation. You can create your own by modify the colours.txt installation directory. A colour scale is composed of 101 colours and each of it contain four information:

- An id formatted: <ID>_heatmap<COLOURNUMBER>
- A Red value ranging from 0 to 255
- A Green value ranging from 0 to 255
- A Blue value ranging from 0 to 255

Theses information are separated by an semi-column.

Example of colour scale:

```
1_heatmap10;0;200;0
1_heatmap11;0;195;0
…
1_heatmap99;245;0;0
1_heatmap100;250;0;0
```

The DensityMap archive bring a script that read the colours.txt file and create a miniature of all colours scales available (Figure 4). You only need to execute the scaleColorDrawer.pl script in the same directory of colours.txt without any arguments.

Figure 1

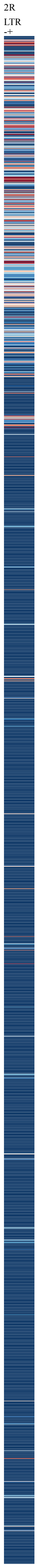
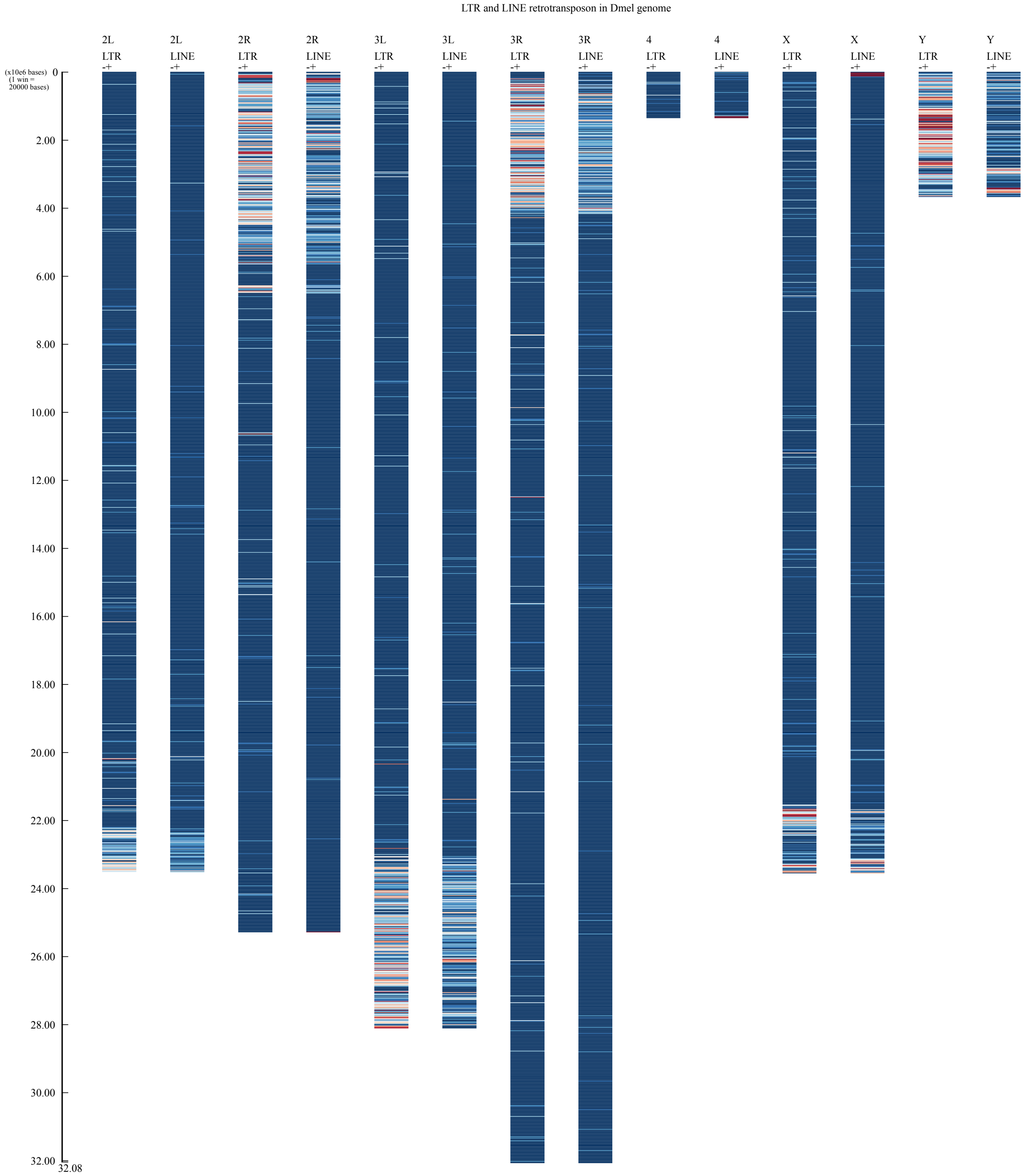
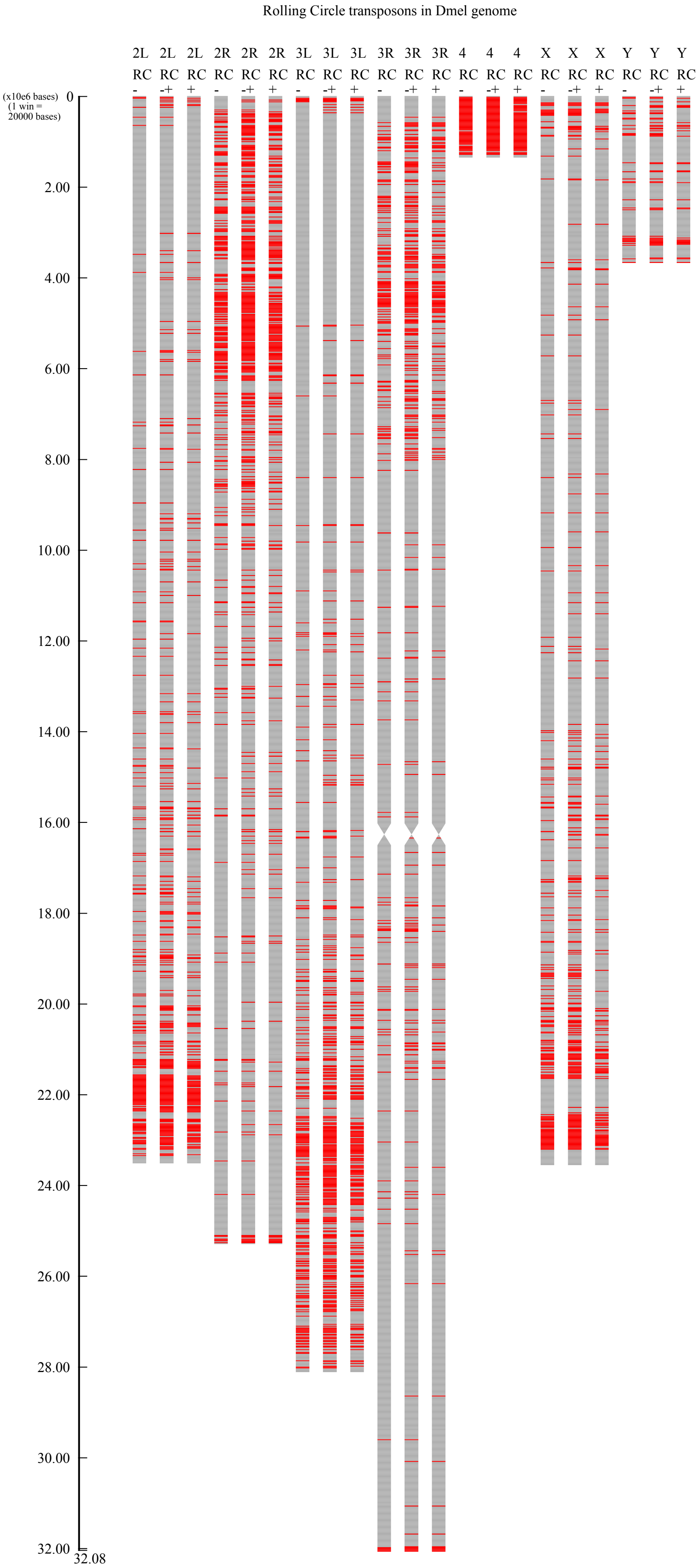


Figure 2



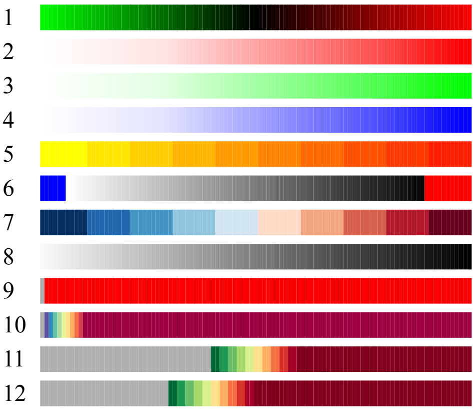
LTR and LINE retrotransposon in Dmel genome
2L LTR -+
2L LINE -+
2R LTR -+
2R LINE -+
3L LTR -+
3L LINE -+
3R LTR -+
3R LINE -+
4 LTR -+
4 LINE -+
X LTR -+
X LINE -+
Y LTR -+
Y LINE -+
(x10e6 bases) (1 win = 20000 bases)
0
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
20.00
22.00
24.00
26.00
28.00
30.00
32.00
32.08

Figure 3



Rolling Circle transposons in Dmel genome
2L RC -
2L RC -+
2L RC +
2R RC -
2R RC -+
2R RC +
3L RC -
3L RC -+
3L RC +
3R RC -
3R RC -+
3R RC +
4 RC -
4 RC -+
4 RC +
X RC -
X RC -+
X RC +
Y RC -
Y RC -+
Y RC +
(x10e6 bases)
(1 win =
20000 bases)
0
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
20.00
22.00
24.00
26.00
28.00
30.00
32.00
32.08

Figure 4:


1
2
3
4
5
6
7
8
9
10
11
12