
REPORT

ON

CNV Detection

1.Data base creater

1.1 Normalization reads count

In this step we used normalization method to find normal reads count for each file and formed new column name(**nor_reads**).



Data ['nor_reads'+new_s] = (data['reads']/(data['reads'].sum()))*1000

Data :- data frame for each file

1.2 Adjusted reads count

In this step we used normalization method to adjusted reads count for each file and formed new column name(**ARC**).



- **D** = data_1_22['reads'].mean()
- **dgc** = data_1_22.loc[(data_1_22['gc'] >= data.gc[i]-0.1) & (data_1_22['gc'] <= data.gc[i]+0.1), 'reads'].mean()
- **ARC** = data.reads[i]*(d/dgc)

Data_1_22 – In these data we have only chromosomes 1 to 22

Arguments :-

- **-N(--normalization_file_location) :-** Files location of all normal files.
- **-M(--normalization_method) :-** Which method we used 'gc' or 'reads'.
- **-R(--reference_data_base) :-** output file(find mean and standard deviation for all normal samples as reference database used in further programme).

Script Name :- data_base_creator.py

How to run this programme??

```
(base) C:\Users\EDGC_IN_01>python data_base_creator.py -N C:\Users\EDGC_IN_01\cnv_FF6\cnv_FF6 -M gc -R reference.txt
```

Result :- We have reference data base file as a result which include space(all 24 chromosomes) , start bin , end bin (start bin and end bin difference 50k) , mean and standard deviation for all normal samples.

Reference data base is used in further programme to find duplication and deletion for each file.

2.CNV

2.1 Detect Duplication and Deletion and Draw Plots

df :- groupby data of all chromosome.

Zscore :- Add a column to zscore to find zscore value for sample.



$$\text{df['zscore']} = (\text{df}[\text{'data_r'}] - \text{df1}[\text{'nor_mean'}]) / \text{df1}[\text{'nor_std'}]$$

df1 :- groupby data of reference data base.

We have zscore database file

Z :- we create z' _score database using concept of rolling surrounding 10 bins

Window size is depend on user

Example:- window =10

```
df.loc[:, 'z'] = df['zscore'].rolling(window=args.window_size, min_periods=1, center=True).mean()
```

2.2 find Duplication and Deletion

- **Duplication -** In chromosomal duplications, extra copies of a chromosomal region are formed, resulting in different copy numbers of **genes** within that area of the **chromosome**. Continue greater than or equal to 10 bins having value greater than 1.5 (standard value).
- **Deletion -** Deletions involve the loss of DNA sequences. Phenotypic effects of deletions depend on the size and location of deleted sequences on the genome. Continue greater than or equal to 10 bins having value less than -1.5 (standard value).

Examples:- Show duplication and deletion

[Condition , name of chromosomes , cont_start_bin , cont_end_bin , mean of continue bin]

['deletion', 'chr13', 20500001, 31000000, -1.991163235452375]

['deletion', 'chr13', 36500001, 47000000, -1.9362880727974903]

['deletion', 'chr13', 48500001, 80500000, -2.1396014496846867]

['deletion', 'chr13', 90000001, 115500000, -2.204397601871069]

Plotting

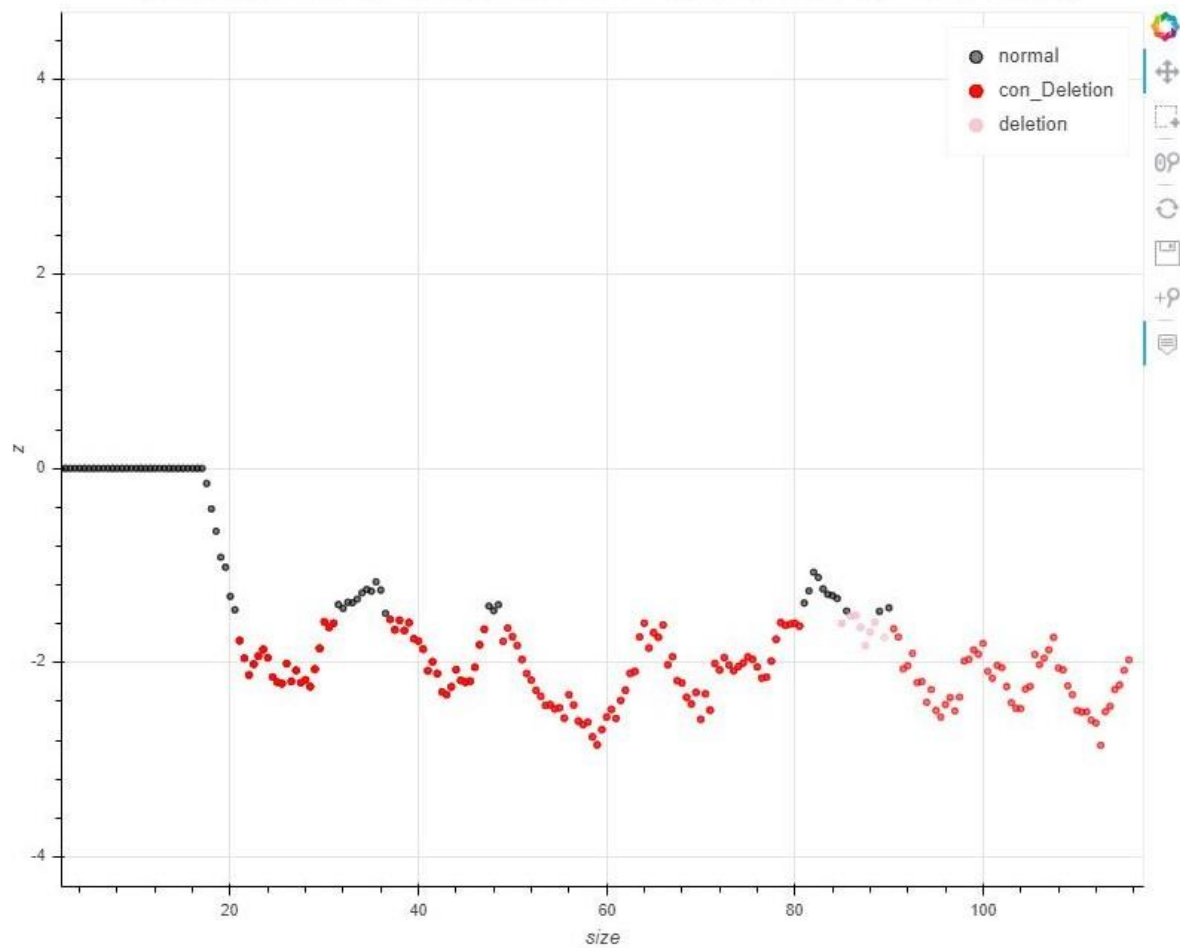
- In the plotting file is saved in html
- We draw interactive plots used bokeh library. In the library we have tools to zoom_in , zoom_out, save, hover tool etc.
- All chromosomes have different graph for (ARC ,reads , z) v/s size of the bin
- z column is show in black color
- duplication show in blue color
- continuous duplication show in purple color
- deletion show in pink color
- continuous deletion show in red color

example :-

chr1 chr2 chr3 chr4 chr5 chr6 chr7 chr8 chr9 chr10 chr11 chr12 chr13
chr14 chr15 chr16 chr17 chr18 chr19 chr20 chr21 chr22 chrX chrY

reads ARC z


relation between z and size of bin



Arguments :-

- **-N(--normalization_file_location)** :- Files location of all normal files.
- **-M(--normalization_method)** :- Which method we used 'gc' or 'reads'.
- **-R(--reference_data_base)** :- reference database of normal samples (output of previous programme)
- **-Z (--z_score_data)** :- output file have zscore data
- **-E(--width_size)** :- difference between start bin and end bin in kb

Example -



	space	start	end	width	reads	gc	map	valid	ideal	cor.gc	cor.map	copy
2	chr1	1	500000	500000	530	-1	0.178026	FALSE	FALSE	0	0	0
3	chr1	500001	1000000	500000	4189	-1	0.581076	FALSE	FALSE	0	0	0
4	chr1	1000001	1500000	500000	5925	0.598332	0.86763	TRUE	FALSE	0	0	0
5	chr1	1500001	2000000	500000	6109	0.539498	0.815132	TRUE	FALSE	0.966146	0.999399	-0.00087
6	chr1	2000001	2500000	500000	5864	0.594508	0.940196	TRUE	TRUE	0	0	0
7	chr1	2500001	3000000	500000	5799	-1	0.794682	FALSE	FALSE	0	0	0
8	chr1	3000001	3500000	500000	6156	0.584572	0.97286	TRUE	TRUE	1.008302	0.989855	-0.01471
9												

- **-W(--window_size)** :- how many window can be rolling
- **-S(--min_cnv_size)** :- minimum size for check how many continue bin give deletion and duplication.
- **-O(--output_file)** :- final file to print continue bin for deletion and duplication
- **-p(--plotting_file)** :- where plots can be show in html

Script Name :- cnv_detection_plotting.py

How to run this programme??

```
(base) C:\Users\EDGC_IN_01>python cnv_detection_plotting.py -N C:/Users/EDGC_IN_01/ON19020508.LP19020010.Normalization.txt -M gc -R C:/Users/EDGC_IN_01/gcf.txt  
-Z tt.txt -W 10 -S 10 -O tc.txt -P plotting.html
```

Result :-

- Print the output file which tell continue deletion and duplication.
- Plotting of all chromosomes.
- Plotting file



plotting.html