GIbPSs 1.0 Manual Part 10: Documentation of depth_analyzer_09.0.pl

5/22/2015 Andreas Hapke,
Institute of Anthropology, Johannes Gutenberg University Mainz, Germany, ahapke2@gmail.com

A GBS library can contain repetitive elements or fragments of repetitive elements. Usually, **indloc** and **poploc** will assign sequences of different copies of a repetitive element to one single locus. Such pseudo loci are not useful for genotyping. Extreme sequencing depth is one clue to the identification of putative repetitive elements. It is however not straightforward to determine extreme depth because a GBS library contains restriction fragments of different length and because the PCR amplification enriches short fragments. Moreover, sequencing depth varies between individuals and not all loci are identified in all individuals.

depth_analyzer compares the sequencing depth of loci within and between individuals and determines variables related to depth. You can use the results to define criteria that identify loci of extreme depth. You can then use **data_selector**, apply these criteria and exclude the respective loci.

Indel variants are a second kind of pseudo loci that may exist in your database. Some loci may have alleles that differ by indel variation. Reads of these alleles can have different lengths and must be properly aligned before a correct pairwise distance can be calculated. **indloc** and **poploc** ignore this problem and assign the respective reads to different loci. These pseudo loci are indel variants of each other. The program **indel_checker** identifies clusters of loci that could be indel variants of each other with the aid of the program USEARCH. Most clusters identified by **indel_checker** will contain a single locus. Some clusters will contain several loci that could be indel variants of each other and are conservatively treated as such.

depth analyzer automatically analyzes the output of **indel_checker** when it finds it and performs a depth analysis based on clusters instead of loci. That means it treats all loci in a cluster as one locus with a common total depth.

Requirements before you use depth_analyzer:

You must run **fdm**, **indloc**, **poploc** and **indpoploc**. You should also run **indel_checker** to identify clusters of loci that could be indel variants of each other.

Usage

depth_analyzer needs a text file with sequence length ranges. The program can analyze all loci as one dataset or analyze loci with different lengths separately. You must define at least one sequence length range. If you define several ranges, please make sure that they do not overlap. The text file must contain two integers separated by a TAB in each line. The two values are the lower and upper bound of one sequence length range. The lower bound may be zero or a positive integer; the upper bound must be a positive integer. Example:

```
32    80
81    100
```

To start the program, call it with the name of the text file as single argument. Examples:

```
depth_analyzer_09.0.pl user_files/length_ranges.txt
```
or
```
perl depth_analyzer_09.0.pl user_files/length_ranges.txt
```

The program reads in the text file. Next, it checks if a selection of data exists. If so, it analyzes the selected subset of data. If not, it analyzes the whole database. The program then searches for output from **indel_checker**. If it finds it, it performs an analysis based on clusters. If not, it performs an analysis based on loci. Clusters may contain several loci of different lengths. **depth_analyzer** treats the length of the longest locus in a cluster as the cluster sequence length and assigns all loci in that cluster to the corresponding sequence length range.

The program produces outfiles in a directory `depth_analysis` within the main database directory. It produces an error message and stops execution when this directory already exists. Rename or delete it before you use **depth_analyzer** a second time.

Split loci

It can happen that **poploc** assigns different alleles of one individual locus to different population loci, i.e. splits up an individual locus. **depth_analyzer** excludes split loci from the analysis. They do not appear in any outfile out of **depth_analyzer**.

Algorithm overview

The program performs separate analyses for the defined sequence length ranges.

It starts by comparing the depths of loci/clusters within individuals: The program looks up the total read depth of each individual locus stored in the database. This depth includes all discarded and retained reads that **indloc** has assigned to the locus. If several individual loci belong to the same cluster, it treats them as one locus and uses their total depth. For a locus/cluster with a given depth d_i, the program then determines the percentage of loci/clusters in this individual with a depth greater d_i. I call this variable a “depth percentile” (*dep_perc*). In order to make sequencing depths comparable between individuals, the program calculates a scaled depth (*scaldep*) for each locus/cluster in each individual it has been observed in by interval scale variable ranging:

$scaldep_i = (d_i - d_{min}) / (d_{max} - d_{min})$

Finally, the algorithm calculates the minimum, median, and maximum of *dep_perc* and *scaldep* for each locus/cluster across all individuals with available data. I chose the median here because it is less influenced by outlier individuals with extremely great or small sequencing depth than the arithmetic mean.

Outfiles

The program writes outfiles in directory `depth_analysis` within the main database directory. It produces separate outfiles for each sequence length range. The filenames start with the minimum and maximum sequence length. For each range, the program produces one outfile for each individual and one outfile across all individuals, e.g.: `32_80_da_ind_01.txt`, `32_80_da_allind.txt`. All files contain tab-delimited tables with header line.

`32_80_da_ind_01.txt`

For each occurring locus/cluster depth (*dep*) in this individual, this file contains the corresponding scaled depth (*scaldep*) and depth percentile (*dep_perc*). Data are based on loci/clusters in the length range 32 – 80.

`32_80_da_allind.txt`

This file contains the minimum, median, and maximum of *scaldep* and *dep_perc* across all individuals with available data for each locus or cluster in the range 32-80. The first column contains the cluster number identified by **indel_checker** when the analysis was based on clusters or the poplocID when the analysis was based on loci. The column header is “clustNo” or “poplocID” respectively.

You can use your favorite spreadsheet program to plot the data in these outfiles:

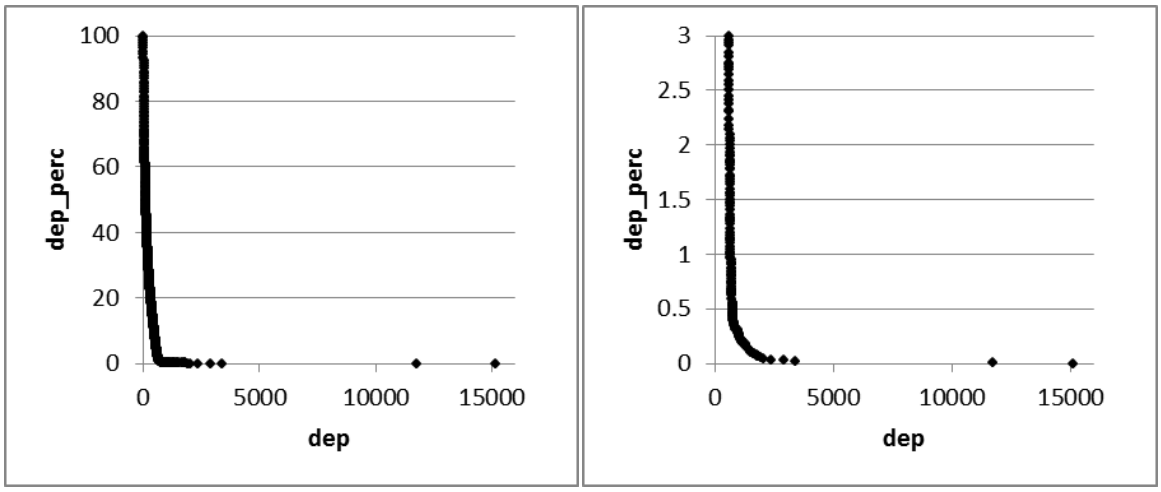

Fig.1: Example plot *dep_perc* and *dep* in one individual from file 32_80_da_ind_01.txt

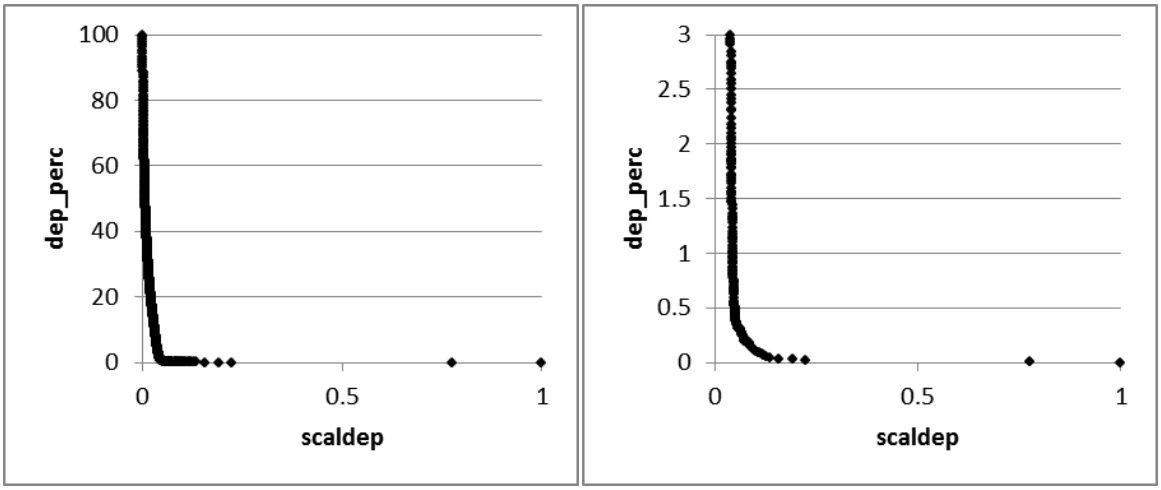

Fig.2: Example plot *dep_perc* and *scaldep* in one individual from file 32_80_da_ind_01.txt

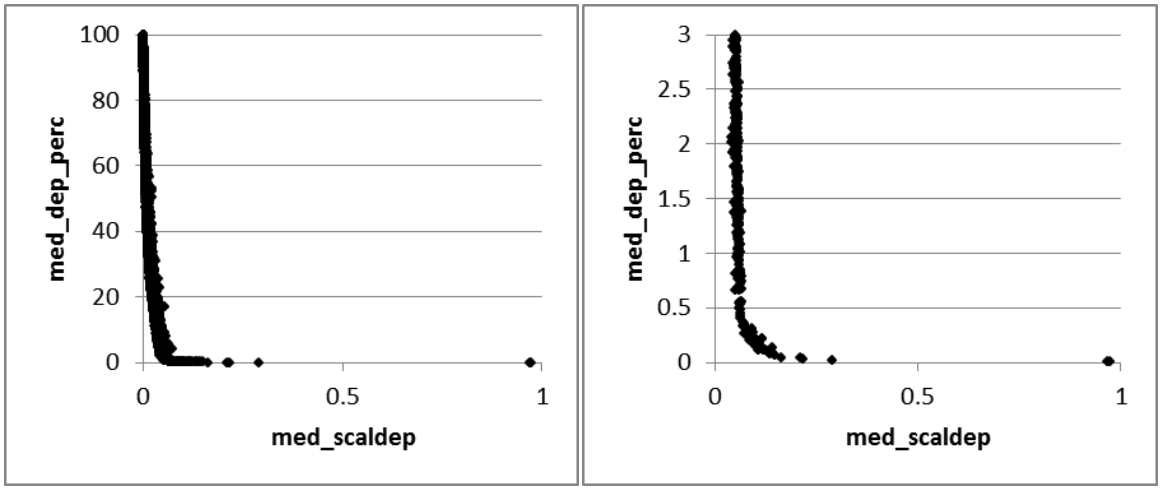

Fig.3: Example plot median of *dep_perc* and *scaldep* across individuals from file 32_80_da_allind.txt

In **data_selector**, you can define a cutoff value for each sequence length range that you analyzed with **depth_analyzer**. Available variables are *min_dep_perc*, *med_dep_perc*, *max_dep_perc*, *min_scaldep*, *med_scaldep*, and *max_scaldep*. Based on figure 3, you could, for instance, select a minimum of 0.5 for *med_dep_perc*, i.e. exclude all loci with *med_dep_perc* < 0.5.

data_selector uses the final outfile out of **depth_analyzer**:

`da_all_loc.txt`

This file contains the poplocID, sequence length and the minimum, median, and maximum of *dep_perc* and *scaldep* for all analyzed loci (poplocs). The minimum, median and maximum values for a specific poplocID are those calculated for this locus when the analysis was based on loci or those calculated for the cluster the locus belongs to when the analysis was based on clusters.

Infiles

The program reads the following infiles. Some of them are optional. The program produces an error message and stops execution when it cannot open a file.

Infile	produced by
`text file with sequence length ranges`	user
`export/sel_ind.txt`	data_selector
`export/sel_loc.txt`	data_selector
`export/sel_gt.txt`	data_selector
`individuals.txt`	indloc
`*_loci.txt`	indloc
`poploc.txt`	poploc
`split_loci.txt`	indpoploc
`*_indpoploc.txt`	indpoploc
`indelcheck/clusters.txt`	indel_checker

*_ stands for an individual ID