DirectRepeateR Tutorial

Megan Copeland, Heath Blackmon

Last updated: 2024-09-23

Introduction

The `DirectRepeateR` package is designed to identify, condense, and visualize direct repeat sequences within genome assemblies. This vignette provides a comprehensive guide to using the core functions of the package, including `GetRepeats` for repeat detection and `PlotRepeats` for visualization.

Installing DirectRepeateR

You can install the most recent version of `DirectRepeateR` from GitHub using the `devtools` package:

```
library(devtools)
install_github("coleoguy/DirectRepeateR",
build_vignettes = TRUE)
library(DirectRepeateR)
```

GetRepeats Function

The `GetRepeats` function searches for and stores the locations of direct repeats within genome assemblies provided in FASTA format. Below is a detailed explanation of its parameters and output.

Example Usage

```
results <- GetRepeats(file = "PATH/TO/genome.fa", query_length = 25, maxdist = 20000, minlength = 50)
```

Parameters

- **file**: A character string specifying the path to the genome assembly file in FASTA format. The function uses the `seqinr` package to read the FASTA file. Recommended to use a FASTA file that has been filtered to remove all small, unplaced contigs/scaffolds.
- **query__length**: An integer specifying the length of the query sequence used to search for direct repeats. The genome sequence is segmented into subsequences of this length to identify repeated patterns. Default is 25 if not specified.
- **maxdist**: An integer specifying the maximum distance between repeat copies within which the function will search for direct repeats. This sets a limit on how far apart repeat sequences can be to be considered related. Default is 20000 if not specified.
- **minlength**: An integer specifying the minimum length of sequences to be considered as direct repeats. Only repeats of this length or longer will be detected. Must be a multiple of the query length. Default is 25 if not specified.

Output

The function produces a data frame and a CSV file with five columns:

1. **Chromosome**: Sequence name of where the repeat is located.

2. **Start_Position**: Start position of the first occurrence of the repeat.
3. **End_Position**: End position of the first occurrence of the repeat.
4. **Match_Position**: Start position of the second occurrence of the repeat (the repeat copy).
5. **Match_End_Position**: End position of the second occurrence of the repeat (the repeat copy).

- **condensed_results.csv**: Contains condensed results where contiguous repeat sequences are merged. This function creates a results file in the working directory and saves this output in that location.

ConvertToGFF Function

The `ConvertToGFF` function is designed to convert the output from `GetRepeats`function into a GFF (General Feature Format) file. The function processes each row of the final_output data frame, which contains location information for the detected repeat sequences, and generates three corresponding GFF entries: one for the full length of the repeat, representing the region from the start of the first copy to the end of the second copy; one for the first detected repeat copy; and one for the second detected repeat copy. The gff file is written out as a text file and saved in a results folder in the working directory.

Example Usage

```r
gff_output <- ConvertToGFF(data = results)
```

PlotRepeats Function

The `PlotRepeats` function visualizes the density of direct repeats across chromosomes using a sliding window approach. This section explains how to use it and customize the visualization.

Example Usage

```r
PlotRepeats(data = results, window_size = 200000, step_size = 200000)
```

Parameters

- **data**: A data frame generated by the `GetRepeats` function, containing details about the locations of direct repeats within a genome assembly.
- **window_size**: An integer specifying the size of the sliding window used to count the number of repeats. Default is 200,000 base pairs.
- **step_size**: An integer specifying the step size for the sliding window. This defines the interval at which the window moves along the chromosome. Default is 200,000 base pairs.

Output

The function generates and displays line plots for each chromosome, showing the number of repeats within each window along the chromosome's length.

Conclusion

The `DirectRepeateR` package provides powerful tools for identifying and visualizing direct repeats in genomic data. By following this vignette, you should be able to use the each of the functions to analyze direct repeats within genome assemblies.