*Journal*, YYYY, 0–0

doi: 10.1093/*Journal*/xxxxx

Advance Access Publication Date: DD Month YYYY

Manuscript Category

|  |
| --- |
| Genome analysis  Studying the genome with plotZcurve  Aura Zelco1,\*  1*Department of Biology, Box 118, 221 00, Lund University, Sweden.*  \*To whom correspondence should be addressed.  Submitted on 2022/03/21  Abstract  **Summary**: Although the genome of different organisms has been studied in the past years, to this day genome analysis is still extremely relevant, from *de novo* genome assemblies to population genetics studies. Thus, new tools which allow for a deeper analysis of the genome can improve such analysis.  The Z-curve is a tridimensional curve representing three different chemical properties of a nucleotide sequence, and has been used in a numerous applications such as detection of horizontal gene transfer, genome visualization and comparison, identification of protein-coding genes, and more. However, currently there is no proper tool to generate such plots.  **Results**: Here, we present plotZcurve, a combined Python and R software which can generate Z-curve plots for any sequence in FASTA format, as well as W/S plots and GC content reports, which can be used in further analysis of the sequence.  **Availability**: plotZcurve is freely available on [Github](https://github.com/aurazelco/plotZcurve), both as Python and R scripts and as a flask web interface.  **Contact**: aura.zelco@gmail.com |

# Introduction

Although many years have passed since the first sequencing of genomes, genome analysis is still relevant to this day. From *de novo* assemblies of newly discovered species to variant calling and population genetics, the genome continues to be a source of new data and new scientific discoveries.

Among the different methods, the Z-curve method was proposed. The Z-curve is a geometrical approach to genome visualization: the three dimensions correspond to three different chemical properties of the nucleotides, which can be used to divide them into sub-groups:

1. puRine/pYrimidine (R/Y): if the nucleotides have a purine or a pyramid ring in their structure (R: A, G; Y: C, T)
2. aMino/KEto nucleotides (M/K): if the bases have an amino or a keto group (M: A, C; K: G, T)
3. Weak/Strong hydrogen-bonds (W/S): which type of H-bonds they form in the double-helix structure (W: A, T; S: C, G)

Despite it having been used over the years (for example Guo et al., 2014), currently, there is no available tool for displaying the Z-curve. The group that first proposed this approach has published a website (Zhang et al., 2003), but at the time of writing is not available.

Thus, in this paper, we present [plotZcurve software](https://github.com/aurazelco/plotZcurve) (v1.0.0), which generates Z-curve plots from FASTA sequences.

# Methods

## Input files

plotZcurve takes as input any nucleotide sequence in FASTA format; in this paper, the samples data are genomic sequences for *E. coli* and Zika virus, obtained from NCBI (links: [E. coli](https://www.ncbi.nlm.nih.gov/assembly/GCF_000005845.2); [Zika virus](https://www.ncbi.nlm.nih.gov/nuccore/NC_012532.1)).

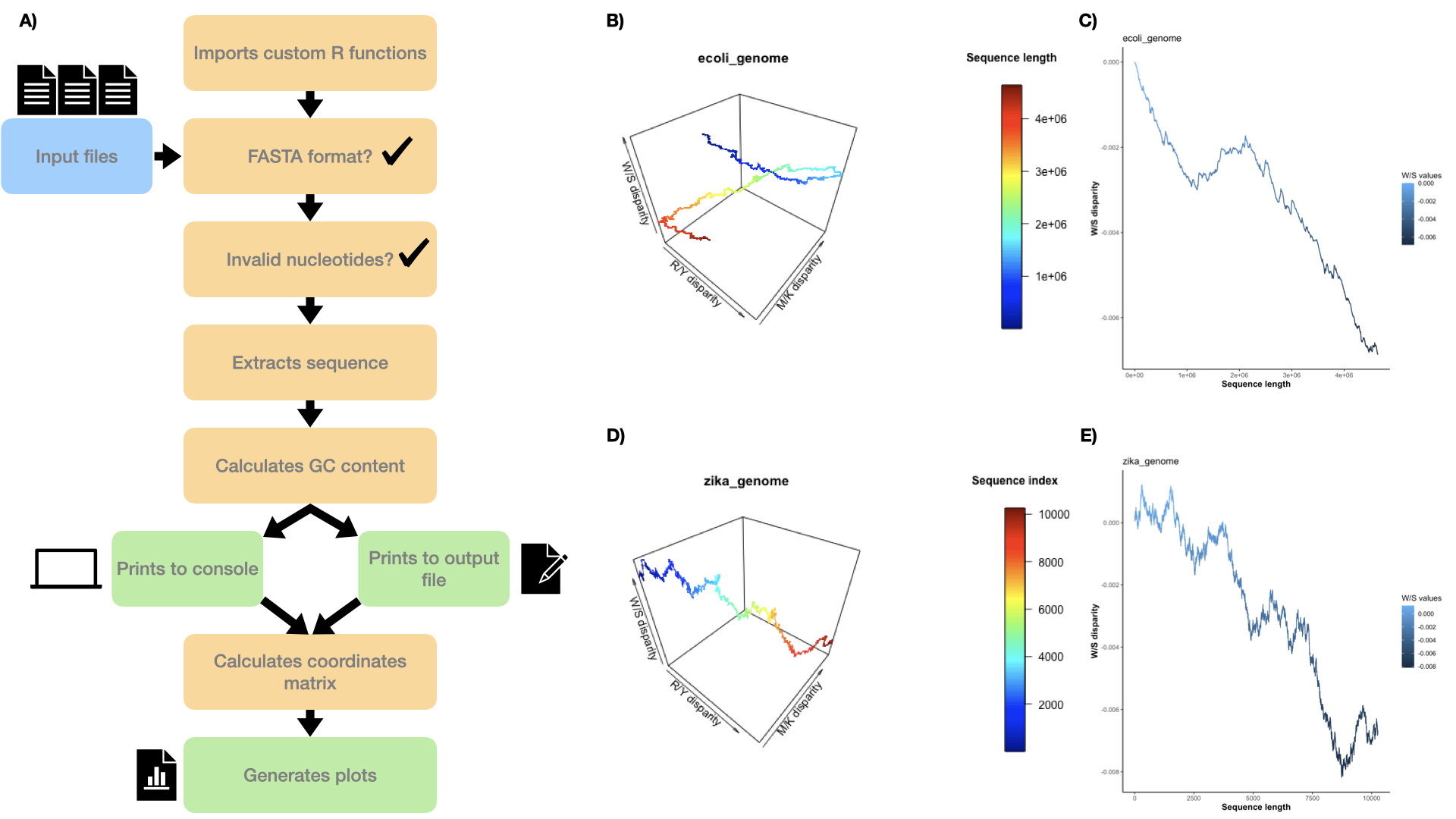
## Calculations for the Z-curve

In this paper, we based our calculations on the geometrical approach used in previous papers (reviewed in Zhang & Zhang, 2014).

Briefly, if we consider the Z-curve, each point *P* can be described by three coordinates - X, Y, and Z - which are calculated based on a tetrahedron space and by the frequencies of the nucleotides (*a, c, g,* and *t*) (for details: Zhang & Zhang, 2014).

**Fig. 1. plotZcurve workflow and examples.** A) plotZcurve workflow diagram. B-E) Examples of Z-curve plots (B, D) and W/S plots (C, E) as generated by plotZcurve from the genomes of *E. coli* (B, C) and Zika virus (D, E). The legend in the Z-curve plots (B, D) represents the sequence length, so the user can follow the genome from start to end. In C, E the legend represents the values of W/S coordinates along the sequence length.

Taking this and the rotations of the tetrahedron into consideration, we obtain the following transformation matrix:



where X, Y, and Z are the coordinates of *P*; the square root of 3 divided by 4 is a coefficient due to the geometry of the tetrahedron; the second matrix is based on the rotations along the diagonals of the tetrahedron, and the last vector is the cumulative frequencies of the nucleotides until the position of *P* in the sequence.

This matrix is at the base of the calculations in plotZcurve to map the coordinates of each point *P* in the Z-curve plot.

## plotZcurve pipeline - command-line version

plotZcurve is a Python script (3.8.10) that imports R custom functions (4.1.1); the modules and libraries used are numpy (1.22.3), pandas (1.4.1), rpy2 (3.1.0), plot3D (1.4), and ggplot2 (3.3.5).

Below are listed the flags which can be used with plotZcurve.py when run from the command line:

* -i: input sequence file(s) in FASTA format
* -f: output format of the plots, e.g. png pdf jpeg (default png)
* -o: output path where to save the plots (default: working directory)
* -s: path to R scripts (if not in working directory)
* -gc: if used, prints the GC content to an output file instead of the console
* -out\_gc: output path where to save the GC output file
* -ws: if used, additional W/S plots will also be generated and saved

The pipeline for plotZcurve is briefly described below and in **Fig. 1A**. After running the commands, the Python script imports the R functions as modules. Then, for each input file, it checks if the file is in FASTA format; if so, it stores the sequence in a string and checks for any non-valid nucleotides. Then, it calculates the GC content, and either prints it to the console or writes it to an output file, followed by the generation of the coordinates matrix. Once the calculations are computed for the whole sequence, it generates the Z-curve plots and additionally the W/S plots if required by the users. Once all files are processed, or if any error is raised, the program exits.

## plotZcurve pipeline - flask web interface

plotZcurve also comes as a flask developer web interface (versions: flask 2.0.3, Werkzeug: 2.0.3). This app is more user-friendly, but it is more restricted in terms of user options.

The flask app is run by the terminal, and it is executed on a local host; therefore, before running the user has to manually specify absolute paths where the input files and the R scripts are, otherwise the flask app will not run.

For each input file, the flask app will print to the screen the GC content, and the Z-curve and W/S plots, side by side.

# Results

We have used the whole genome for *E. coli* and Zika virus to show the results of plotZcurve.

The plots generated from plotZcurve can be found in **Fig. 1B-E**. The first two plots represent the Z-curve for *E. coli* and Zika virus respectively here saved as png files.

As seen from the Z-curve plots (**Fig. 1B, D**), the two genomes seem to have many differences. Firstly, the sequence length varies between the two organisms, as could be expected when comparing bacterial and viral genomes. Secondly, *E. coli* seems to have quite similar conditions for R/Y and M/K disparity at the beginning and end of the sequence, but not in the middle, causing the Z-curve in a spiral-like shape. The Zika genome instead seems to be quite consistent in the M/K and R/Y ratios, therefore we see a diagonal line in the XY space. Lastly, both genomes seem to tend to increase W/S disparity towards the end of the sequence, as seen also from **Fig. 1C, E**. This indicates an increasing presence of GC content throughout the sequence, which might not be obvious by looking at the overall GC content alone, which for both species was around 50% (not shown, but present on [Github](https://github.com/aurazelco/plotZcurve)).

The same plots shown in the figure can be displayed also in the flask web interface. However, it is possible to download the plots only as png files, and the GC content is only printed to the screen.

Thus, if the user wants a quick overview, the flask app would be the best way to visualize the plots. If instead, the user wants to save the plots in a specific format and/or save the GC content outputs, the command line is recommended.

# Discussion

plotZcurve displays the Z-curve and W/S plots, which can be useful for genome analysis pipelines. Although there is controversy in the field on the usefulness of the Z-curve and its visualization (Elhaik et al., 2010; Zhang, 2011), we still believe that plotZcurve is a valid tool that can assist scientists in the genome analysis workflows.

An example of use could be when the genome has been *de novo* assembled, and the user would like to compare it to similar organisms to see if and what differences are present. Moreover, it can also plot the W/S disparity alone, which could be used to compare the GC/AT disparity in multiple genomes/samples/individuals. This feature could be quite interesting since the GC content is one of the parameters most used in genome analysis and as mentioned in the Results section, overall GC content can be very different from GC distribution.

However, this tool comes with limitations. Since the main script is a combination of Python and R, the right version of rpy2 (the module which handles the conversion between Python and R objects) must be used. If not, the tool will encounter errors, and will not run.

Additionally, since the coordinates are calculated for each position in the sequence, the software is quite affected by the length of the sequence. From the examples, the Zika genome runs quite fast, with all different combinations of flags; if the *E. coli* is used as input, alone or with other files, the software will take more time to produce the plots. The flask app is also affected by this.

As mentioned in the results, the flask interface is more user-friendly, but it is quite limited. This is mainly because the flask app is in developer mode, which poses a series of security issues when files have to be up- or downloaded.

As for future developments, a potential fix to the slower runtime could be adding a sliding window, which might speed up the calculations. Another potential development would be to build a proper website so that the limitations of the flask app are removed for a better and more friendly user experience.

# Conclusions

plotZcurve is a novel tool for plotting Z-curves and W/S disparity, and even if with limitations, can still be quite useful in the genome or sequence analysis pipeline.

Acknowledgments

We thank the reviewers for their insightful feedback during the manuscript revision and Lund University for their financial support.

Funding

*Conflict of Interest:* none declared.

Bibliography

Elhaik, E., Graur, D., & Josić, K. (2010). ’Genome order index’ should not be used for defining compositional constraints in nucleotide sequences - a case study of the Z-curve. *Biology Direct*.

Guo, F.-B., Lin, Y., & Chen, L. (2014). Recognition of Protein-coding Genes Based on Z-curve Algorithms. *Current Genomics*. https://doi.org/10.2174/1389202915999140328162724

Zhang, C. T., Zhang, R., & Ou, H. Y. (2003). The Z curve database: a graphic representation of genome sequences. *Bioinformatics*, *19*(5), 593-599. https://doi.org/10.1093/bioinformatics/btg041

Zhang, R. (2011). A rebuttal to the comments on the genome order index and the Z-curve. *Biol Direct*, *6*, 10. https://doi.org/10.1186/1745-6150-6-10

Zhang, R., & Zhang, C. (2014). A Brief Review: The Z-curve Theory and its Application in Genome Analysis. *Current Genomics*, *15*, 78-94. https://doi.org/10.2174/1389202915999140328162433