# Sequence Bioinformatics Report: Comparison of two whole-genome alignment tools

**Nantia Leonidou, Florian Riedl, Alexander Röhl**

**Whole-genome alignment has been of main interest the last few years, as it provides valuable information about evolutionary relationships between two or more genomes enabling the better understanding of their biological background. However, over the years multiple alignment tools have been published based on different approaches making it harder for the scientists to choose the most appropriate one. Our project focuses on comparing two different whole-genome alignment tools and study the effect of their different methods on aligning two reference genomes against one assembled one: Mauve and MUMmer.**

## Introduction

Since the sequencing of the 1.8 Mb-sized genome of the bacterium Haemophilus influenzae in 1995, more and more whole genomes become available to scientists. However, genome evolution can become extremely complicated as re-combinations can occur causing large-scale changes in genomes, such as gene loss, duplication and rearrangement. Thus, scientists have became highly interested in aligning each genome against other available genomes to understand better its evolutionary and biological background. This led to the formulation of the whole-genome alignment (WGA) problem. Whole-genome alignment (WGA) is the prediction of evolutionary relationships at the nucleotide level between two or more genomes. It combines aspects of both collinear sequence alignment and gene orthology prediction, and is typically more challenging to address than either of these tasks due to the size and complexity of whole genomes. For each segment of the given genome WGA aims in giving information about where its corresponding segments are located in other genomes (references). In other words, WGA is predicting evolutionary relationships at nucleotide level based on collinear sequence alignment and prediction of orthologous genes. As it may be concluded, due to the size and complexity of genomes, WGA is more complicated than either tasks. (1)

Our project focuses on conducting such a whole-genome alignment using two different tools, visualizing and comparing the results. Our main focus is to compare how these two whole-genome aligners performed in terms of predicted insertions, deletionsand mismatches by aligning one assembled genome against two reference sequences, both derived from *Escherichia coli* (*E. coli* ). More detailed description of our used dataset can be found in the Materials and Methods section. We decided to use Mauve (2) and MUMmer (3), more specifically its main pipeline named nucmer. The reason is that the first tool has an already built-in visualization tool, thus we did not need to create plots from scratch, and both tools rely on different approaches making it easier and more reasonable to compare them in terms of deletions, insertions, mismatches and translocations. For this particular analysis we implemented our own tool, which you can find in the hand-in file.

**Mauve: Multiple Alignment of Conserved Genomic Sequence With Rearrangements.** Mauve is a multiple-genome alignment program and uses the anchored alignment technique to construct a WGA. Its approach includes five basic steps as described in the original paper: (1) Search for local sequences shared by two more genomes and occur in once in those genomes, the so-called Multiple Maximal Unique Matches (multiMUMs), (2) Construction of a phylogenetic guide tree using the multi-MUMs as a distance metric and Neighbor Joining (4), (3) Selection of a subset of the local alignments to use as anchors which are then divided into locally collinear blocks (LCB), (4) Conduct recursive anchoring to identify additional alignment anchors, (5) Perform a progressive alignment of each LCB ClustalW (5) or MUSCLE (6) progressive global

alignment algorithms (in our case Muscle 3.6 version was used) (2, 7).

The first output file is the complete genome alignment created by Mauve in the so-called extended multi-Fasta (XMFA) file format. This format enables the storing of several collinear sub-alignments separated by "=". Each such sub-alignment contains one FastA format sequence entry per genome where the entry's line started with ">" gives the orientation and location of the strand in the genome of the sequence in the alignment. Additionally, a .backbone file is created, which, depending on the number of aligned sequences, consists of several columns. For instance, if two sequences become aligned, the .backbone file will contain four columns named: seq0_leftend, seq0_rightend, seq1_leftend, seq1_rightend. Then, each subsequent line corresponds to a segment of DNA conserved among two or more genomes. For instance, the i-th line would correspond to the i-th segment between the coordinates seq0_leftend and seq0_rightend in the first genome, which is homologous to the section between the coordinates seq1_leftend and seq1_rightend of the second genome. The third output file is a .bbcols file which stores the alignment columns that are predicted to be part of larger conserved segments among each group of genomes. The first column contains the ID, the second one stores the alignment column where a conserved region begins with, the third is the length of this particular conserved region and the firth records the genome IDs taking part in the conserved block. Also, separately and manually a file containing all SNPs or Gaps can be created from the GUI, too.

**MUMmer (nucmer).** MUMmer, with its main pipeline nucmer, is a software for whole genome alignments. As the name suggests MUMmer uses Maximal Unique Matches (MUMs). At first the MUMs are detected using a suffix tree or since MUMmer4 a suffix array. In the next step the MUMs are clustered based on the distance to each other. At last, a banded Smith-Waterman is used to connect the MUMs in each cluster by aligning the bases between them. In the same step the first and the last MUM in the cluster are extended in the outgoing direction. The MUMmer (nucmer) output is a delta file which can be further processed with tools from the mummer package. MUMmer4 is the first version that can output SAM files(3).

## Materials and Methods

**Used Datasets.** All used sequences are derived from *E. coli*, which is a gram-negative bacterium commonly found in the lower intestine of endotherms. It can exist in pairs or alone and has been firstly isolated by Theodor Escherich. *E. coli* is the main cause of common bacterial diseases like urinary tract infection, traveler's diarrhea, bacteremia and pneumonia. Although most of its types are harmless, some others can cause diarrhea while eating contaminated food or drinking polluted water (8). We used an assembled *E. coli* genome from the NCBI Assembly database (9) with the ID **ASM972046v1** (https://www.ncbi.nlm.nih.gov/assembly/GCF_009720465.1/). It has been submitted from The University of Texas Health Science Center at Houston on 29.11.2019 and assembled using the Celera Assembler v. 1.8. It has a genome coverage of $715x$ and was sequenced using the Oxford Nanopore GridION technology. Furthermore, the genome has a total sequence length of $5,201,916$ bases as well 1 chromosome and five plasmids. Moreover, we used two closely related genomes of *E. coli* taken from the NCBI Nucleotide database under the references **NC_002695.2** and **NC_004431.1** as reference sequences. The first one is known as the *E. coli* O157:H7 Sakai strain (genome length: $5,498,578$ base-pairs), which is a pathogenic strain resulting to abdominal cramps and bloody diarrhea and is the main cause of acute kidney failures in children. The second one is the *E. coli* CFT073 strain which is mainly responsible for urinary tract infections (10) and has a genome length of $5,231,426$ base-pairs.

**Conducting the WGA.** Before starting with the project we created a workflow (see Figure 1) to

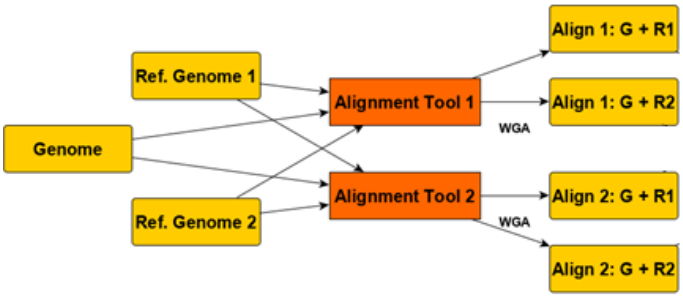


**Fig. 1.** Workflow to conduct whole genome alignment between two references and an assembly

be able to have an overview of our upcoming tasks. These include the alignment of assembly against both references using both tools. After aligning the assembly against the two references, we observed that only the two references are in consistent directions. Thus, we decided to write our own Python script to construct the reverse complement of the assembly to get better insights from the visualization afterwards. The corresponding code can be found in the hand-in file. Then we ran once again both aligning tools using the reversed complement of the assembly. Moreover, we discarded the plasmids from the assembled genome, as they do not belong to our actual assembly.

To conduct the whole genome alignment we used Mauve (2), which is a multiple-genome alignment program. Mauve uses the anchored alignment technique to construct a WGA. Compared to other genome alignment tools, Mauve allows the identification of genome rearrangements by allowing the anchors' order to be rearranged in each genome. More details can be found in the Materials and Method section. Some caveats of this algorithm are that it is mainly applicable to closely related organisms and that alignment of genomes with many segmental duplications is difficult. This is why Darling et al. developed a version of Mauve, called progressive Mauve aligner, which we also used to align our genomes. In more detail, Mauve was ran using the tool's default parameters. For instance, the seed size, which sets the minimum weight of the seed frame used to create matches during the first pass of anchoring the alignment. Setting this value too low is appropriate when aligning divergent genomes however it may reduce sensitivity. In our case, the seed size was set to be 15. Furthermore, only the options "Use seed families" and "Assume colinear genomes" were disabled. The first one prevents from using multi-spaced seed pattern, while the second one is relevant only if it is certain beforehand that there are no rearrangements the genomes to accelerate the process.

The second alignment tool we used was MUMmer. It is build up on a seed and extend concept with MUMs as seed and a adapted Smith-Waterman as extend tool. MUMmer was used through the NUCmer pipeline. For NUCmer the default parameters have been used. The most important (default) parameter is "–mumreference". It states that a match has to be unique in the reference but not mandatorily in the query. Furthermore the default for the maximum distance any match will be extended in a direction in a region with a poor alignment is 200. The minimum seed length in NUCmer is 20.

**Analysis of WGAs.** For the analysis of the two WGAs we created a custom tool using python. More details about running the tool and mre related information can be found in the provided README.md file. The tool accepts both Mauve and MUMmer output formats. All information from Mauve is extracted from the created *.alignment files, while for MUMmer from the *.maf files. In both cases all aligned sequences for all input genomes are sorted into a respective list. Furthermore, all of the alignment's starting positions are saved into another list for each aligned genome. These four lists are once again saved together if the alignment-tool's name, and the IDs of the genomes. This container list, called `sequence information`, is parsed multiple times afterwards.

First of all, for the analysis of the overall alignment performances of the two tools, all aligned sequences were parsed, while at the same time six parameters were counted: Insertions, Deletions, Mismatches, Is_Inverse, Is_Translocated,

Total_Length. The first three parameters are important for the alignment patterns themselves. Insertions are defined as positions in both aligned sequences, where the first sequence possesses a gap, while the second sequence possesses a base. Deletions are the opposite, where a position possesses a base in first sequence and a gap in the second. Mismatches are defined for positions with bases in both sequences at this position, but not of the same kind.
The last three parameters are important for the overall capabilities of the alignment-tools for whole-genome-alignments. Is_Inverse is a boolean parameter, which defines whether the second sequence is the inverse of the first sequence. If this parameter would be true, we could prove the ability of the tool to align even inverse subsequences, which is important in recombinatory genomes in bacteria. In the end there were no such findings in either of the four alignment files. However, this does not implicate, that the tools are not able to align reversed sequences. It just shows that in none of our sequences, even though we used bacterial genomes, no reverse recombinatorial sub-sequences were integrated. The second last parameter Is_Translocated is a boolean parameter as well, which is defined as True if the start position of the second sequence is not in order of its genomes sequence anymore. This means that this fragment was translocated during bacterial recombination to another position in the genome.

The last parameter Total_Length is defined by the length of each aligned sub-sequence. While this is a rather trivial factor, it shows the alignment tools tendency to further extend aligned sequences, even at the cost of aligning subsequences with lower identity, or the tendency to include multiple alignments into one. With these parameters describing the alignment behaviour of both tools we can later make according conclusions for the results.
During the second (optional) parsing of the sequence information all aligned sequences are collected into a single fasta file for each genome. By comparing the fasta files of the same genome between different tool results a more general comparison of alignment behaviours can be observed.

For the third and last parsing run, all aligned sequence lengths are gathered into a tsv-file in order to be used as inputs for a software package for visualizing data, named Circos(11). Circos visualizes data in a circular layout enabling the better exploring of relationships between different positions in two or more sequences.

## Results and Discussion

**Mauve.** To visualize the results obtained from Mauve, we used the built-in viewer, which helped us see the genome rearrangements and the locally aligned blocks at once. In Figure 2 you can see the constructed whole genome alignment obtained using the the reference genome NC_002695 and the assembly. Each sequence is represented by a horizontal panel of blocks and each coloured block is a region of the sequence that aligns to region of the other genome. These are the LCBs and they are connected by vertical lines. If the blocks are below the corresponding genome line means that this particular region is inverted with respect to the other sequence. For instance, you can see the purple block in the second sequence at position $3,000,000$ is the reversed complement of the same block in the first sequence. The same illustration of whole genome alignment but between the second reference and the assembly can be seen in Figure 3. In this Mauve plot one can directly observe more LCBs and thus more vertical lines, compared to the previous figure, where only few but bigger LCBs were seen.

**MUMmer.** The visualization of MUMmer alignments has been done with the tool mummerplot which is included in the MUMmer installation. Figure 4 depicts a dotplot of NC_002695 against the assembly. The diagonal line without gaps indicates a high similarity and no evolutionary events between the two sequences. The small diagonal on the bottom right is due to the fact that the starting point of the reference and the assembly are different. The plot for NC_004431 can be seen in Figure 5. It shows similar results as in

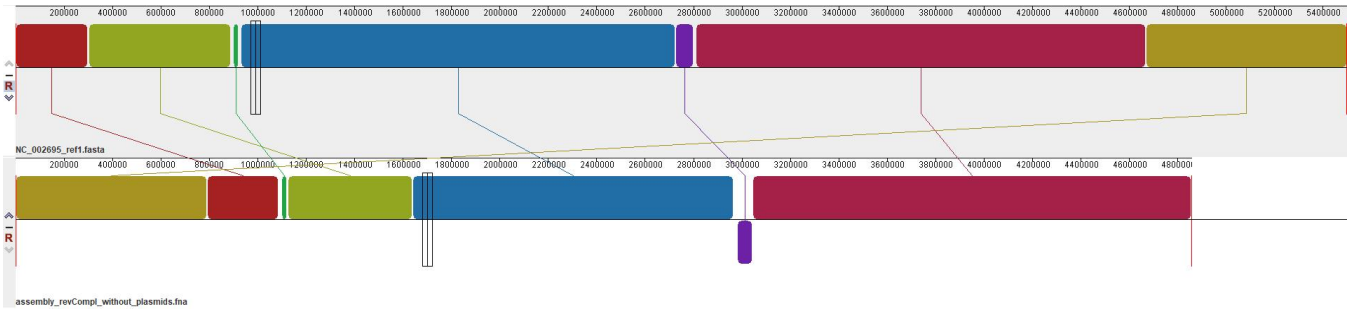


**Fig. 2.** Alignment of the first reference genome and our assembly.

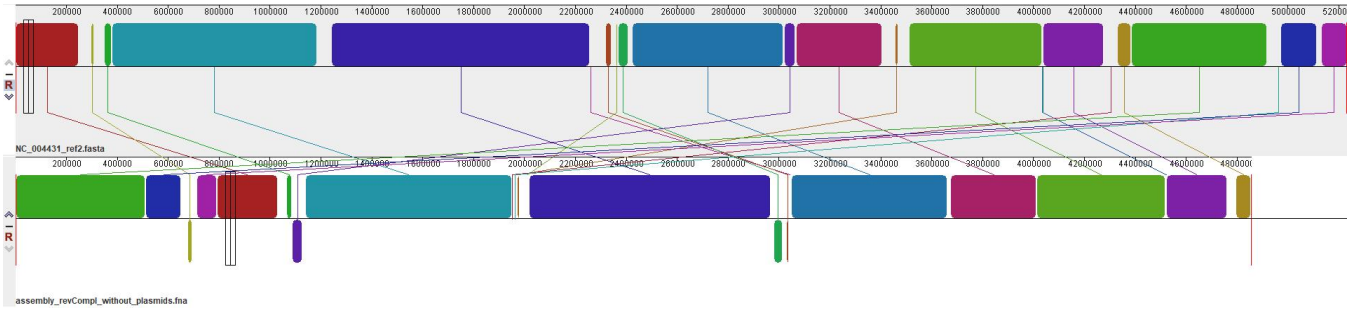


**Fig. 3.** Alignment of the first reference genome and our assembly.

the previous one, although there are some small translocations in this whole-genome alignment.

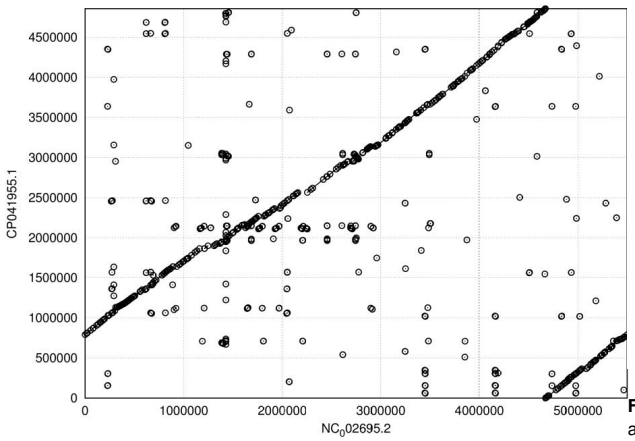


**Fig. 4.** MUMmer alignment of the first reference genome (NC_002695) and the assembly.

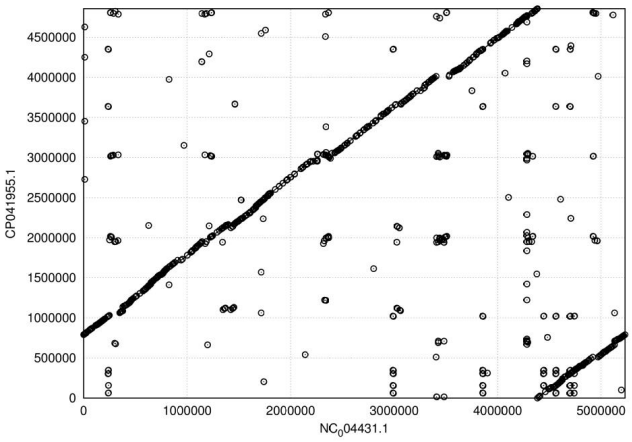


**Fig. 5.** MUMmer alignment of the second reference genome (NC_004431) and the assembly.

**Analysis Tool.** The statistics computed by our analysis tool for all sub-alignments of the four WGA files can be seen in the following tables (Table 1 - 2). When looking at the results the first obvious difference between Mauve and MUMmer is the number of sub-alignments of the WGA. Mauve contains 7 for the first reference genome and 23 for the second. MUMmer contains 623 sub-alignments for the first reference and 723 for the second. As mentioned earlier this hints to Mauve having a tendency to extend alignments over each other what makes it less sensitive to the parts between these sequences.

| Tool | Reference | Number of sub-alignments | Insertions | Deletions | Mismatches |
|---|---|---|---|---|---|
| Mauve | NC_002695 | 7 | 615701 | 1263025 | 91546 |
| Mauve | NC_004431 | 23 | 659951 | 804429 | 120656 |
| MUMmer | NC_002695 | 623 | 3567 | 2822 | 89031 |
| MUMmer | NC_004431 | 733 | 3575 | 3570 | 119701 |

**Table 1. Sum of all sub-alignments for each WGA: Insertions, Deletions, Mismatches**

| Tool | Reference | Number of sub-alignments | Is_Inverse | Is_Translocated | Total_Length |
|---|---|---|---|---|---|
| Mauve | NC_002695 | 7 | 0 | 1 | 6086445 |
| Mauve | NC_004431 | 23 | 0 | 8 | 5454359 |
| MUMmer | NC_002695 | 623 | 0 | 171 | 4482627 |
| MUMmer | NC_004431 | 733 | 0 | 158 | 4326346 |

**Table 2. Sum of all sub-alignments for each WGA: Is_Inverse (Number of booleans equal to 'True'), Is_Translocated (Number of booleans equal to 'True'), Total_Length**

This is probably also the reason why the number of Insertions, Deletions, and Mismatches is much higher for Mauve. On the one hand, this does give a more general overview over the WGA, but makes the result less accurate. On the other hand, MUMmer splits alignments much more frequently making an overview over the WGA harder, but increasing its quality of the individual alignments. The parameter Is_Inverse returned False for all sub-alignments, meaning that there were no reversed fragments in our sequences. In the beginning we hoped to prove for one or both tools to be able to find such reversed fragments and nevertheless align them. But as neither of the two could find any we couldn't show this for either. When observing the number of translocated sequences MUMmer is again more sensitive, returning a significantly higher number of translocations.

Last but not least, we created Circos plots get a better visualization of our results obtained from both Mauve and MUMmer (Fig. 6 and 7). These plots consist of three main parts. The first one is the inner circle, which is divided into multiple segments, showing the aligned blocks coming from both sequences (assembly and reference). The assembly's LCBs are marked with the letter "A", while the LCBs coming from the references with the letter "B". The outer rings are stacked bar plots that represent relative contribution of a cell to row and column totals, while the coloured ribbons connect the corresponding row and column segments. Thus, each ribbon represents a single LCB and is coloured according to the segment it is coming from. From the corresponding figures you can validate the above mentioned observation, that MUMmer creates much more and smaller LCBs compared to Mauve.

## Conclusion

To sum up with, the obtained whole-genome alignment of two reference sequences against an assembly of *E.coli* showed us, that two well-known tools could deliver diverse results. While Mauve is based on finding LCBs creates less collinear blocks, MUMmer focuses on suffix trees and on finding MUMs resulting to significantly more sub-alignments increasing the quality of WGA. However, Mauve reports importantly more insertions and deletions. When it comes to mismatches, both tools are observed to be more sensitive regarding only one reference sequences resulting to a higher number of mismatched nucleotides. Lastly, MUMmer defined more blocks to be translocated compared to Mauve and none of the tools seemed to identify inversed fragments. These results can be nicely recapitulated using our created Circos plots. All in all Mauve seems to give a better overall picture of the whole WGA by dismissing tinier aligned fragments or including multiple into a single sub-alignment. It trades off individual alignment sensitivity compared to MUMmer thus reducing the overall sub-alignment's quality, making it more prone to inaccuracies. However, to be able to draw reliable conclusions about the differences between both tools, more sequences need to be tested coming also from different organisms.

## References


1. Dewey CN (2012) Whole-genome alignment in *Evolutionary Genomics*. (Springer), pp. 237–257.
2. Darling AC, Mau B, Blattner FR, Perna NT (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome research* 14(7):1394–1403.

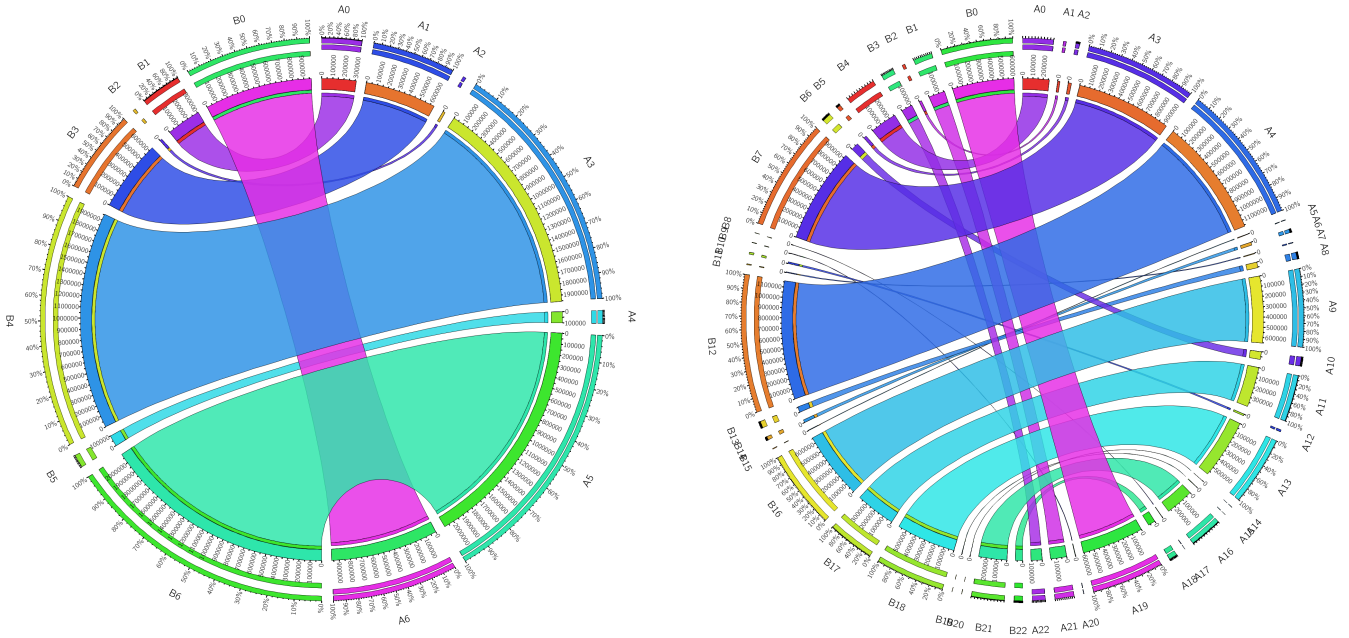

**Fig. 6.** Cyclic alignment of reference sequence NC_002695 in the left figure and NC_004431 in the right figure and the assembly sequence using Mauve
A=reference sequence, B=assembly sequence

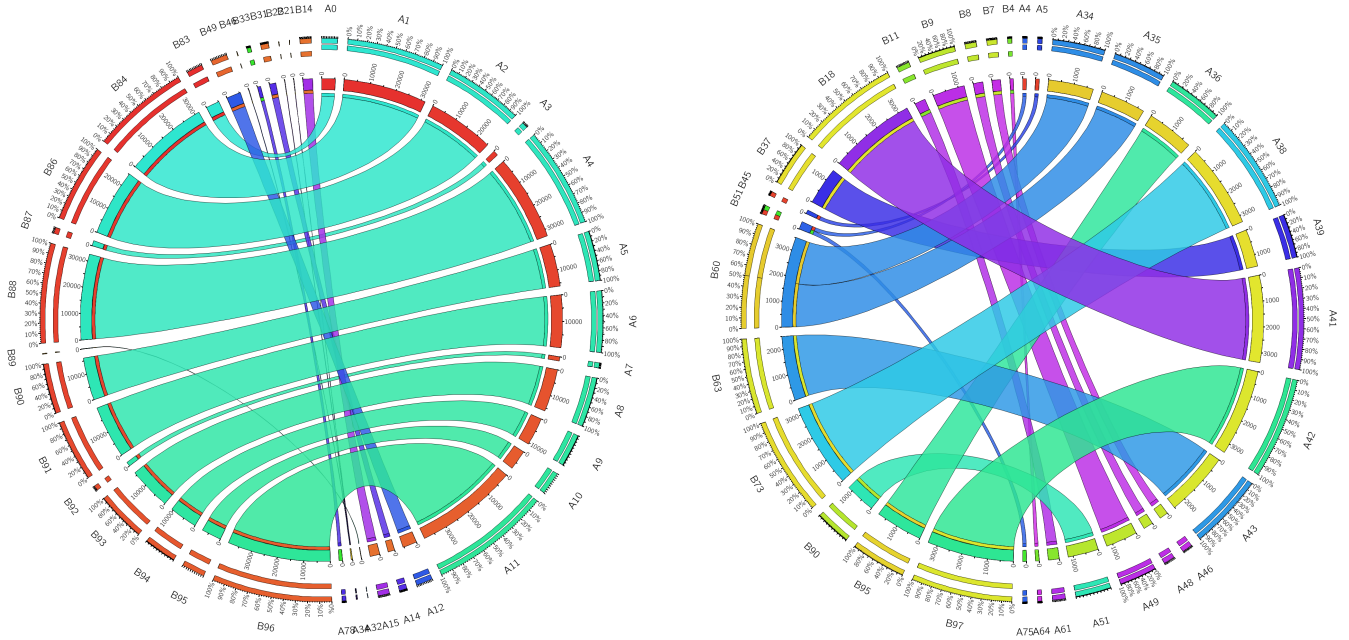

**Fig. 7.** Cyclic alignment of reference sequence NC_002695 in the left figure and NC_004431 in the right figure and the assembly sequence using MUMmer
A=reference sequence, B=assembly sequence


3. Marçais G, et al. (2018) Mummer4: a fast and versatile genome alignment system. *PLoS computational biology* 14(1):e1005944.
4. Saitou N, Nei M (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution* 4(4):406–425.
5. Thompson JD, Higgins DG, Gibson TJ (1994) Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research* 22(22):4673–4680.
6. Edgar RC (2004) Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research* 32(5):1792–1797.
7. Pagel P, et al. (2007) Comparative genomics.
8. Tenaillon O, Skurnik D, Picard B, Denamur E (2010) The population genetics of commensal escherichia coli. *Nature Reviews Microbiology* 8(3):207.
9. Kitts PA, et al. (2015) Assembly: a resource for assembled genomes at ncbi. *Nucleic acids research* 44(D1):D73–D80.
10. Luo C, Hu GQ, Zhu H (2009) Genome reannotation of escherichia coli cft073 with new insights into virulence. *BMC genomics* 10(1):552.
11. Krzywinski M, et al. (2009) Circos: an information aesthetic for comparative genomics. *Genome research* 19(9):1639–1645.