BINF-F-402 — Genomics Proteomics and Evolution
Assignment 2 – PlaySynteny

Robin Petit
MA1-CS ULB
robpetit@ulb.ac.be

Wednesday 20 December 2017

Contents

1 Files preprocessing

Source code availability All the source code that has been used in order to perform this study is available at: `https://github.com/RobinPetit/BINF-F402`, unless one time use commands such as global calls to `sed`, `head` or `grep`.

Before starting this study, the programs `minimap2` and `minidot` (from `miniasm`) must be installed and the executables must be in a directory accessible from the `PATH` variable (either by adding the directory/directories of the binaries to the `PATH` variable or by copying the binaries to `/usr/bin` or assimilate).

First of all, the data files have been downloaded and renamed as:

- `Sce.fasta` for the DNA sequence of *Saccharomyces cerevisiae: S288C*.
- `Kyokai7.fasta` for the DNA sequence of *Kyokai7*.
- `Adineta.fasta` for the DNA sequence of *Adineta vaga*.

The filenames were reduced in order to lighten the commands and to clarify the report.

As `Adineta.fasta` contains a lot of different fragments, and only the 50 first of them are required, the structure of the file has been determined by `grep '>' Adineta.fasta | head` which shows the ten first headers:

```
>scaffold_1 1087316 bp
>scaffold_2 1048610 bp
>scaffold_3 1045751 bp
>scaffold_4 1034232 bp
>scaffold_5 1021499 bp
>scaffold_6 1008552 bp
>scaffold_7 962643 bp
>scaffold_8 956989 bp
>scaffold_9 947474 bp
>scaffold_10 934921 bp
```

We can see that the header of a scaffold follows the form `>scaffold_%d %d bp`, therefore, finding the 51st is performed by `grep -n '>scaffold_51 '` (note the space after 51 to discard 51x or 51xy scaffolds), which returns: `627309:>scaffold_51 578006 bp`.

Therefore, only the 627,308 first lines of the file are required. These are extracted from `Adineta.fasta` and written in a new file `Adineta_reduced.fasta` with `head -n 627308 Adineta.fasta > Adineta_reduced.fasta`.

Now that the important data has been extracted, the scaffolds are renamed to clarify the plots of `minidot` by using `sed`. The goal is to remove the *scaffold_* part of the headers, which is performed by executing: `sed 's/>scaffold_/>/ Adineta_reduced.fasta > Adineta_reduced_renamed.fasta'`. A different file has been used to avoid rewriting the file currently being read. Then the original one is removed by `rm Adineta_reduced.fasta`, and the new one is renamed to replaced the original one: `mv Adineta_reduced_renamed.fasta Adineta_reduced.fasta`.

Note that the `sed` command has a particular argument to modify the file *in place*: `-i`.

Therefore, simplifying the Kyokai7.fasta file is made by:

```
sed -i 's/>.*chromosome \(.*\) scaffold, .*$/>Kyo_\1/' Kyokai7.fasta
```

which extracts only the chromosome number from the headers (and adds `Kyo_` in front of it). `\(.*\)` creates a *capturing group* that can be referred to later by `\1`.

According to the same procedure, the header of `Sce.fasta` is cleaned by:

```
sed -i 's/.*chromosome=\(.*\)]/>Sce_\1/' Sce.fasta
```

And the mitochondrion DNA is removed by the combination of `head` and `grep`.

As chromosome numbers are written in roman numbers, they are displayed in the wrong order by `minidot`. Therefore, to avoid this, `sed` has again been used. Yet, no regular expression (at least quite simple one) can perform this translation. So a Python3 script has been written in order to substitute the roman numbers (using the `roman` standard Python3 package) using `sed`. The script has been called using:

```
./rename_chromosomes.py "Sce_" Sce.fasta 16
./rename_chromosomes.py "Kyo_" Kyokai7.fasta 16
```

2 Data analysis

2.1 Comparison between S288C and Kyokai7

Figure 1 shows the genome comparison of S288C vs Kyokai7. The first observation is that the first diagonal is nearly complete, and some chromosomes (e.g. the chromosome 11) are fully conserved. Non conserved part are usually moved from an area in a chromosome to the same area in another chromosome.

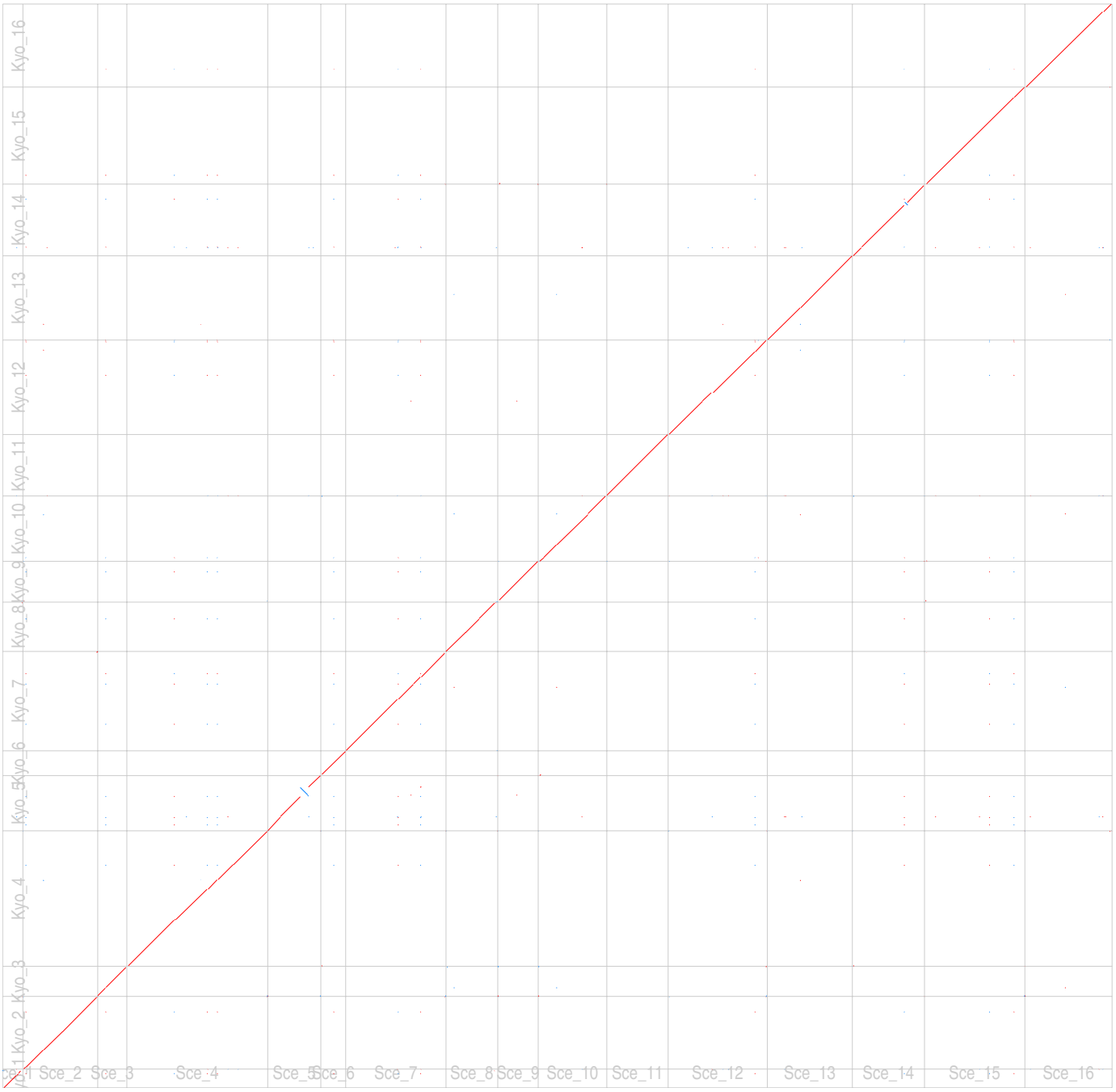

Figure 1: Similarity of genome of both S288C and Kyokai7.

Also, two (quite) large areas of the genome are reversed: on chromosome 5 and 14. We can see that for each of those two regions, they occupy the same position on the chromosome, the difference between the two species being that the area is reversed.

Despite these large conserved areas, a lot of smaller regions can be seen on multiple chromosomes, or even several times on a given chromosome. These are probably duplicated genes that appear in a lot of different

places for some.

These duplicated genes appear mostly as *simple copies*, but some part of them appear as *reversed* copies. A good example can be seen on chromosome 5 of Kyokai and chromosome 12 of S288C.

2.2 Self-comparisons

For the comparison between a species and itself, the `-X` parameter is provided to `minimap2` to not analyze the obvious first diagonal which is "conserved" by definition.

2.2.1 S288C

Figure 2 shows the self similarity of S288C. As previously, there are a lot of gene duplications. Also, some chromosomes have more copies than others. For instance, chromosome 4 (on the horizontal axis) shows a lot of similarities with other chromosomes (1, 2, 3, 4, 10, 12, 13, 14, 15, 16). Also, similarities between chromosome 4 and itself are interesting: we can observe many copies of genes (in both senses: regular and reversed) on this particular chromosome.

This can be observed on a few chromosomes (including 13 and 16), but it's on the fourth chromosome that it is the most visible.

Some interesting patterns can be observed, e.g. on chromosome 12: the beginning and the end of the chromosome seem to be pretty palindromic since the upper left and the lower right corners show somehow linear reversed behaviour, meaning that the beginning and the end of the chromosome are reversed form of one another.

2.2.2 Kyokai7

At first sight, it is clear that Kyokai7 presents a lot less self similarities (see Figure 3) than S288C, and even though S288C and Kyokai7 presented a lot of similarities, the interesting observations made on S288C with 5% similarity analysis (Figure 2) don't apply here. For instance, the palindromic ends of chromosome 12 are not present.

2.2.3 Adineta

Figure 4 shows the self-similarity of Adineta scaffolds. Again, we can see important palindromic regions on chromosomes 15 and 40. Beside these, some similarities (either in regular sense or in reverse) can be found across the scaffolds; e.g. chromosomes couples (17, 3), (18, 5) or (16, 47).

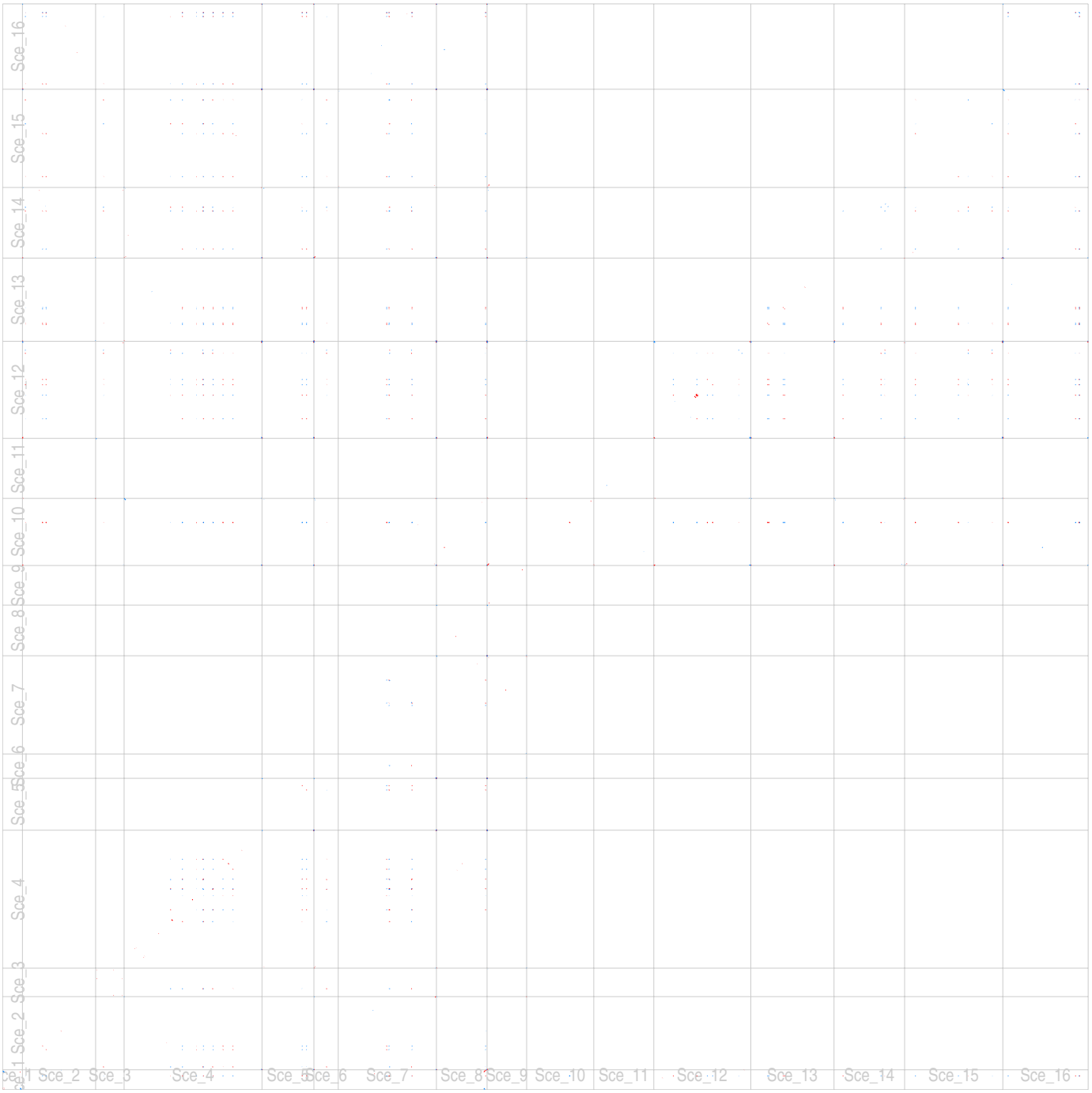

Figure 2: Self-similarity of S288C.

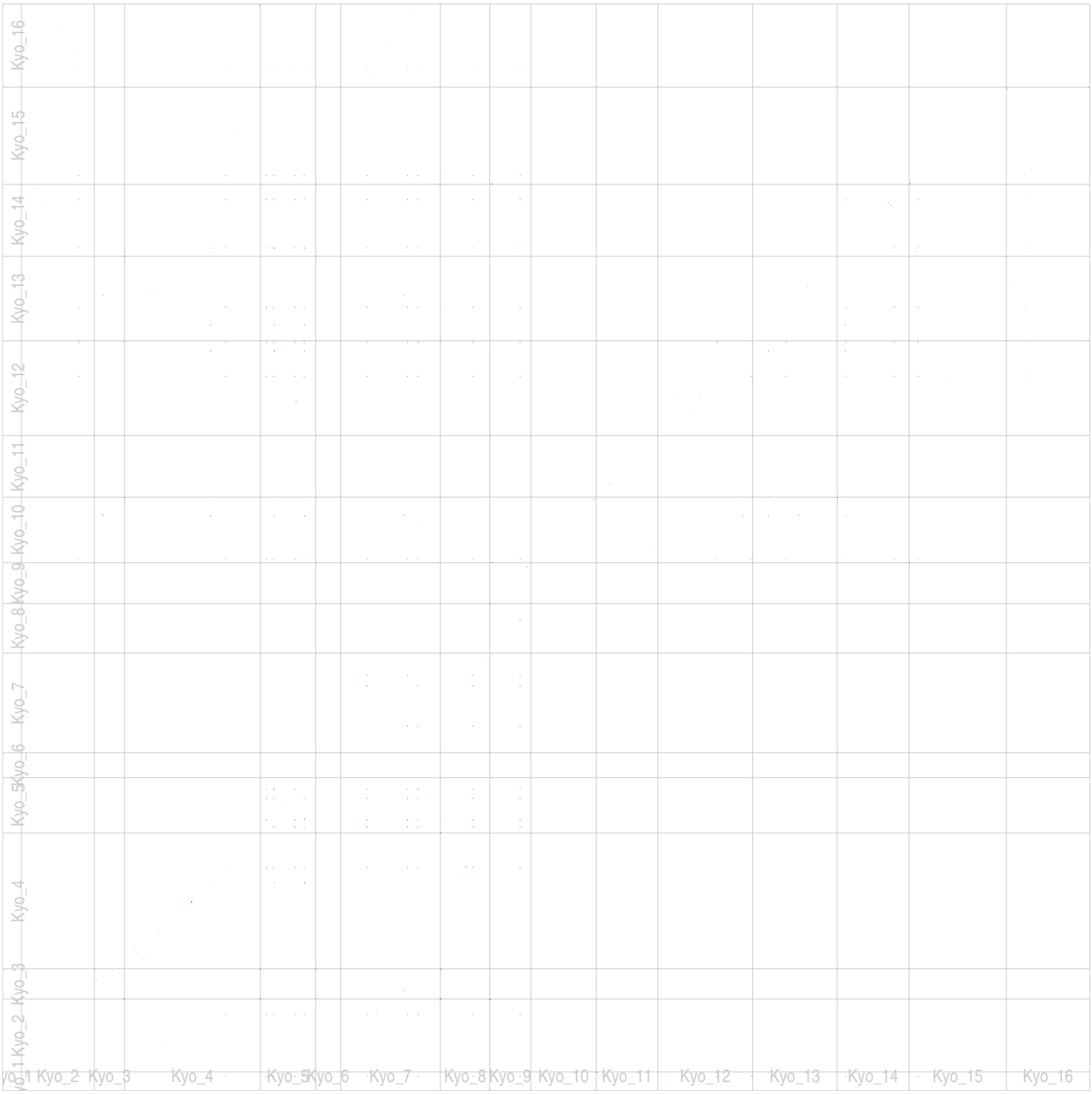

Figure 3: Self-similarity of Kyokai7.

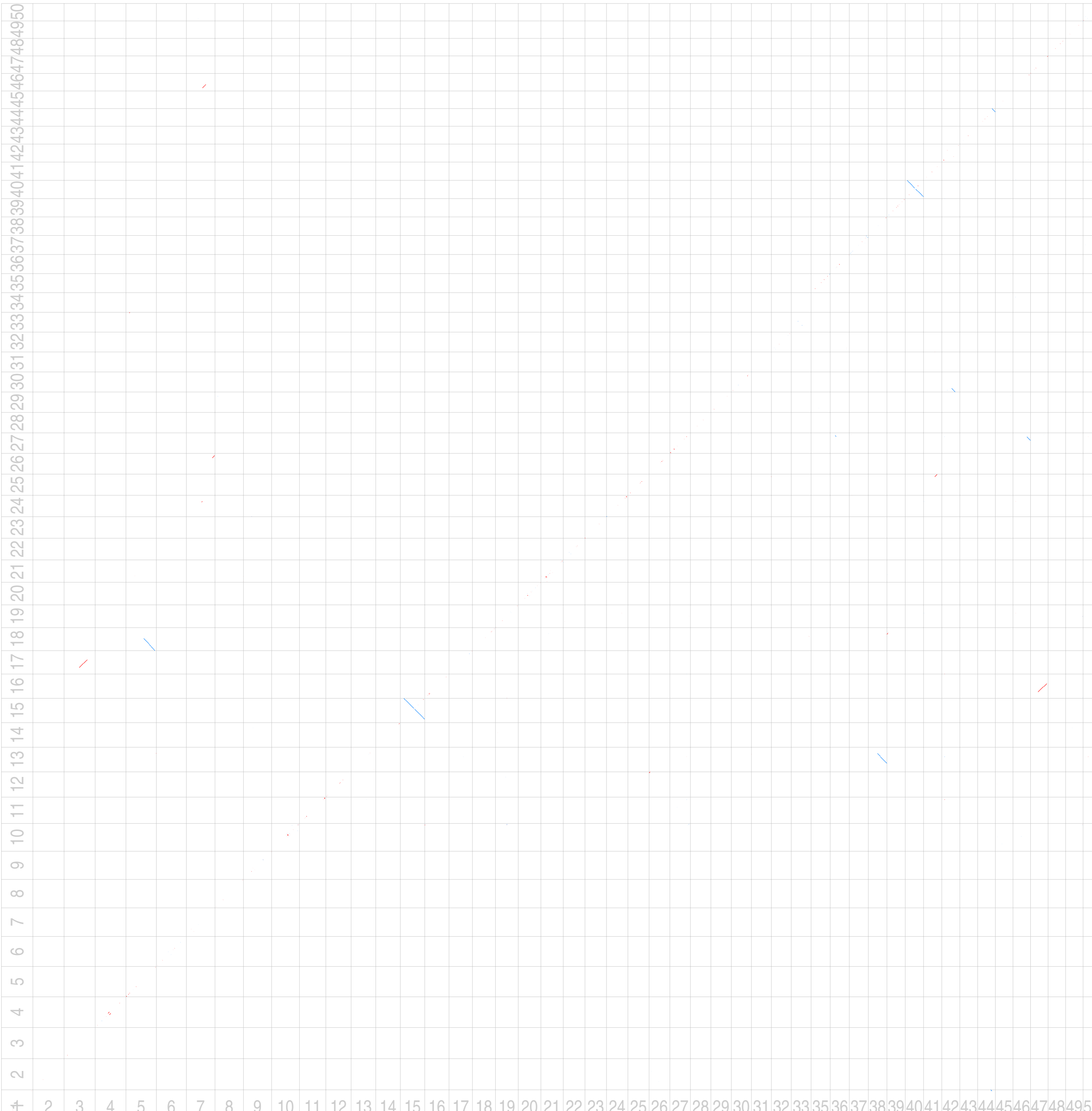

Figure 4: Self-similarity of Adineta.

3 Additional Analysis

Note that the only project has been performed locally on my personal laptop, hydra has not been used. Yet, for this last analysis which is about the human genome, computations have been performed on Hydra that allows more RAM that I could ever think of, i.e. 64Gb for a non prepared job running on personal session. And as the human genome is pretty big (over 3Gb in non compressed form), its analysis require more memory than my laptop can provide.

The human genome has been downloaded from `ftp://ftp.ncbi.nlm.nih.gov/genomes/all/GCA/000/001/405/GCA_000001405.15_GRCh38/seqs_for_alignment_pipelines.ucsc_ids/GCA_000001405.15_GRCh38_no_alt_analysis_set.fna.gz`[1] and `http://hgdownload.cse.ucsc.edu/goldenPath/hg38/chromosomes/` has been tried as well. Yet, both these versions lead `minimap2` to crash. Therefore, only the 5 first chromosomes have been feed into `minimap2`.

Figure 5 doesn't show anything quite useful: a given position of a chromosome is identical to the same position on the same chromosome...

Yet, Figure 6 is widely more complete: a humongous amount of similarities is found. It is complicated to consider all of these to be gene copies or duplications. A hypothesis would be that human genome contains several DNA portions that are repeated, even though these are not entire genes. These can be non-coding regions, or some recurrent motifs of proteins that have a particular function (3D shape), making them recurrent inside the DNA sequence.

Figures 5 and 6 were made according to the results of `minimap2` with the `asm5` preset. Figures 7 and 8 show the same analysis but with the `asm10` preset of `minimap2`, allowing even more divergence.

Figures 5 and 7 are the same, no new similarities were added by by adding the divergence threshold, and Figures 6 and 8, despite being too crowded are identical as well.

[1] See `https://github.com/lh3/minimap2/issues/58` for more details.

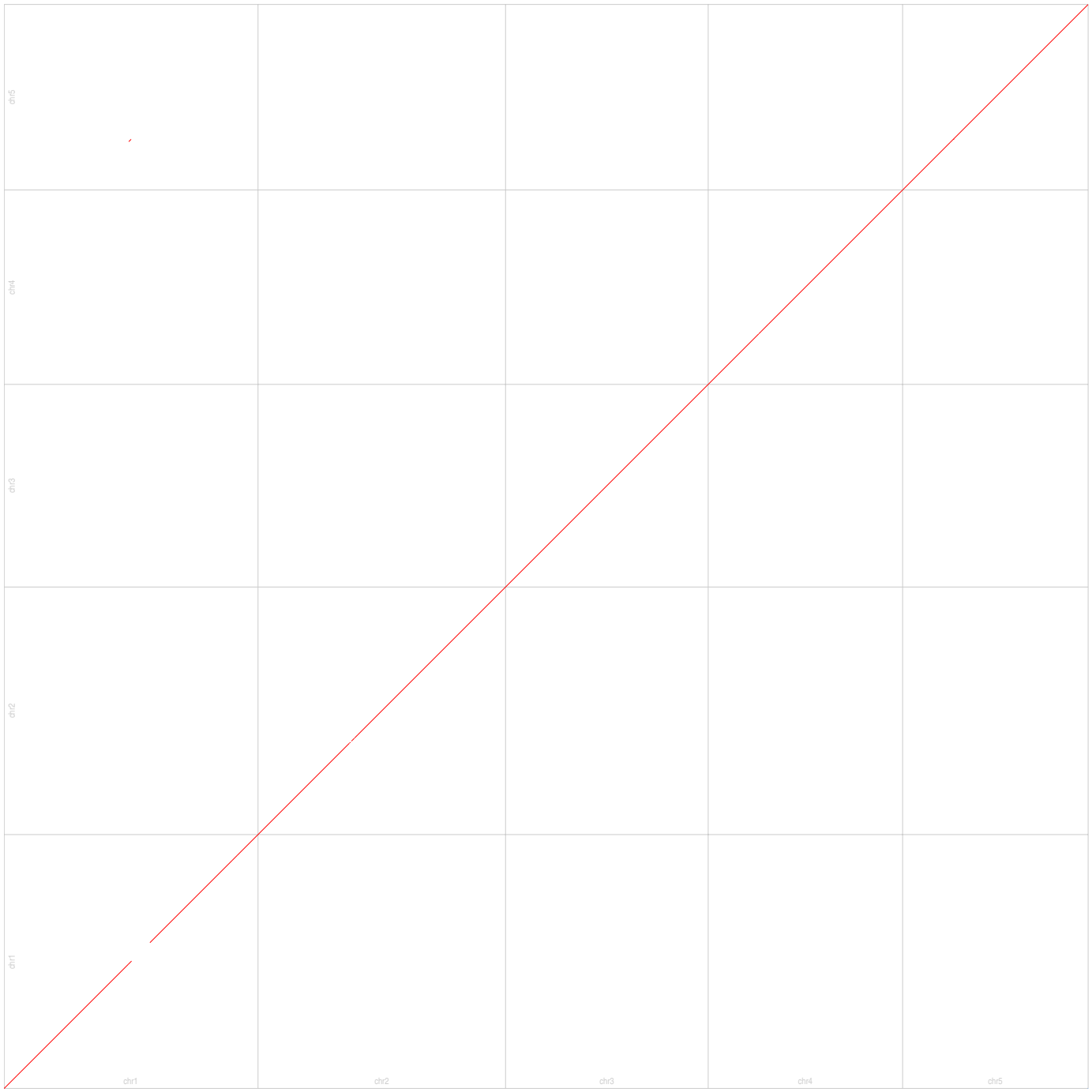

Figure 5: Self-similarity *asm5* of the 5 first chromosomes of the human genome (assembly hg38) without `minimap2`'s `-X` argument.

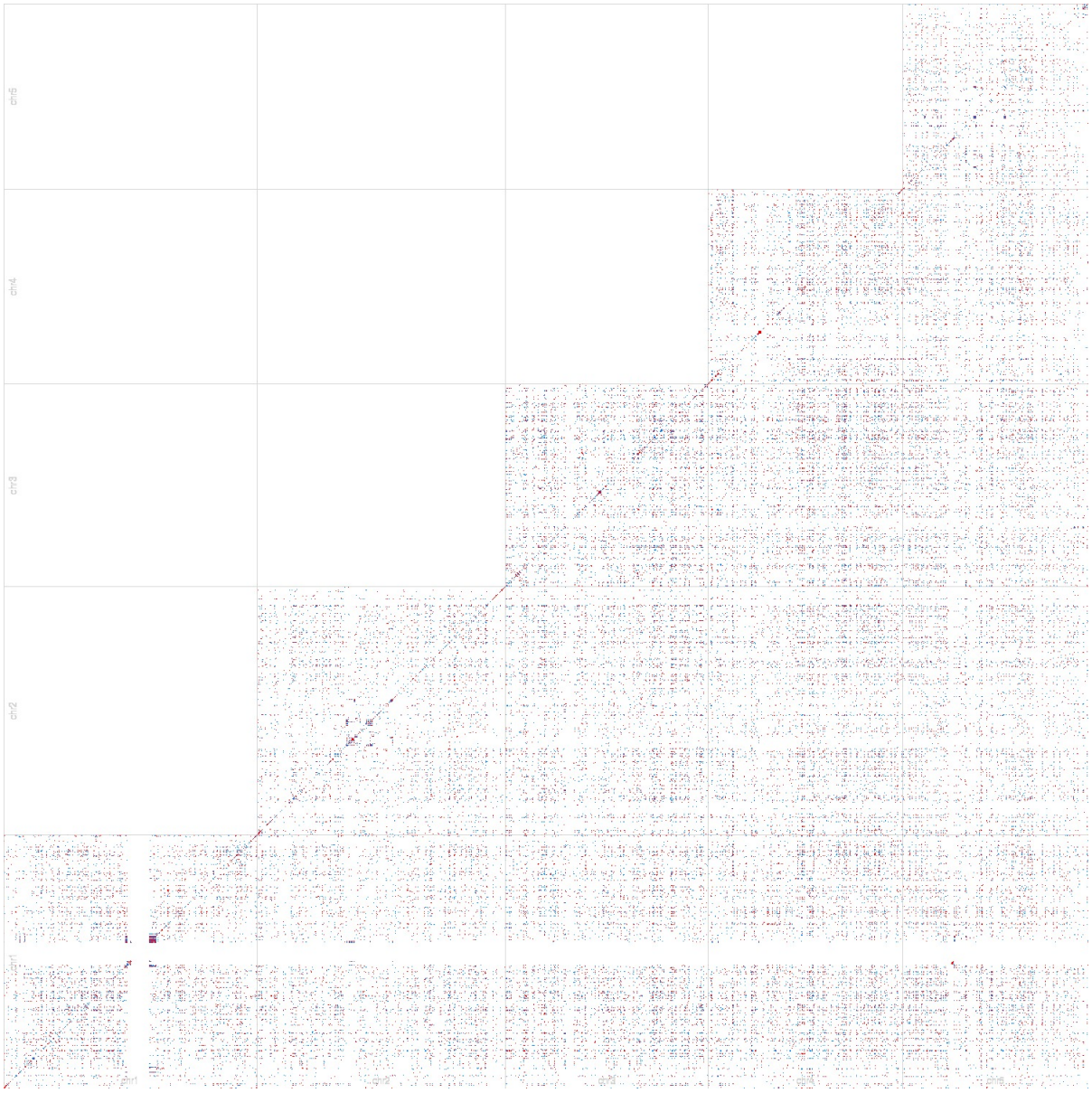

Figure 6: Self-similarity *asm5* of the 5 first chromosomes of the human genome (assembly hg38) with `minimap2`'s `-X` argument.

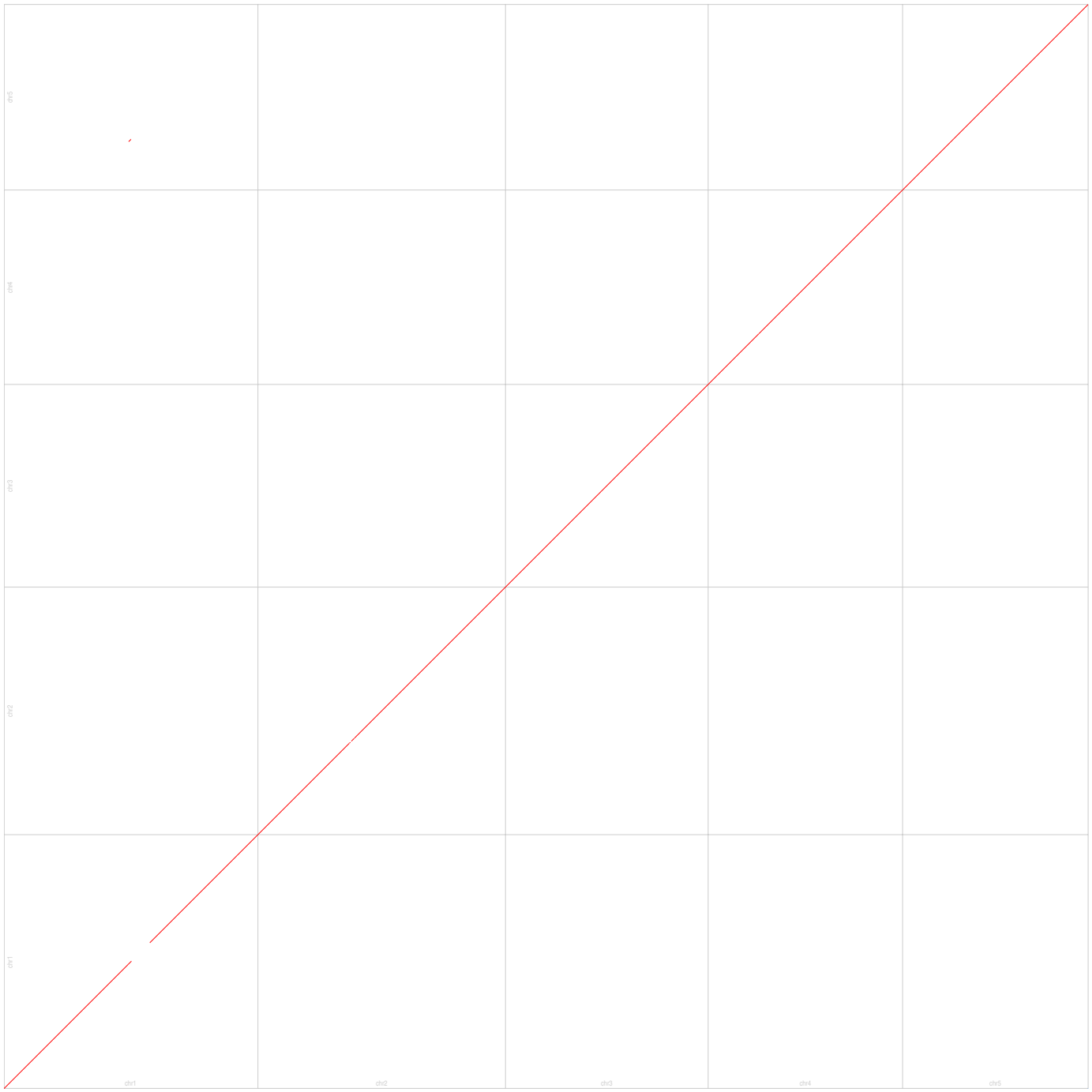

Figure 7: Self-similarity *asm10* of the 5 first chromosomes of the human genome (assembly hg38) without `minimap2`'s `-X` argument.

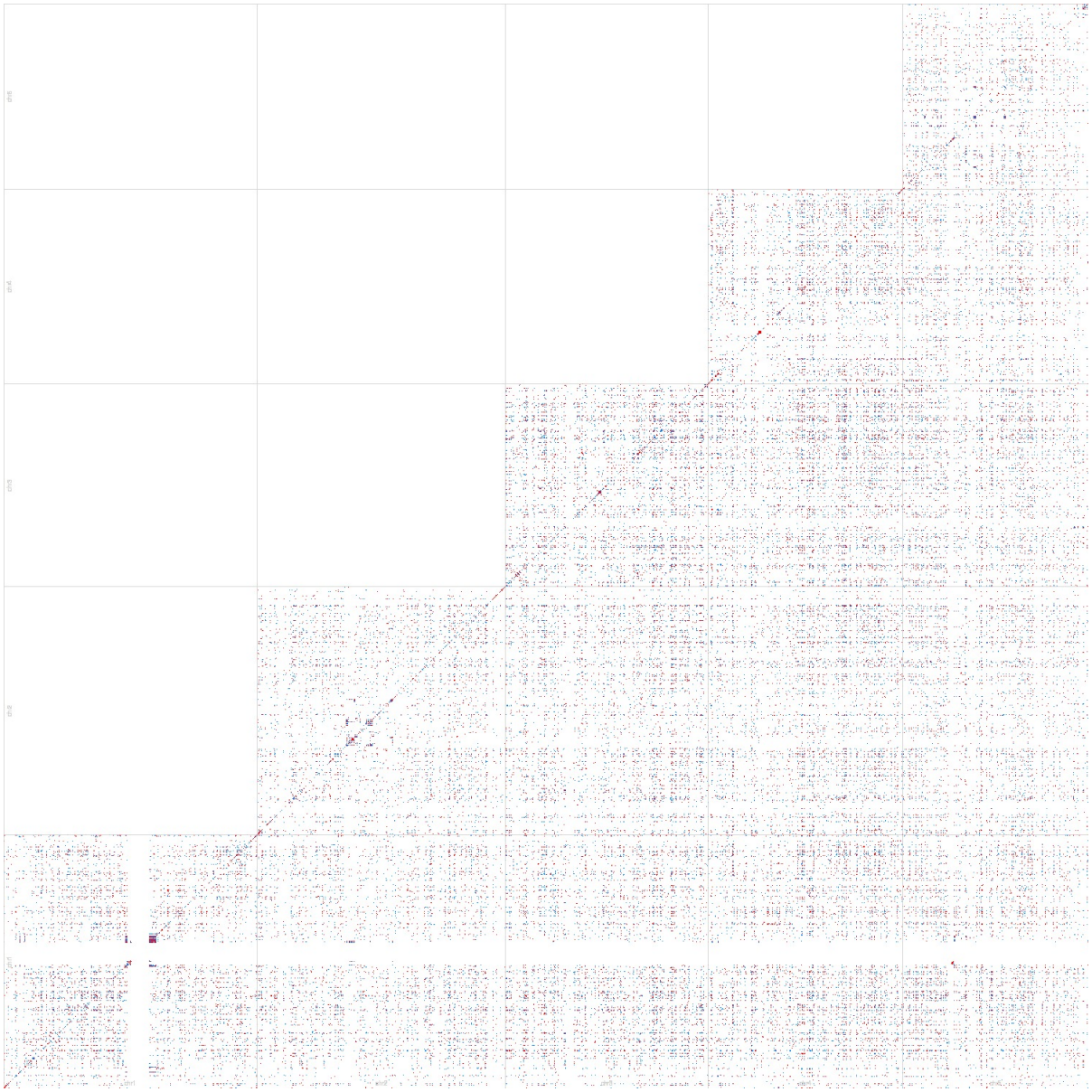

Figure 8: Self-similarity *asm10* of the 5 first chromosomes of the human genome (assembly hg38) with `minimap2`'s `-X` argument.