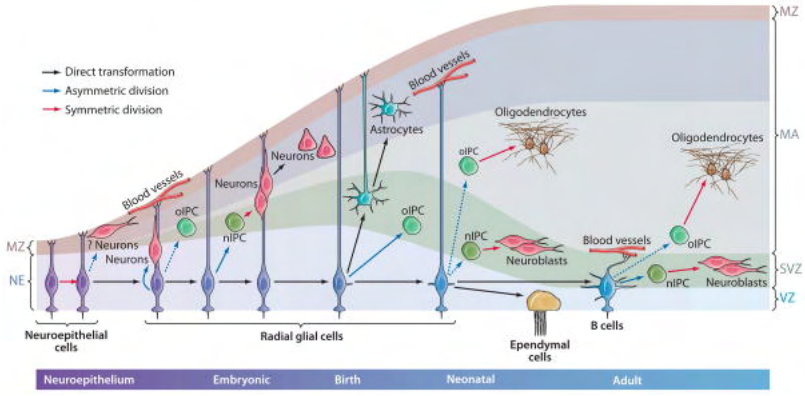
Project

Created by	Ekaterina Kashuk
Created time	@May 7, 2023 5:38 PM
Last edited by	Alice Fedorenko
Last edited time	@May 11, 2023 11:50 PM
Tags	

Introduction and Student input

One can observe a huge difference in chromatin structure patterns between the neuronal and glial cells. In mammals, DNA methylation plays a critical role in genomic imprinting, and X chromosome inactivation, as well as cellular differentiation and development, and is generally considered to be associated with transcriptional repression. We applied ChiP-seq, RNA-seq and Hi-C to the data from Chandrasekaran, S. et al, 2021 to perform a genomic-scale single-site resolution DNA methylation analysis in neuronal and nonneuronal nuclei separated from the postmortem mice brain.



Picture1. The nature of development on neural and non-neural (glial) cell types.

The aim of our project is to clarify these changes in mouse epigenomic profile of the mentioned cells and to provide an explanation of such chromatin variability. Project goals:

1. To explore chromatin difference in neuronal (NeuN+) and non-neuronal (NeuN−) nuclei (TADs density, loops prominence, compartment changes) - **Ekaterina Kashuk**;
2. To detect polycomb interacting regions (fithic) + to identify H3K27ac track for open chromatin - **Leonid Sidorov**;
3. To find out a correlation between compartments, TADs, loops and gene expression - **Ksenia Kubenko**.
4. To compare neuronal and glial H3K9me3 tracks around the regions, compare H3K27me3 to H3K9me3 tracks, investigate Ring1B tracks in *in vitro* differentiated neurons (NPC and CN) - **Alisa Fedorenko**;

Hi-C analysis

To explore chromatin differences in NeuN+ and NeuN- we need to perform Hi-C analysis.

The aim is to find differences between NeuN+ and NeuN- for:

- TADs density
- loops prominence
- compartment changes

For these purposes we used following preprocessed Hi-C files from the article:

- for NeuN+ (neurons): GSE168524_neurons_fem_wt_allValidPairs.hic
- for NeuN- (glia): GSE168524_female_glia_allValidPairs.hic

First of all, .hic files were converted to .cool using HiCExplorer [1]:

```
hicConvertFormat --matrices ${file.hic} --outFileName ${file} --inputFormat hic
--outputFormat cool
```

Compartments

For compartment calling we used cooltools [2]. Compartments were called at **250 kb resolution**, as it was done in the article.

Computing expected:

```
cooltools expected-cis ${file.mcool}::/resolutions/250000
-o {expected.tsv} --regions ${genome_file.bed}
```

Call compartments:

```
cooltools eigs-cis ${file.mcool}::/resolutions/250000 -o
${compartment_prefix}
```

Saddle plot:

```
cooltools saddle ${file.mcool}::/resolutions/250000 ${compartment_prefix}.cis.vecs.tsv
${expected.tsv} -o ${saddle} --fig png --qrange 0.02 0.98 --regions ${genome_file.bed}
```

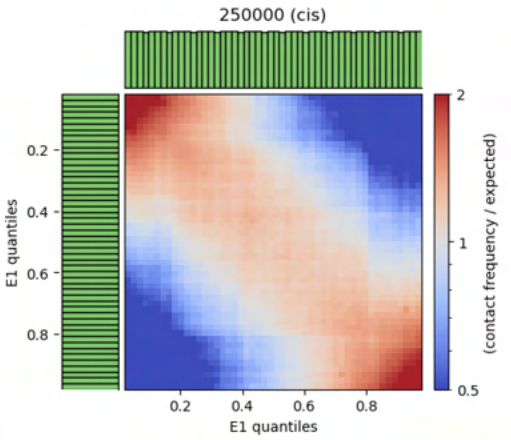


Fig.1. Saddle plot of Hi-C data binned at 250 kb resolution for glia.

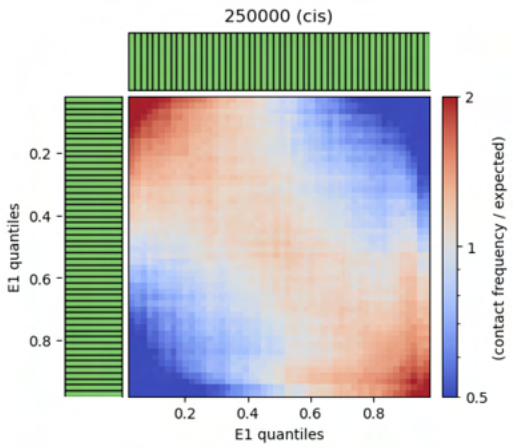


Fig.2. Saddle plot of Hi-C data binned at 250 kb resolution for neurons.

The saddle plot shows enrichment of interactions as a function of the compartment vector. Rows and columns correspond to the genomic region bins, and matrix entries reflect the bins' interactivity. Positive eigenvector values reflect more than expected contacts, while negative values reflect less than expected contacts. All in all, positive EV values could mean that a region is A compartment and negative EV values could mean that a region is B compartment.

In these plots, interactions in the top left corner represent interactions between B compartments (B-B interactions) and interactions in the bottom right corner represent A–A interactions. Interactions in the top right corner represent B-A interactions and interactions in the bottom left corner represent A-B interactions [3]. (Abramo et al., 2019).

Interactions on saddle plot for neurons (Fig.2) seem to be weaker than interactions on saddle plot for glia (Fig.1).

TADs

For TADs calling we used Chromosight [4]. TADs were detected at **10 kb resolution**.

TADs detection:

```
chromosight detect --threads 2 --min-dist 6000 --max-dist 45000
${file.mcool}::/resolutions/10000 TAD_10000 --pattern borders
```

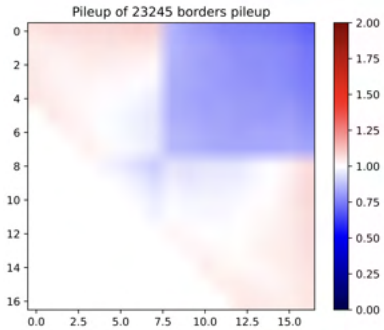


Fig.3. Average border plot of Hi-C data binned at 10 kb resolution for glia.

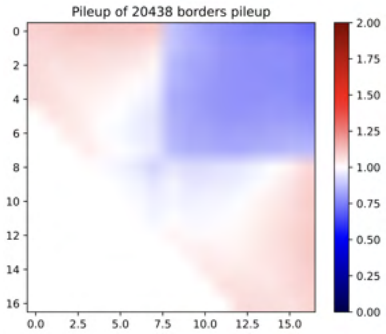


Fig.4. Average border plot of Hi-C data binned at 10 kb resolution for neurons.

Chromosight computes correlation coefficients by convolving the template (represents a 3D structure of interest: loop, boundary, etc.) over a whole-genome contact map [5]. Correlation coefficients are quite high for both glia and neurons data. Therefore, we could consider with relatively high evidence presence of TADs both in neurons and glia data. Moreover, correlation coefficients are relatively higher for neurons data compared to glia data. Thus, with relatively high evidence it might be that TADs in neurons have 'stricter' borders than TADs in glia (i.e. the number of contacts outside of TAD might be **not** dramatically lower than number of contacts within the TAD in glia).

Loops

For loop calling we used Chromosight. Loops were detected at **10 kb resolution**.

Detection of loops:

```
chromosight detect --threads 2 --min-dist 6000 --max-dist 45000
${file.mcool}::/resolutions/10000 loop_10000 --pattern loops
```

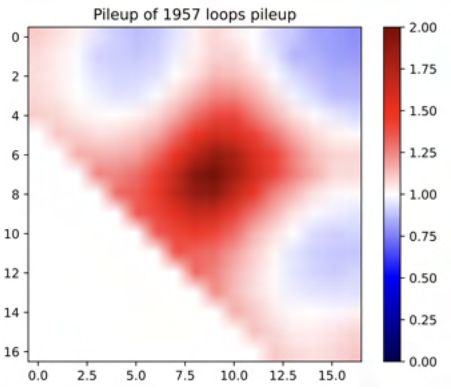


Fig.5. Average loop plot of Hi-C data binned at 10 kb resolution for glia.

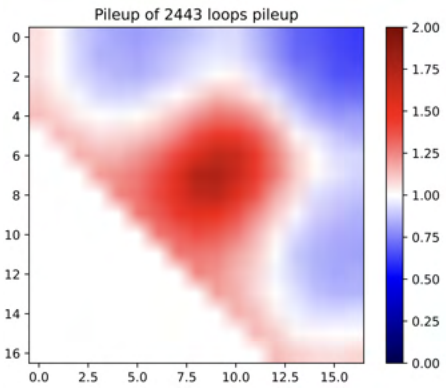


Fig.6. Average loop plot of Hi-C data binned at 10 kb resolution for neurons.

Loop enrichment is higher for glia compared to loop enrichment for neurons.

However, it is essential to use additional biological information to conclude on presence of compartments, TADs and loops both for neurons and glia.

Correction of Hi-C matrix

To compute A/B compartments and call TADs data should be corrected to remove GC, open chromatin biases and to normalize the number of restriction sites per bin.

For all further steps I used HiCExplorer.

Correction for further calculations of compartments:

```
hicCorrectMatrix diagnostic_plot -m ${file.mcool}::/resolutions/250000
-o ${hic_corrected.png}
```

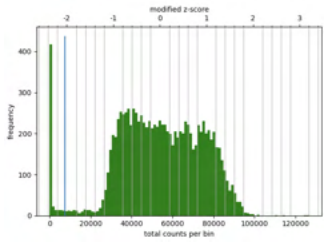


Fig.7. Histogram of the number of counts per bin (resolution 250 kb) for glia.

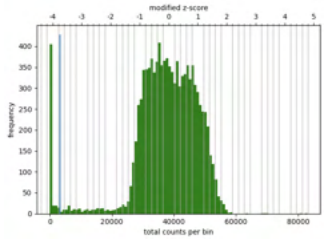


Fig.8. Histogram of the number of counts per bin (resolution 250 kb) for neurons.

Correction for further TADs calling:

```
hicCorrectMatrix diagnostic_plot -m ${file.mcool}::/resolutions/10000
-o ${hic_corrected.png}
```

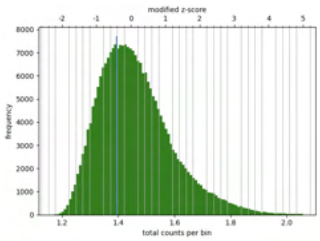


Fig.9. Histogram of the number of counts per bin (resolution 10 kb) for glia.

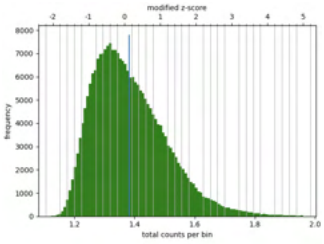


Fig.10. Histogram of the number of counts per bin (resolution 10 kb) for neurons.

Compartments:

For the upper threshold is only important to remove very high outliers and thus a value of 2 could be used. For the lower threshold it is recommended to remove relatively low outliers and thus a value of -1 could be used:

```
hicCorrectMatrix correct -m ${file.mcool}::/resolutions/2500000
--filterThreshold -1 2
-o ${hic_corrected.cool}
```

TADs:

For the upper threshold is only important to remove very high outliers and thus a value of 4 could be used. For the lower threshold it is recommended to remove relatively low outliers and thus a value of -1.6 could be used:

```
hicCorrectMatrix correct -m ${file.mcool}::/resolutions/10000
--filterThreshold -1.6 4
-o ${hic_corrected.cool}
```

TADs visualization

```
hicFindTADs -m ${hic_corrected.cool}
--outPrefix myHiCmatrix_min30000_max100000_step10000_thres0.05_delta0.01_fdr
--minDepth 30000 --maxDepth 100000 --step 10000
--thresholdComparisons 0.05 --delta 0.01 --correctForMultipleTesting fdr -p 64
```

```
hicPlotTADs --tracks ${tracks.ini} --region ${1:4000000-8000000}
-o ${TAD_calling_comparison.png}
```

tracks.ini:

```
[x-axis]
fontsize=10

[hic]
file = ${file.cool}
colormap = Spectral_r
depth = 4000000
transform = log
file_type = hic_matrix
show_masked_bins = false

[tads]
file = ${file_domains.bed}
file_type = domains
border_color = black
line_width = 3
color = none
overlay_previous = share-y
```

TADs visualization for glia:

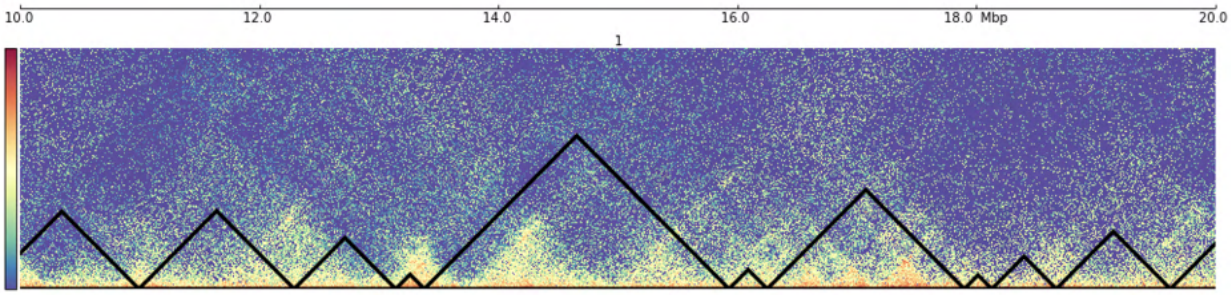


Fig.11. TADs binned at 10 kb resolution, chr.1: 10-20 Mb, for glia.

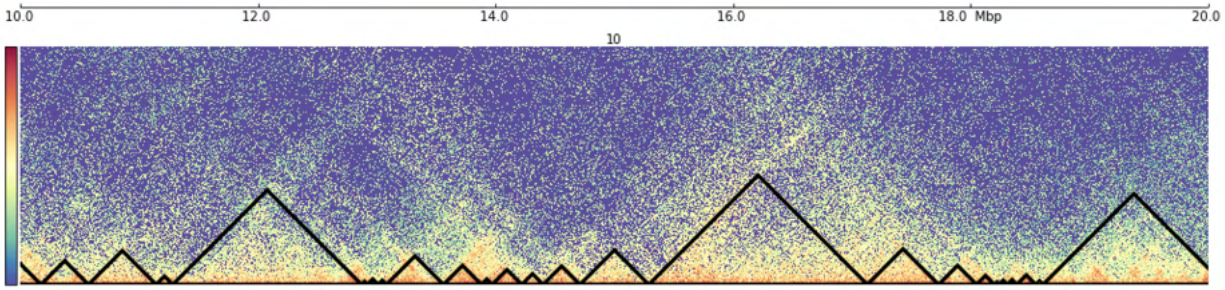


Fig.12. TADs binned at 10 kb resolution, chr.10: 10-20 Mb, for glia.

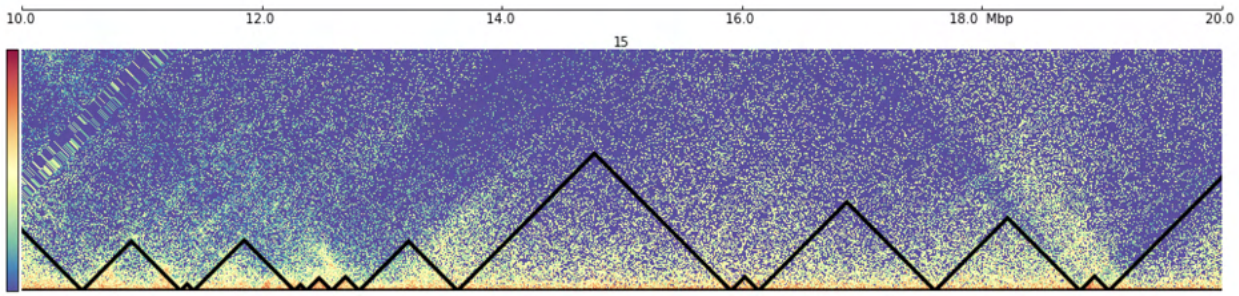


Fig.13. TADs binned at 10 kb resolution, chr.15: 10-20 Mb, for glia.

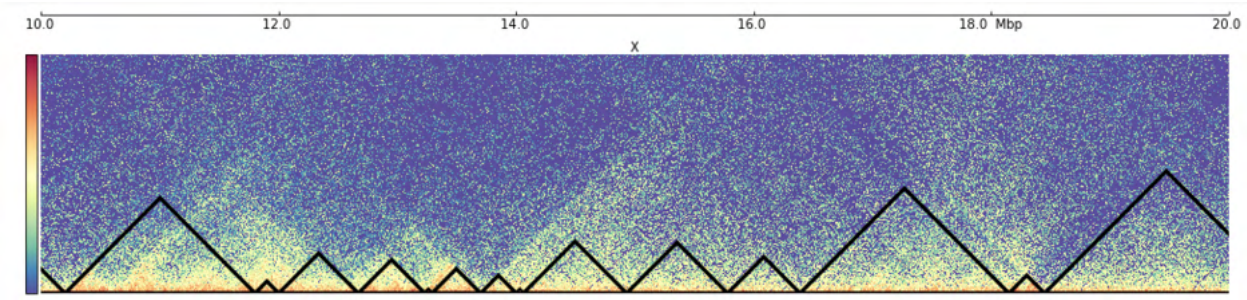


Fig.14. TADs binned at 10 kb resolution, chr.X: 10-20 Mb, for glia.

TADs visualization for neurons:

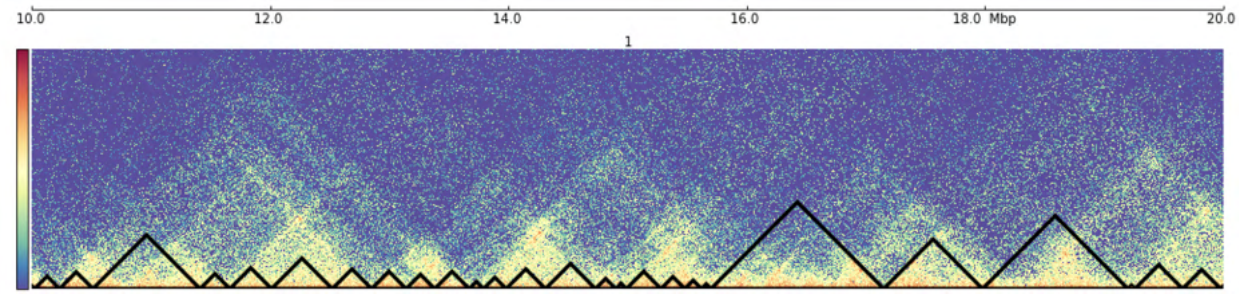


Fig.15. TADs binned at 10 kb resolution, chr.1: 10-20 Mb, for neurons.

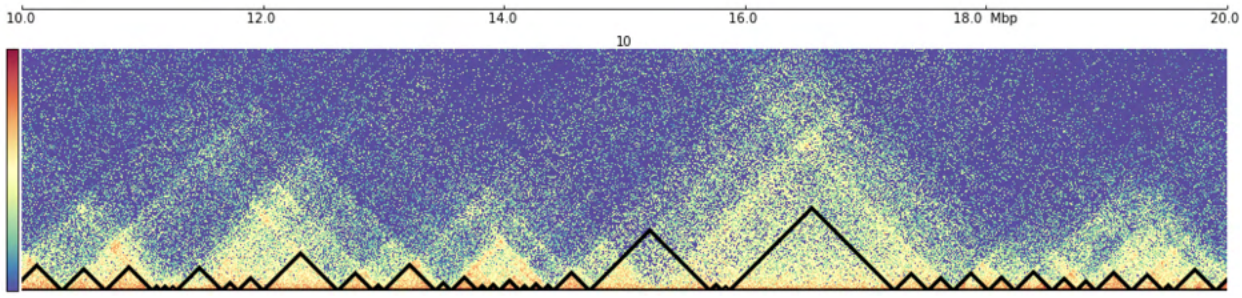


Fig.16. TADs binned at 10 kb resolution, chr.10: 10-20 Mb, for neurons.

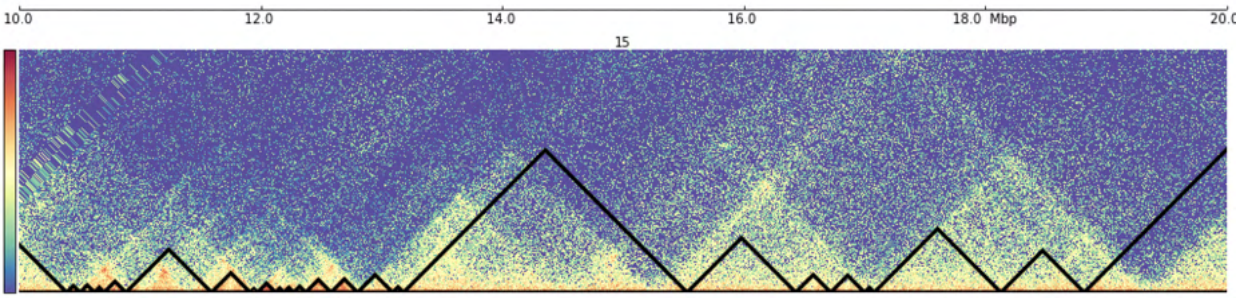


Fig.17. TADs binned at 10 kb resolution, chr.15: 10-20 Mb, for neurons.

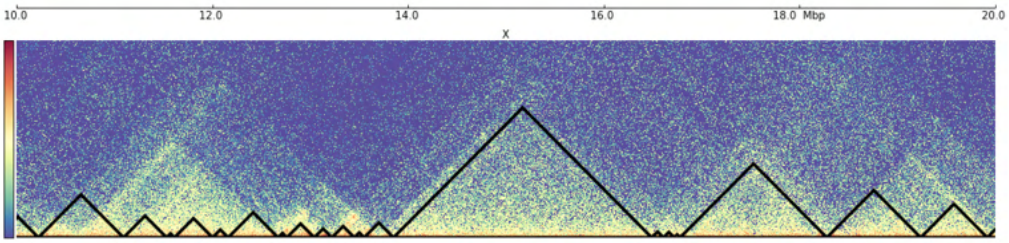


Fig.18. TADs binned at 10 kb resolution, chr.X: 10-20 Mb, for neurons.

All in all, it could be concluded that TADs are presented in both glia and neurons, however, the amount of TADs is larger for neurons than amount of TADs for glia. Moreover, TADs in neurons seem to be ‘smaller’ and to have stricter borders.

Compartment visualization

Transform initial matrix to an observed/expected matrix:

```
hicPCA -m ${hic_corrected.cool} --outFileName pca1.bw pca2.bw --format bigwig
```

Compute pearson correlation matrix and based on it a covariance matrix and compute the eigenvectors based on the covariance matrix:

```
hicTransform -m ${hic_corrected.cool} --outFileName all.cool --method pearson
```

Plot A/B compartments for each chromosome:

```
hicPlotMatrix -m ${pearson_all.cool} --outFileName pca1.png
--perChr --bigwig pca1.bw
```

A/B compartments binned at 250 kb resolution per each chromosome for glia and neurons:

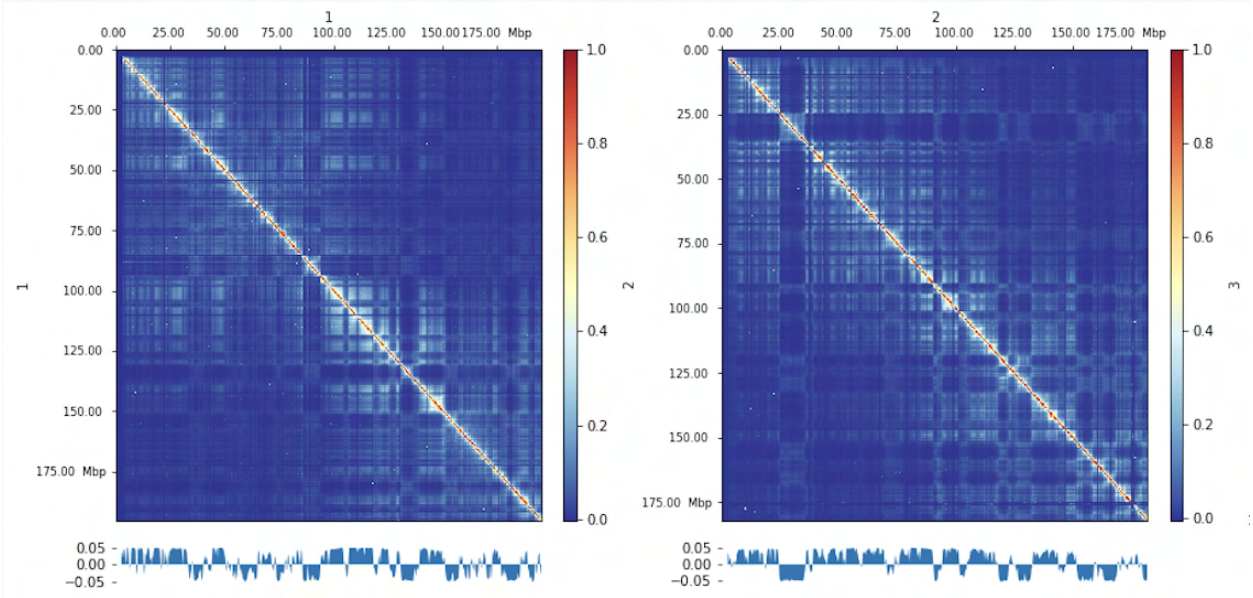


Fig.19. A/B compartments binned at 250 kb resolution, chr. 1, 2 for glia.

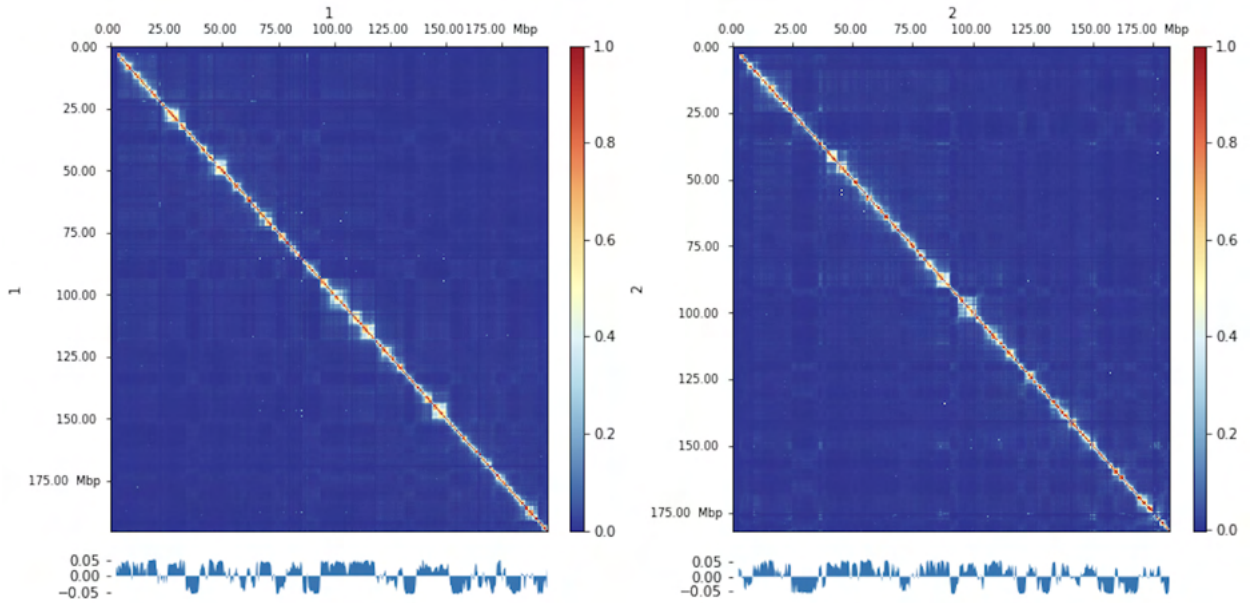


Fig.20. A/B compartments binned at 250 kb resolution, chr. 1, 2 for neurons.

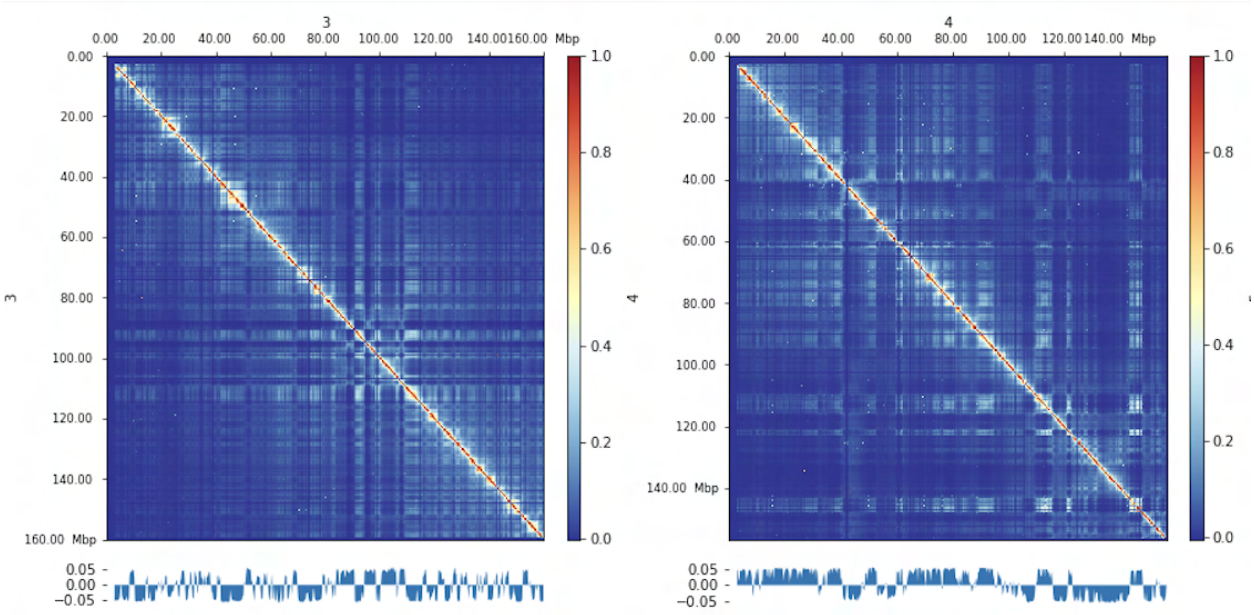


Fig.21. A/B compartments binned at 250 kb resolution, chr. 3, 4 for glia.

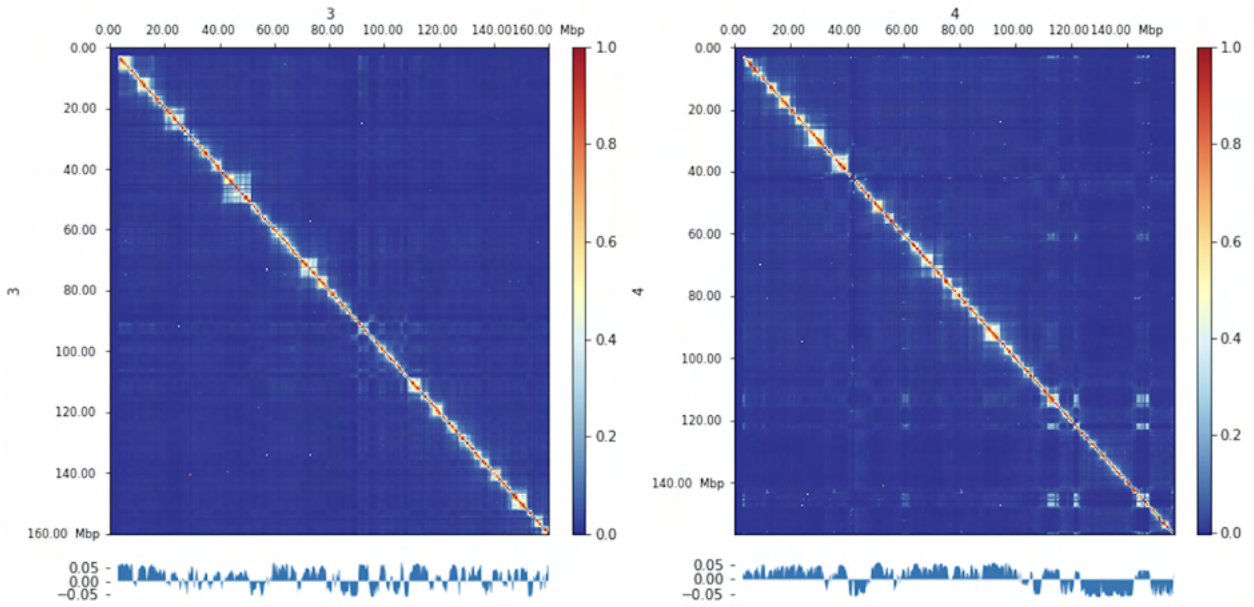


Fig.22. A/B compartments binned at 250 kb resolution, chr. 3, 4 for neurons.

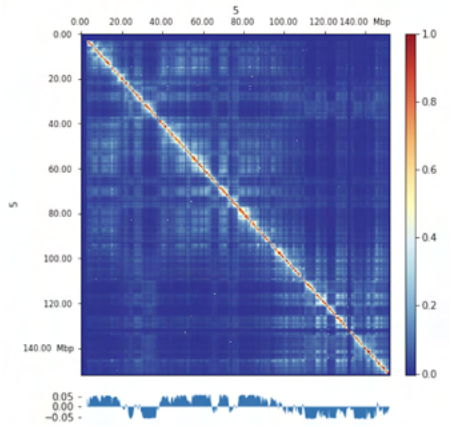


Fig.23. A/B compartments binned at 250 kb resolution, chr. 5 for glia.

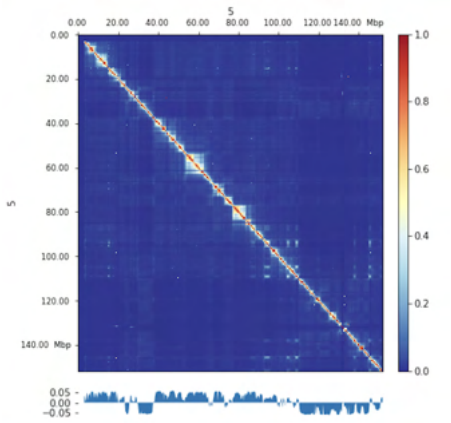


Fig.24. A/B compartments binned at 250 kb resolution, chr. 5 for neurons.

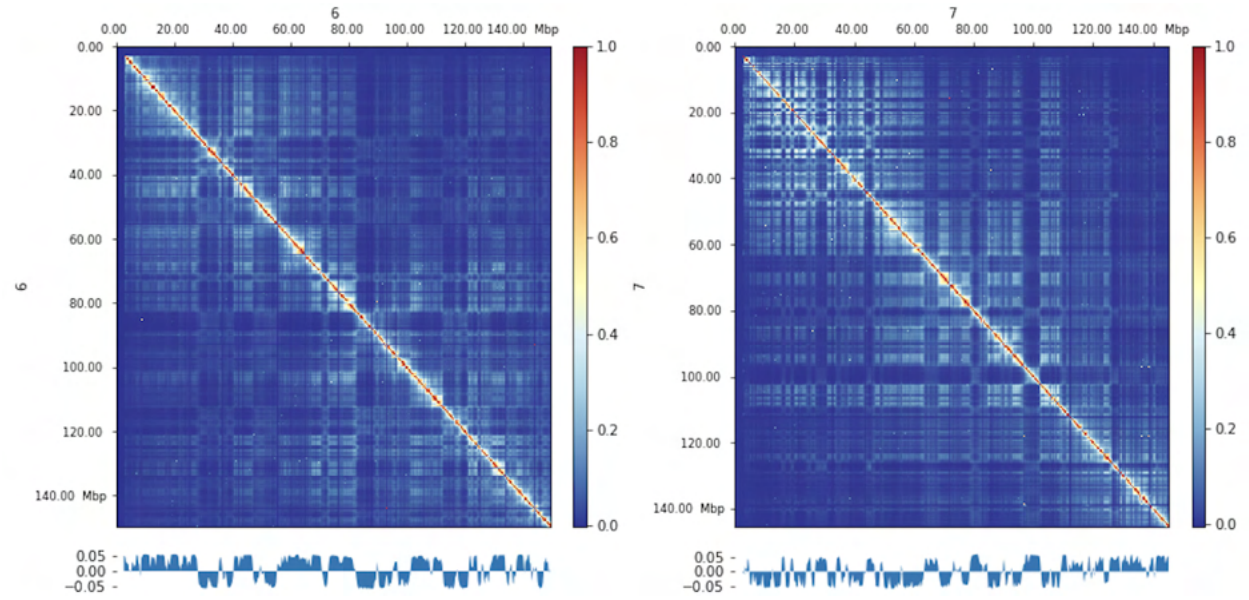


Fig.25. A/B compartments binned at 250 kb resolution, chr. 6, 7 for glia.

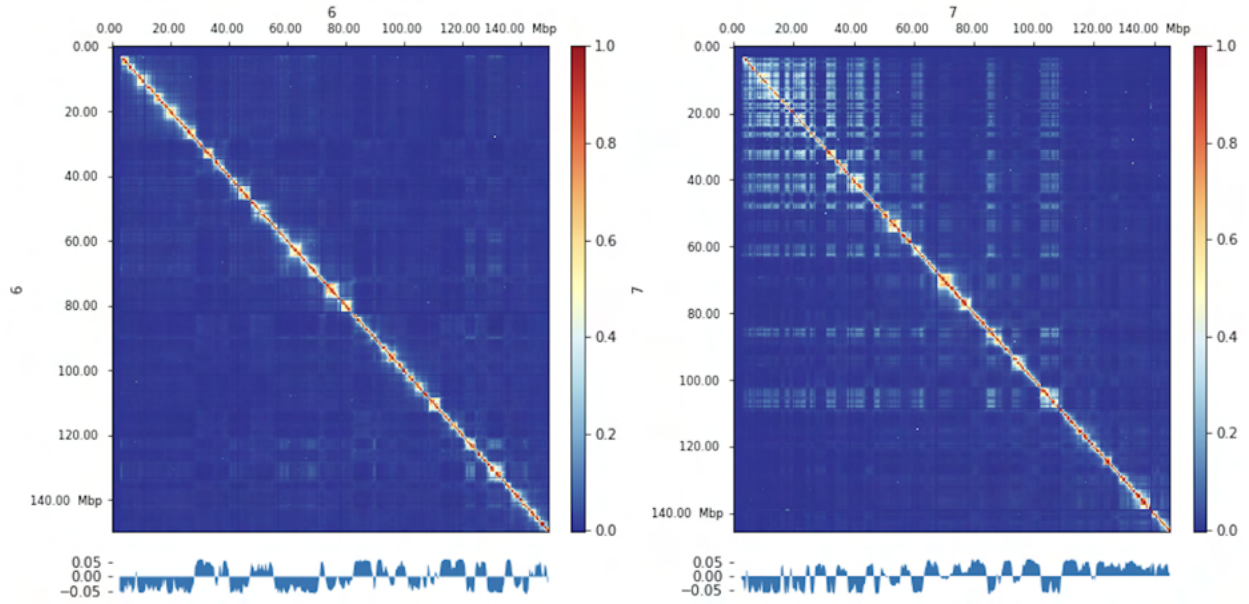


Fig.26. A/B compartments binned at 250 kb resolution, chr. 6, 7 for neurons.

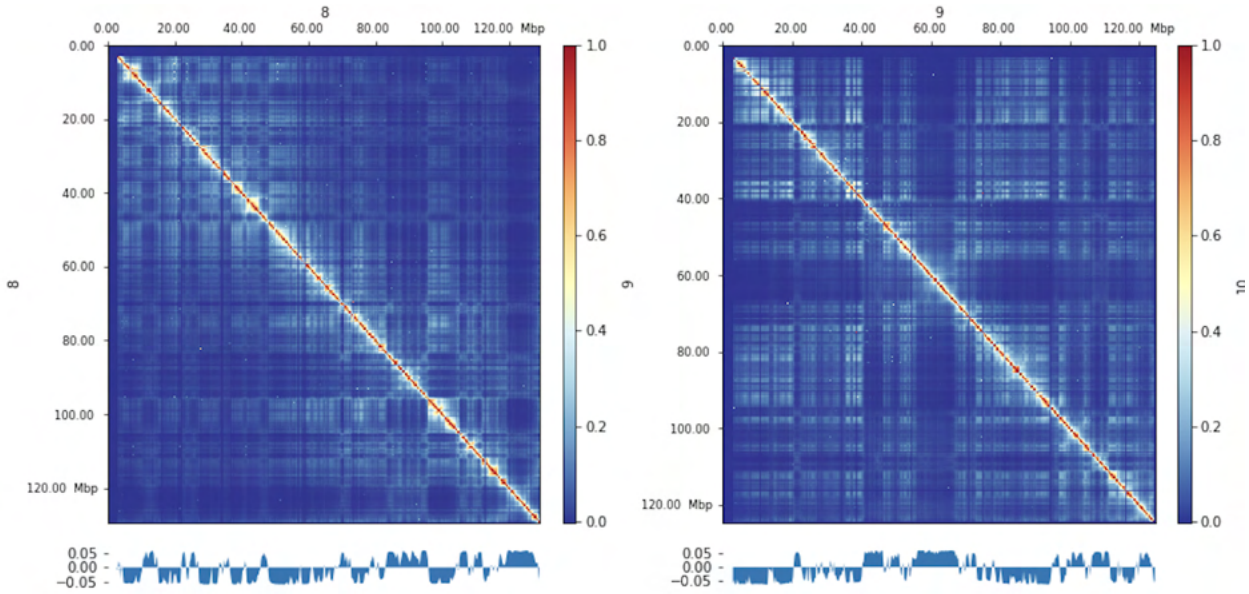


Fig.27. A/B compartments binned at 250 kb resolution, chr. 8, 9 for glia.

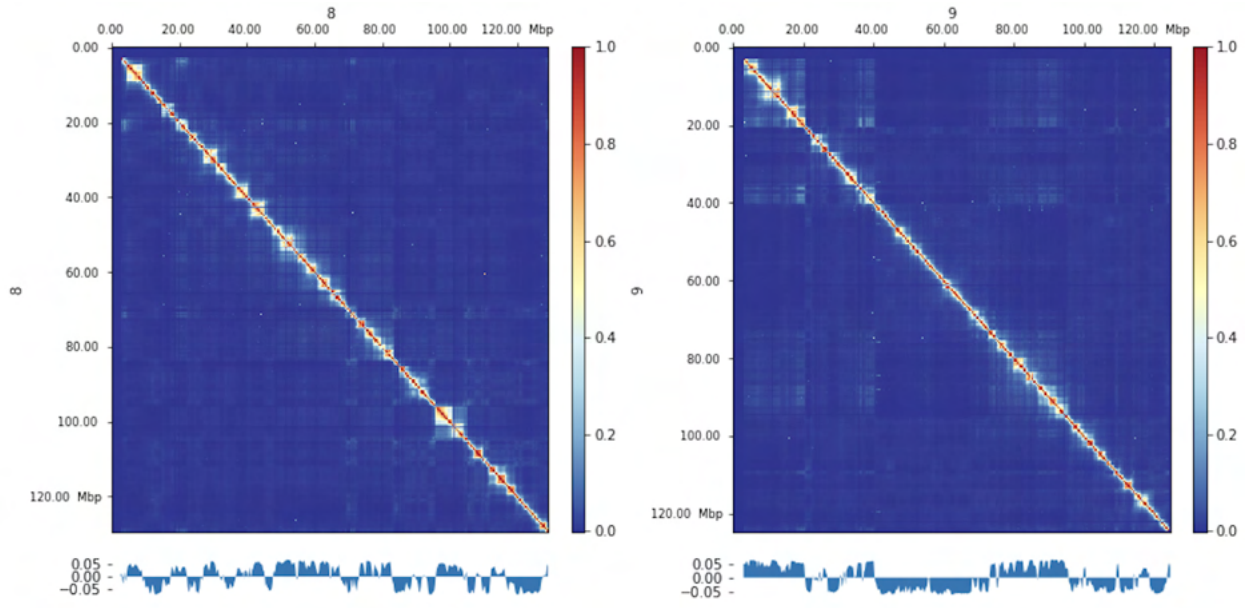


Fig.28. A/B compartments binned at 250 kb resolution, chr. 8, 9 for neurons.

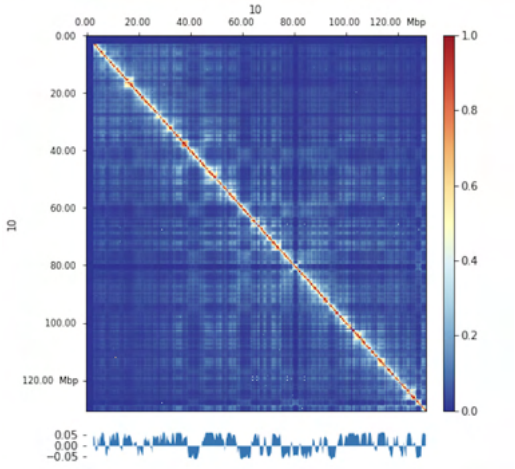


Fig.29. A/B compartments binned at 250 kb resolution, chr. 10 for glia.

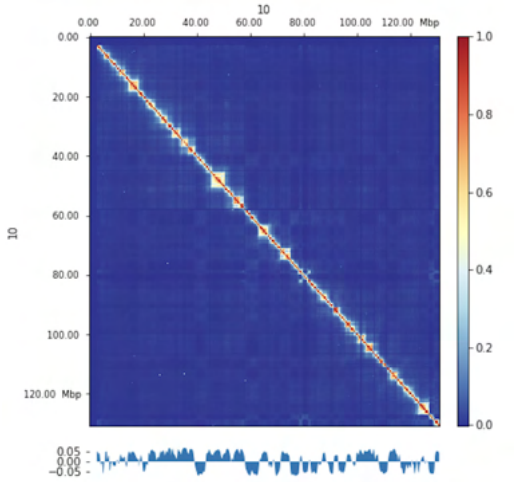


Fig.30. A/B compartments binned at 250 kb resolution, chr. 10 for neurons.

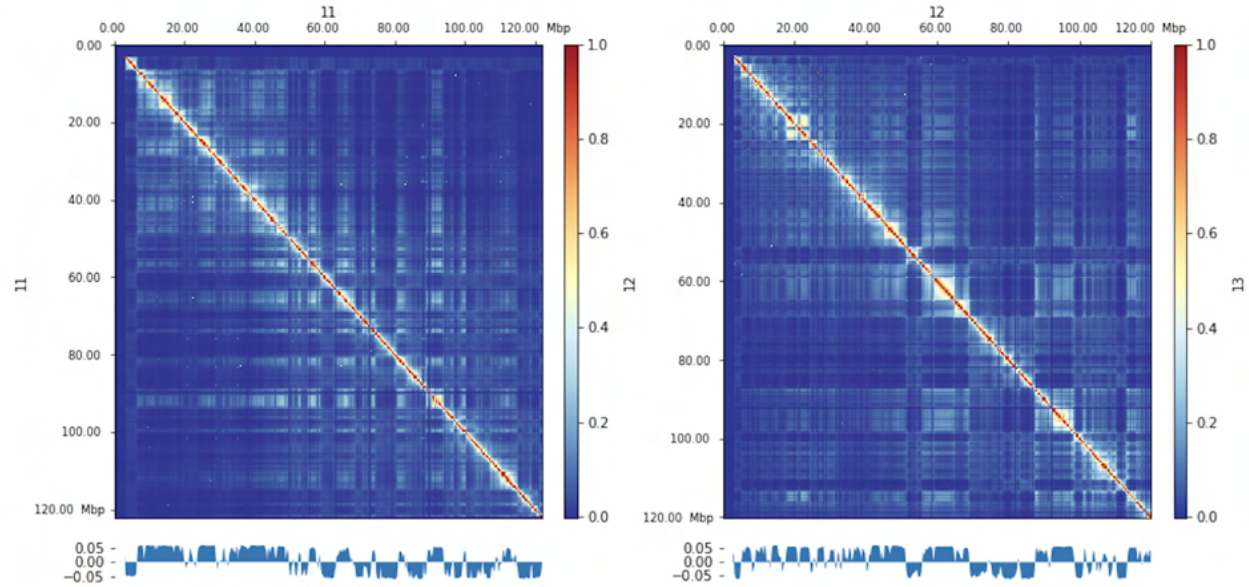


Fig.31. A/B compartments binned at 250 kb resolution, chr. 11, 12 for glia.

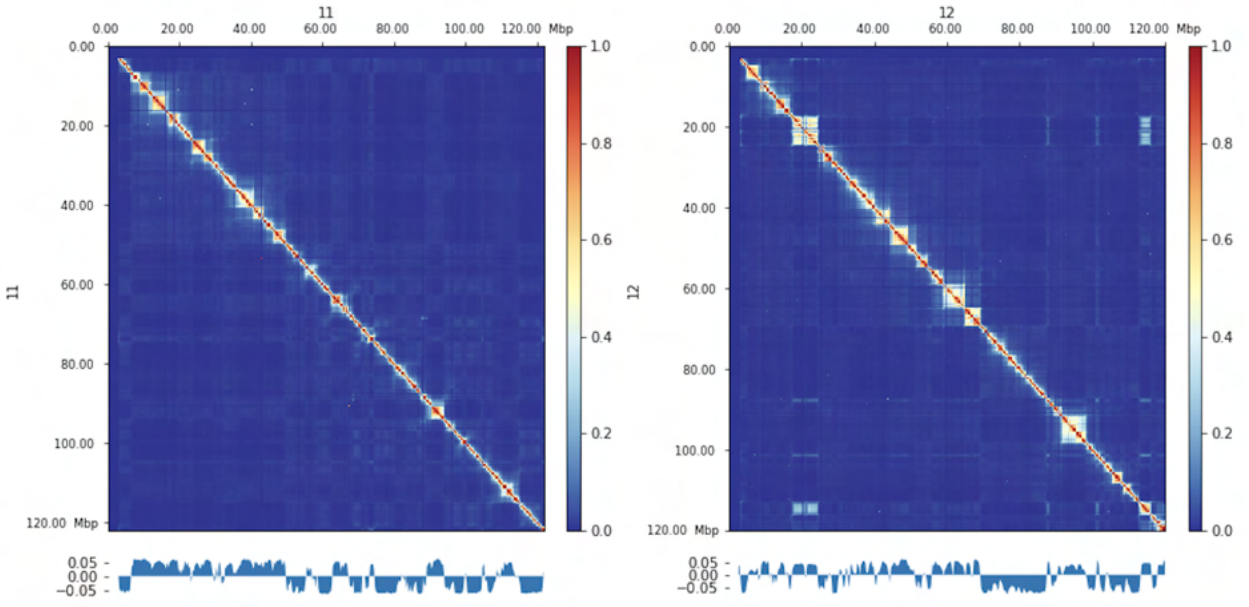


Fig.32. A/B compartments binned at 250 kb resolution, chr. 11, 12 for neurons.

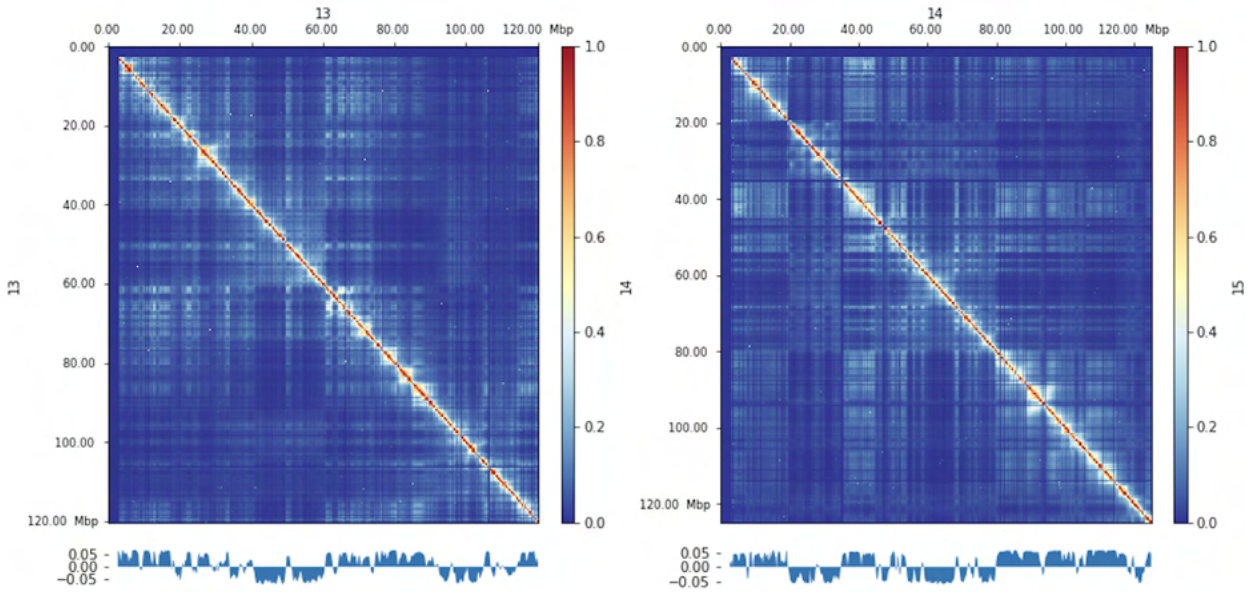


Fig.33. A/B compartments binned at 250 kb resolution, chr. 13, 14 for glia.

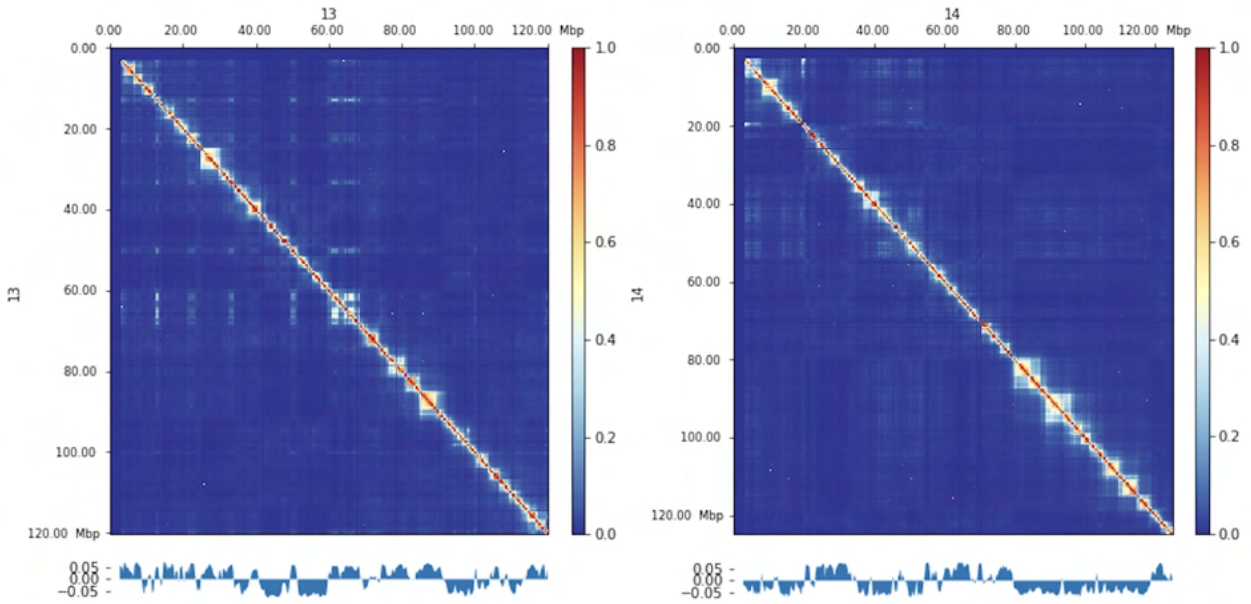


Fig.34. A/B compartments binned at 250 kb resolution, chr. 13, 14 for neurons.

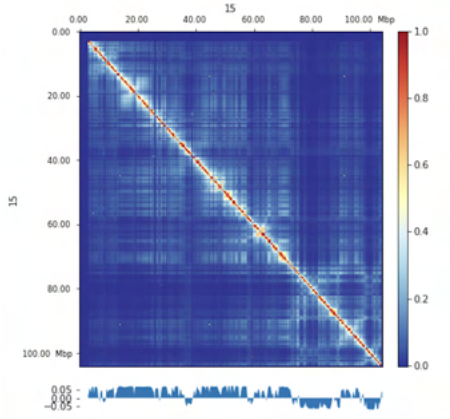


Fig.35. A/B compartments binned at 250 kb resolution, chr. 15 for glia.

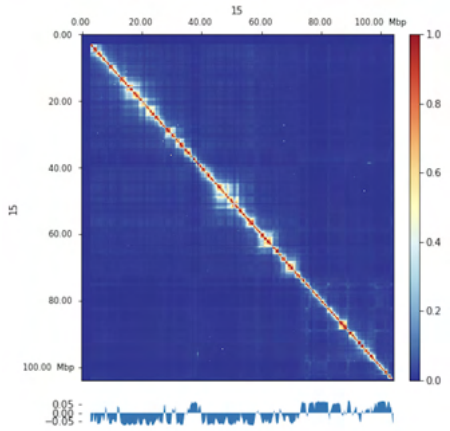


Fig.36. A/B compartments binned at 250 kb resolution, chr. 15 for neurons.

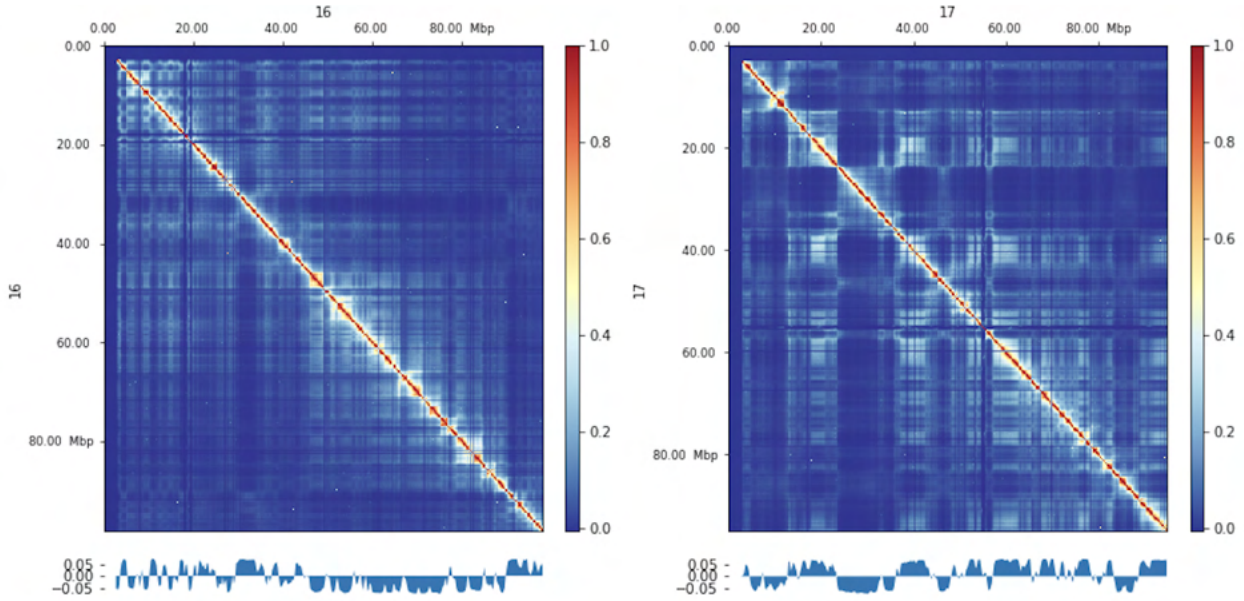


Fig.37. A/B compartments binned at 250 kb resolution, chr. 16, 17 for glia.

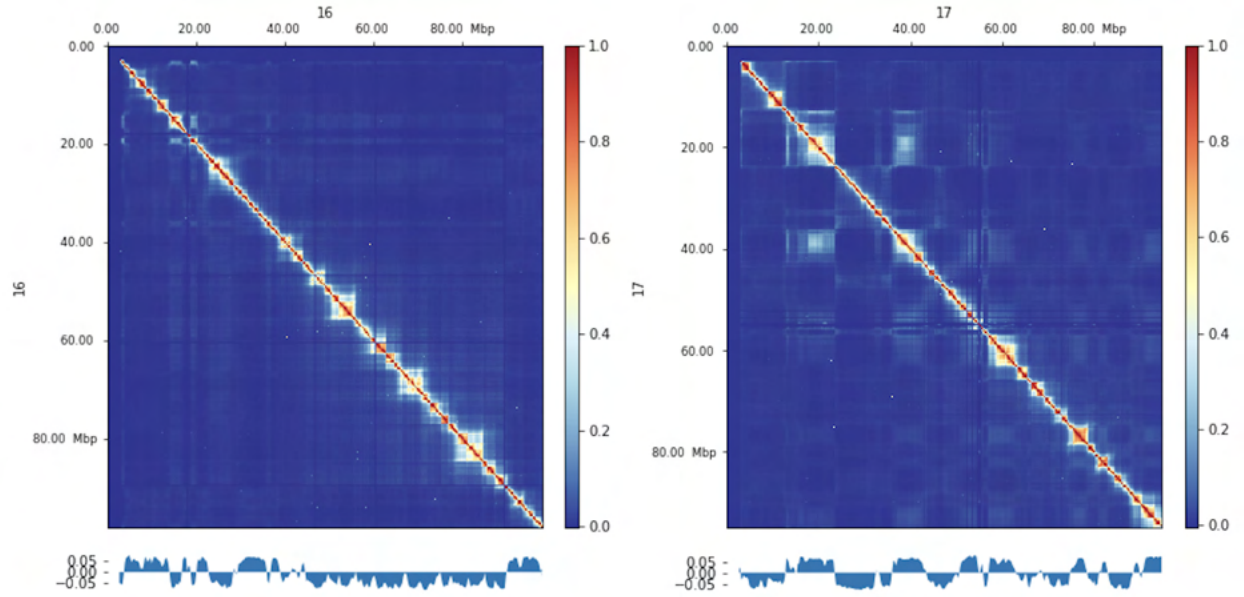


Fig.38. A/B compartments binned at 250 kb resolution, chr. 16, 17 for neurons.

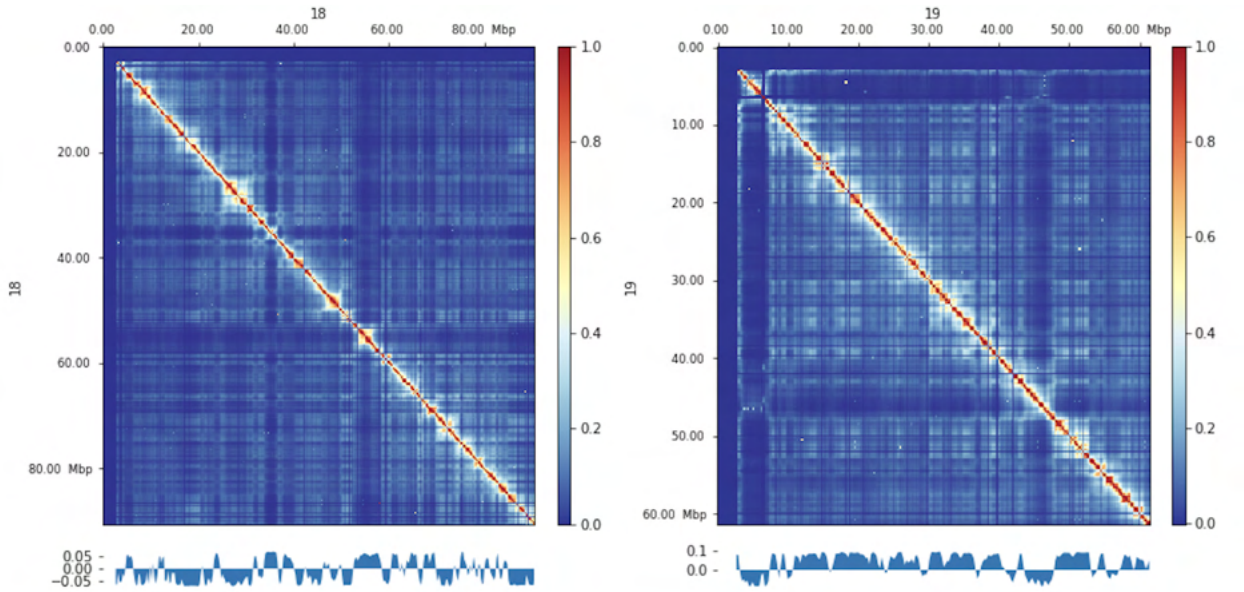


Fig.39. A/B compartments binned at 250 kb resolution, chr. 18, 19 for glia.

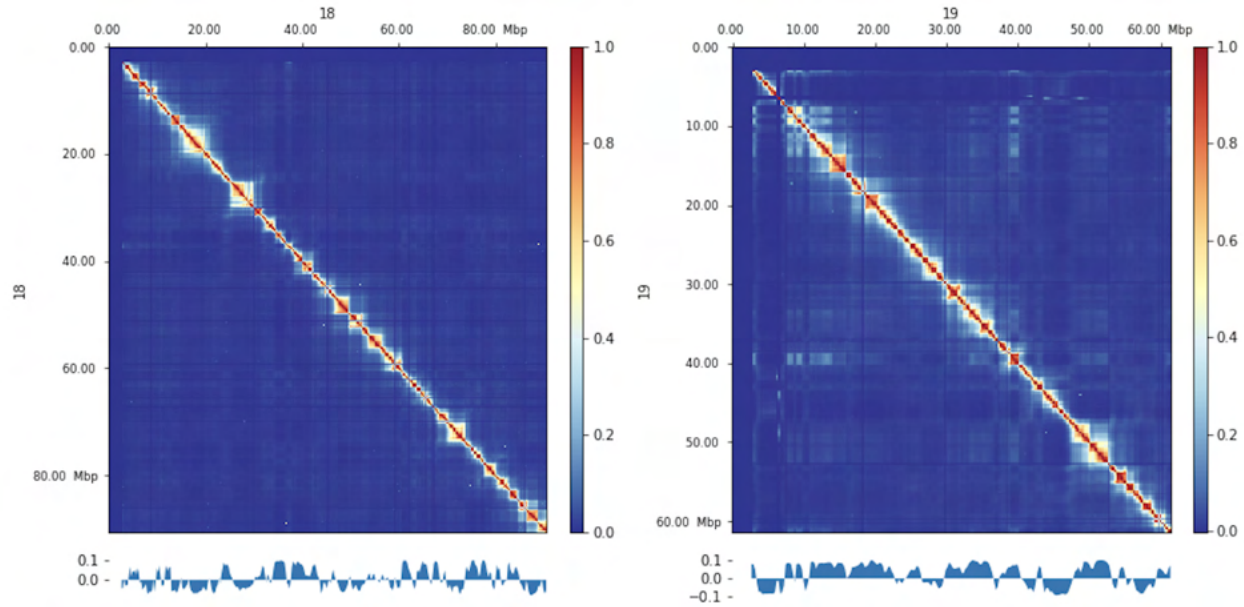


Fig.40. A/B compartments binned at 250 kb resolution, chr. 18, 19 for neurons.

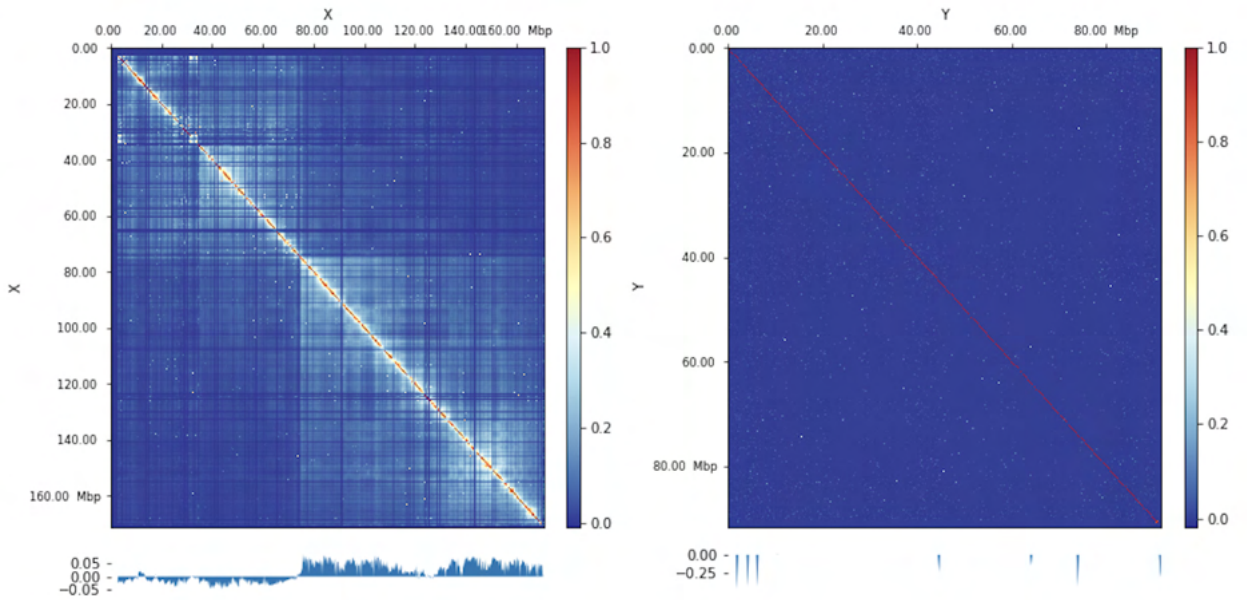


Fig.41. A/B compartments binned at 250 kb resolution, chr. X, Y for glia.

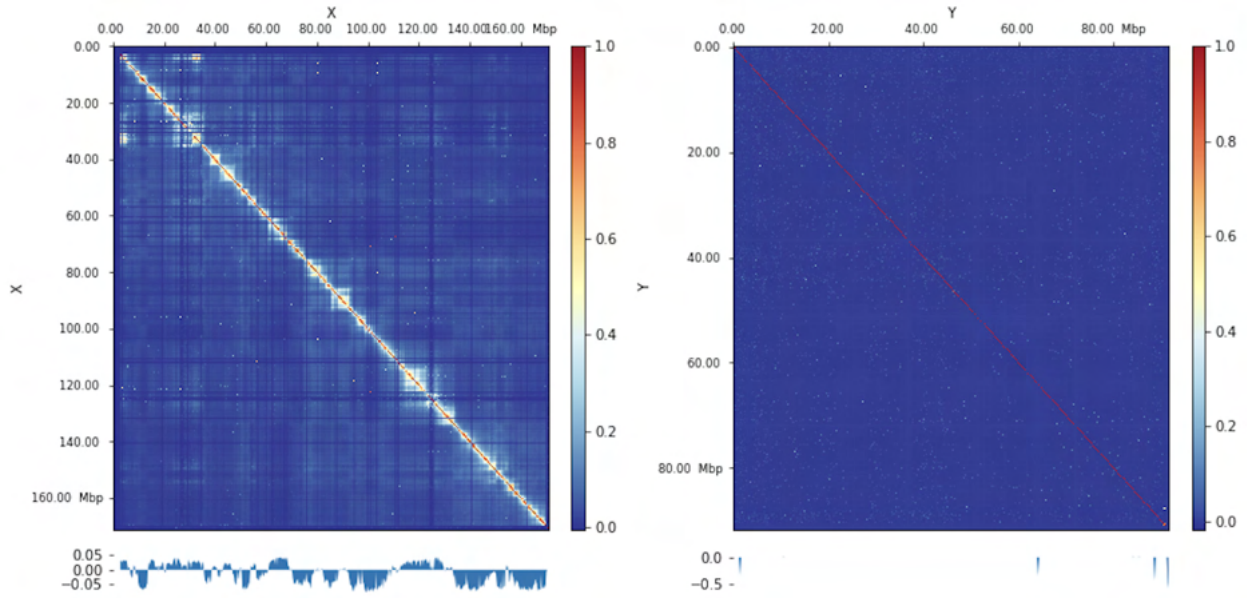


Fig.42. A/B compartments binned at 250 kb resolution, chr. X, Y for neurons.

All in all, it could be seen that glia Hi-C data displays stronger compartment pattern than Hi-C data of neurons.

It is also could be seen that Hi-C data of neurons demonstrates stronger TADs compartment pattern than glia Hi-C data.

chr.Y does not demonstrate any interactions at all for either glia and neurons.

Loops visualization

Detect enriched interaction regions (loops) at **25 kb resolution** (I used 25 kb resolution and not 10 kb because ‘hicDetectLoops’ was not able to find any loops at 10 kb resolution):

```
hicDetectLoops -m ${file.mcool}::/resolutions/25000 -o ${loops.bedgraph}
--maxLoopDistance 2000000 --windowSize 10 --peakWidth 6
--pValuePreselection 0.05 --pValue 0.05
```

2884 loops were detected for glia, 3089 loops were detected for neurons.

Plot loops:

```
hicPlotMatrix -m ${file.mcool}::/resolutions/25000
-o plot.png --log1p --region 1:10000000-20000000 --loops ${loops.bedgraph}
```

Visualization of loops for glia:

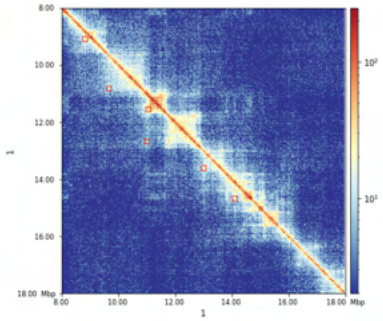


Fig.43. Loops binned at 25 kb resolution, chr.1: 8-18 Mb, for glia.

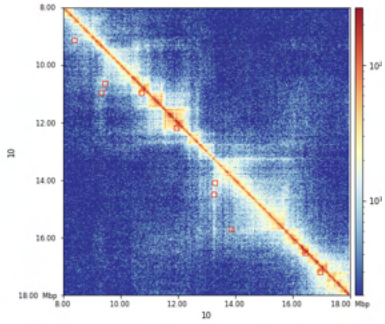


Fig.44. Loops binned at 25 kb resolution, chr.10: 8-18 Mb, for glia.

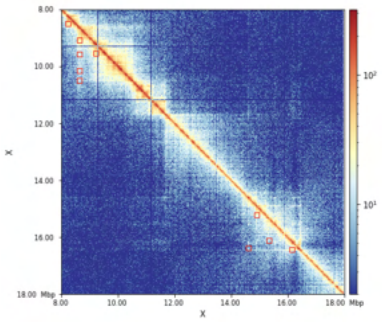


Fig.45. Loops binned at 25 kb resolution, chr.X: 8-18 Mb, for glia.

Visualization of loops for neurons:

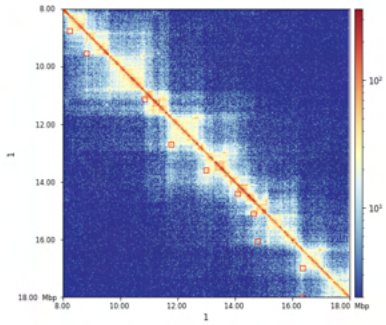


Fig.46. Loops binned at 25 kb resolution, chr.1: 8-18 Mb, for neurons.

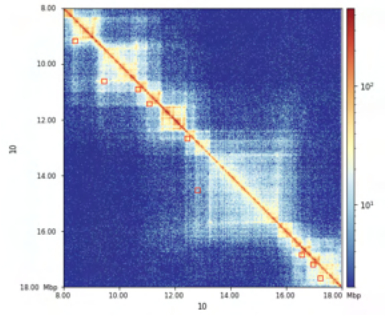


Fig.47. Loops binned at 25 kb resolution, chr.10: 8-18 Mb, for neurons.

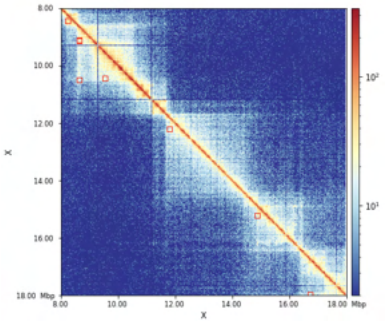


Fig.48. Loops binned at 25 kb resolution, chr.X: 8-18 Mb, for neurons.

In general, it could be concluded that the amount of loops is almost the same for both neurons and glia.

Conclusions:

1. Amount of loops is almost the same for both neurons and glia.
2. Glia Hi-C data displays stronger compartment pattern than Hi-C data of neurons.
3. Hi-C data of neurons demonstrates stronger TADs compartment pattern than glia Hi-C data.

References:

RNA-seq

To find correlation between density of chromatin structures (TADs, compartments) and gene expression we performed differential gene expression analysis.

Single-cell RNA-seq data from 8 mice (20-30 months old) was used for the analysis. The data was taken from the following article: https://www.nature.com/articles/s41593-019-0491-3 .

Files were already preprocessed by the authors of paper. Cell Ranger (version 1.2) (10X Genomics) was used to perform sample de-multiplexing, barcode processing and single cell gene unique molecular identifier (UMI) counting, while a digital expression matrix was obtained for each experiment with default parameters, mapped to the 10x reference for mm10, version 1.2.0. After the initial sequencing, the samples in each pool were re-pooled based on the actual number of cells detected by Cell Ranger, aiming to sequence each sample to a similar depth (number of reads/cell) (median: 40,007). Multiple NextSeq runs were conducted to achieve over 70% sequencing saturation as determined again by Cell Ranger (median: 75%).

Data from eight samples was integrated together. The data contained the following cell types: oligodendrocyte precursor cells (OPC), oligodendrocytes (OLG), olfactory ensheathing glia (OEG), neural stem cells (NSC), astrocyte-restricted precursors (ARP), astrocytes (ASC), neuronalrestricted precursors (NRP), immature neurons (ImmN), mature neurons (mNEUR), neuroendocrine cells (NendC), ependymocytes (EPC), hypendymal cells (HypEPC), tanycytes (TNC), choroid plexus epithelial cells (CPC), endothelial cells (EC), pericytes (PC), vascular smooth muscle cells (VSMC), hemoglobin-expressing vascular cells (Hb-VC), vascular and leptomeningeal cells (VLMC), arachnoid barrier cells (ABC), microglia (MG), monocytes (MNC), macrophages (MAC), dendritic cells (DC) and neutrophils (NEUT). For better analysis only glial, neuronal and precursor cells were kept. Each cell was assigned to one of two clusters: neurones or non neuronal cells. Neurones: neuronalrestricted precursors (NRP), immature neurons (ImmN), mature neurons (mNEUR). Non neuronal cell: oligodendrocyte precursor cells (OPC), oligodendrocytes (OLG), olfactory ensheathing glia (OEG), neural stem cells (NSC), astrocyte-restricted precursors (ARP), astrocytes (ASC), ependymocytes (EPC), hypendymal cells (HypEPC), tanycytes (TNC), choroid plexus epithelial cells (CPC), arachnoid barrier cells (ABC), microglia (MG).

```
sc_OX2X <- read.table("GSM3722109_OX2X_10X.txt")
sc_OX1X <- read.table("GSM3722108_OX1X_10X.txt")
sc_OX3X <- read.table("GSM3722110_OX3X_10X.txt")
sc_OX4X <- read.table("GSM3722111_OX4X_10X.txt")
sc_OX5X <- read.table("GSM3722112_OX5X_10X.txt")
sc_OX6X <- read.table("GSM3722113_OX6X_10X.txt")
sc_OX7X <- read.table("GSM3722114_OX7X_10X.txt")
sc_OX8X <- read.table("GSM3722115_OX8X_10X.txt")
sco <- cbind(sc_OX1X, sc_OX2X, sc_OX3X, sc_OX4X, sc_OX5X, sc_OX6X, sc_OX7X, sc_OX8X)
sco <- subset(sco, sco$cell !='NendC'& sco$cell !='NEUT' & sco$cell !='Hb-VC' & sco$cell !='MNC' & sco$cell !='MAC' & sco$cell !='DC' & sco
meta <- subset(meta, meta$V5 == 'old')
sco <- as.data.frame(t(sco))
sco <- cbind(sco, cell = meta$V4)
sco$cluster <- sco$cell
sco <- sco%>%mutate(cluster = case_when(cluster == 'NSC' ~ 'Neuro',cluster == 'mNEUR' ~ 'Neuro', cluster == 'ImmN' ~ 'Neuro', T ~ as.charac
sco <- sco%>%mutate(cluster = case_when(cluster != 'Neuro' ~ 'nonNeuro', T ~ as.character(cluster)))
sco <-  sco %>% relocate(cell, .before = Sox17)
sco <-  sco %>% relocate(cluster, .before = Sox17)
```

Dataset contained zero values, which were substituted with 1/10 of minimal value for each gene.

```
sco[, 1:14699] <- lapply(sco[, 1:14699], function(x) (x == 0) * min(x[x != 0])/10 + x)
```

Next differential expression analysis was performed using Mann-Whitney test. Obtained p-values were corrected for multiple comparison using fdr.

```
variables<-colnames(sco)[4:dim(sco)[2]]
pvals <- {}
vars <- {}
for (i in variables) {
  res <- wilcox.test(sco[,i] ~ cluster,
    data = sco,
    exact = FALSE)
pvals <- c(pvals, round(res$p.value, 10))
vars <- c(vars, i)
}
res_df <- data.frame(Genes = vars, pvalues = pvals)
res_df$p_adj <- p.adjust(res_df$pvalues, method = "fdr")
```

Log2 fold change was calculated by calculating log2 mean expression for each gene in neurones and in non neuronal cells and then subtracting these means.

```
sco_nonneuro <- subset(sco, sco$cluster == "nonNeuro")
sco_nonneuro <- as.data.frame(t(sco_nonneuro))
sco_nonneuro <- sco_nonneuro[-1, ]
sco_nonneuro <- sco_nonneuro %>% mutate_at(1:15312, as.numeric)
sco_nonneuro$mean <- rowMeans(sco_nonneuro)

sco_neuro <- subset(sco, sco$cluster == "Neuro")
sco_neuro <- as.data.frame(t(sco_neuro))
sco_neuro <- sco_neuro[-1, ]
sco_neuro <- sco_neuro %>% mutate_at(1:3184, as.numeric)
sco_neuro$mean <- rowMeans(sco_neuro)

res_df <- cbind(res_df, mean_neuro=sco_neuro$mean)
res_df <- cbind(res_df, mean_nonneuro=sco_nonneuro$mean)
res_df[, 4] <- log(res_df[4], 2)
res_df[, 5] <- log(res_df[5], 2)
res_df$lgfc <- res_df$mean_neuro - res_df$mean_nonneuro
```

In order to find out what is difference in gene expression between two groups of cells functional analysis was performed. Two gene sets were created - genes unregulated in neurones compared to non neuronal cells and genes down regulated in neurones compared to non neuronal cells.

```
res_df_up <- subset(res_df, res_df$lgfc > 0)
res_df_down <- subset(res_df, res_df$lgfc < 0)
res_df_up_sig <- subset(res_df_up, res_df_up$p_adj < 0.05)
res_df_down_sig <- subset(res_df_down, res_df_down$p_adj < 0.05)
```

For both gene sets gene ontology term enrichment was carried out.

```
ego_up <- enrichGO(gene = res_df_up_sig$Genes,
  universe = res_df$Genes,
  keyType = "SYMBOL",
  OrgDb = org.Mm.eg.db,
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05)

barplot(ego_up, showCategory=10)

ego_down <- enrichGO(gene = res_df_down_sig$Genes,
                     universe = res_df$Genes,
                    keyType = "SYMBOL",
                     OrgDb = org.Mm.eg.db,
                    ont = "BP",
                     pAdjustMethod = "BH",
                    pvalueCutoff = 0.05)

barplot(ego_down, showCategory=10)
```

Genes upregulated in neurones are enriched in some processes related to chromatin organisation so we can expect to find differences in chromatin organisation between neurones and glia. Gene sets downregulated in neurones are enriched in some immune related processes.

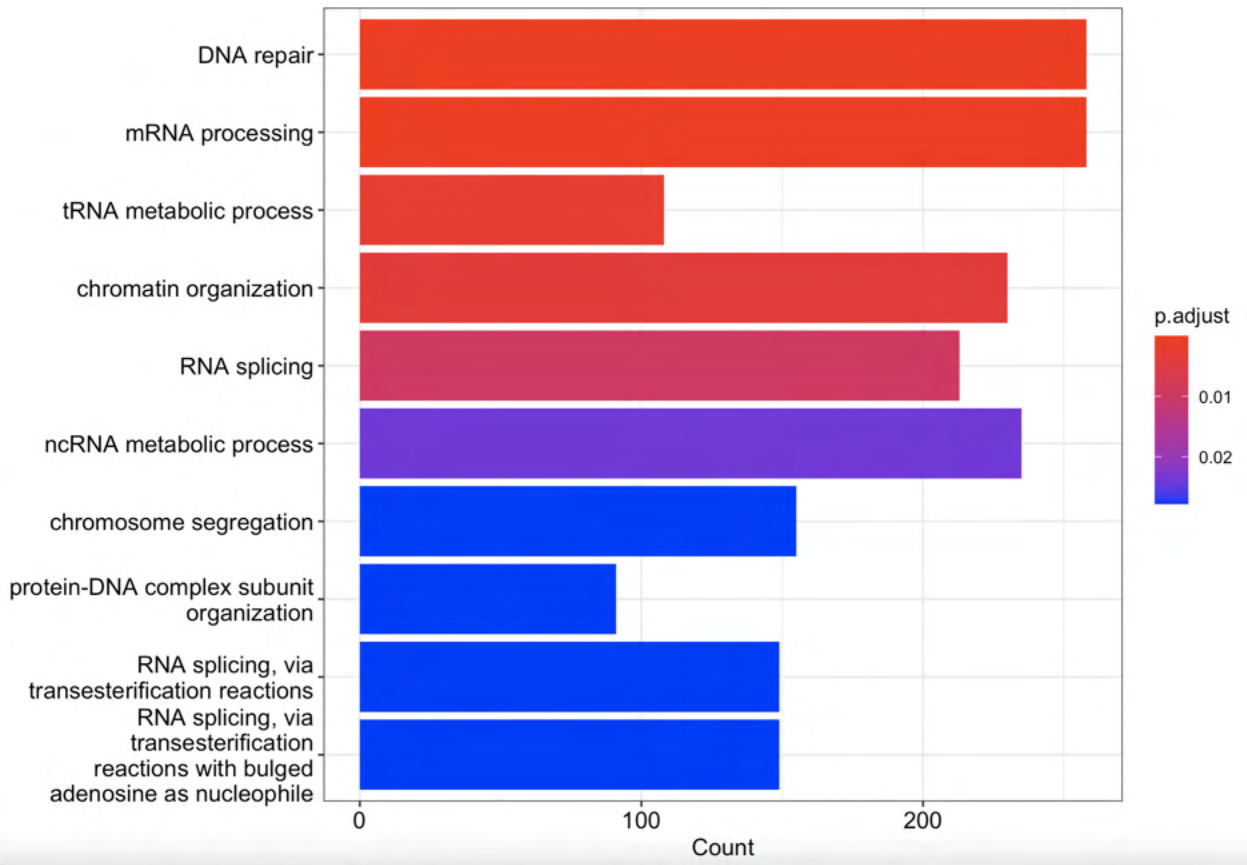


Fig.1 GO enrichment results for genes upregulated in neurones.

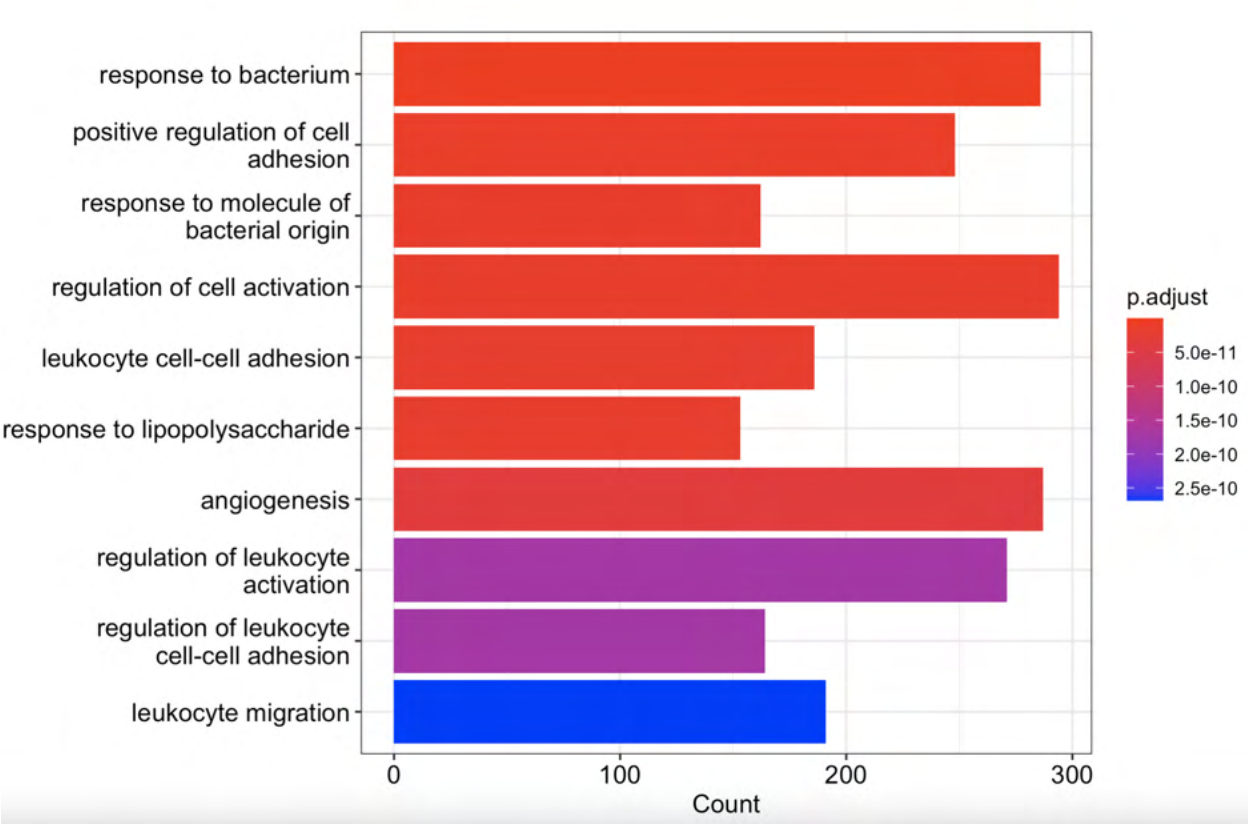


Fig.2 GO enrichment results for genes downregulated in neurones.

Based on this results we can expect to see correlation between cell type and chromatin structures. Further analysis will be carried out including gene set enrichment analysis.

References:

1. Ximerakis, M., Lipnick, S.L., Innes, B.T. et al. Single-cell transcriptomic profiling of the aging mouse brain. Nat Neurosci 22, 1696–1708 (2019). https://doi.org/10.1038/s41593-019-0491-3 2. Wu T, Hu E, Xu S, Chen M, Guo P, Dai Z, Feng T, Zhou L, Tang W, Zhan L, Fu x, Liu S, Bo X, Yu G (2021). “clusterProfiler 4.0: A universal enrichment tool for interpreting omics data.” *The Innovation*, **2**(3), 100141. doi:10.1016/j.xinn.2021.100141.

Detection of polycomb regions in neuronal cells

It would be interesting to look how differently the polycomb bodies are located at chromatin of neuronal and glial cells. These bodies are responsible for the transcriptional silencing and they are consisted of PcG proteins. For this task one can obtain the H3K27me3 marks (they indicate the proteins which constitute the polycomb bodies) and to integrate it with the data with the statistically significantly interacting chromatin contacts. Also, with the H3K27ac mark it is possible to do quite reverse task: to look at the active chromatin region, which surely does not interact with the polycomb proteins. The former part (H3K27me3 mark) of the analysis can be done with the Chip-seq analysis (it is described below in Chip-Seq analysis), and for the latter one can be done with a Fit-Hi-C2

tool. [1] This application was designed for “finding inter-chromosomal significant interactions” and “reporting the expected contact count between interacting pairs of bins.” [https://github.com/ay-lab/fithic]

For this analysis we have HiC data for both neuronal and glial cells, but unfortunately, we did not find the available H3K27me3 data for glial cells. So, we are going to identify the chromatin regions associated with polycomb bodies only for neuronal cells. Although, we are also going to integrate the H3K27ac mark with the information about TADs.

The HiC and Chip-Seq data were taken from the given to us article, and we took it from NCBI’s Gene Expression Omnibus (GSE168524). The code for this part can be found here below and in out Github repository. The analysis was performed in Arkuda Skoltech cluster.

I. Significantly interacting contacts

To find the significantly interacting contacts in chromatin, the following tools were used:

> Python 3.10.11
>
> pandas v. 2.0.1
>
> numpy v. 1.24.3
>
> cooler 0.9.1
>
> FitHiC v. 2.0.8
>
> hic2cool v. 0.8.3
>
> coolpuppy v. 1.0.0

The most time was spent for the creation and the setting of conda environments.

Firstly, the HiC data file (`SE168524_neurons_fem_wt_allValidPairs.hic`) was converted into the *.cool* format with the following command:

```
hic2cool convert \
/gss/home/l.sidorov/omics_final_project/data/GSE168524_neurons_fem_wt_allValidPairs.hic \
/gss/home/l.sidorov/omics_final_project/GSE168524_neurons_fem_wt_allValidPairs.mcool \
-p 1
```

Fit-Hi-C tool requires two mandatory files as an input: “fragments” and “interactions”. For this, with the help of our TA, Anna Kononkova, the following python script was written and saved in the `interactions_and_fragments_file.py`:

```
import numpy as np
import pandas as pd
import scipy
import cooler


resol=100000
c=cooler.Cooler('/gss/home/l.sidorov/omics_final_project/GSE168524_neurons_fem_wt_allValidPairs.mcool::resolutions/%s'%resol)

pix=c.pixels(join=True)[:]

pix.chrom1=pix.chrom1.astype(str)
pix.chrom2=pix.chrom2.astype(str)


pix1=pix.groupby(['chrom1','start1']).sum().reset_index()[['chrom1','start1','count']]
pix=pix.loc[~((pix.chrom1==pix.chrom2)&(pix.start1==pix.start2))]

pix=pix.groupby(['chrom2','start2']).sum().reset_index()[['chrom2','start2','count']]
pix1.columns=[0,1,2]
pix.columns=[0,1,2]


pix=pd.concat([pix1,pix])

pix1=None
```

```
pix=pix.groupby([0,1]).sum().reset_index()

pix['lab']=pix[0].astype(str)+':'+pix[1].astype(str)

chr_size=pd.read_csv('chrsizes.csv',sep='\t',index_col=0)

bins=cooler.binnify(chr_size['size'],resol)
bins.chrom=bins.chrom.astype(str)

bins['lab']=bins.chrom.astype(str)+':'+bins.start.astype(str)
bins.index=bins.lab

bins['count']=0
bins.loc[list(pix.lab),'count']=list(pix[2])

bins['x1']=0
bins['x2']=1

pix=None

fragments=bins[['chrom','x1','start','count','x2']]
fragments.sort_values(['chrom','start'],inplace=True)
fragments.to_csv('./fithic_inputs/fragments.gzip',sep='\t',
                      header=None,index=None,compression='gzip')

pix=c.pixels(join=True)[:]
interactions=pix[['chrom1','start1','chrom2','start2','count']]
interactions.to_csv('./fithic_inputs/interactions.txt.gzip',sep='\t',header=None,index=None,compression='gzip')
```

Then, we launched the script and it gave us two files: `interactions.txt.gzip` and `fragments.gzip`. Using them and script provided by the developers of Fit-Hi-C2, we generated another needed input file: a bias file which was calculated with Knight-Ruiz normalization. It was done with the following command:

```
python3 ../../fithic/fithic/utils/HiCKRy.py -i ./fithic_inputs/interactions.txt.gzip \
-f ./fithic_inputs/fragments.gzip -o ./fithic_inputs -x 0.05
```

Then, we finally launched the Fit-Hi-C2 tool:

```
fithic -i ./fithic_inputs/interactions.txt.gzip -f ./fithic_inputs/fragments2.gzip \
-o ../output_fithic/ -r 100000 -t ./fithic_inputs/bias_file
```

It gave us three files: `FitHiC.fithic.log`, `FitHiC.fithic_pass1.res100000.txt`, `FitHiC.spline_pass1.res100000.significances.txt.gz`. For the third one, we filtered it for significant contacts: *q-value* is lower than 0.05; and the contacts wich have the distance between no lower than two bins (200 000). After the filtering, we obtained **444 142** significant chromatin contacts.

The, with the following script we tried to visualise the results of these significant contacts using the `coolpuppy` tool: [2]

```
import matplotlib as mpl
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

from coolpuppy import coolpup
from coolpuppy.lib import numutils
from coolpuppy import plotpup
import cooler
import bioframe
import cooltools
from cooltools import expected_cis, expected_trans
from cooltools.lib import plotting

resolution = 100000
clr = cooler.Cooler(f'./GSE168524_neurons_fem_wt_allValidPairs.mcool::/resolutions/{resolution}')

df = pd.read_csv('./filtered_sig_contacts.csv', header=0, sep = '\t', low_memory=False)

# here I created .bed file from fithic output, because coolpuppy requires one
```

```
df2 = df[['chr1', 'fragmentMid1', 'fragmentMid2']]
header = ['chrom', 'start', 'end']
df2.columns = header[:len(df2.columns)]

df2.to_csv('xxx.bed', index=False, sep='\t', header=None)

# and with this command we launched the coolpup.py script to visualise the results

!coolpup.py ./GSE168524_neurons_fem_wt_allValidPairs.mcool::/resolutions/100000 ./xxx.bed --clr_weight_name -p 2
```

However, due to `--clr_weight_name` flag, during the writing of this report we had not got the result of this command yet.

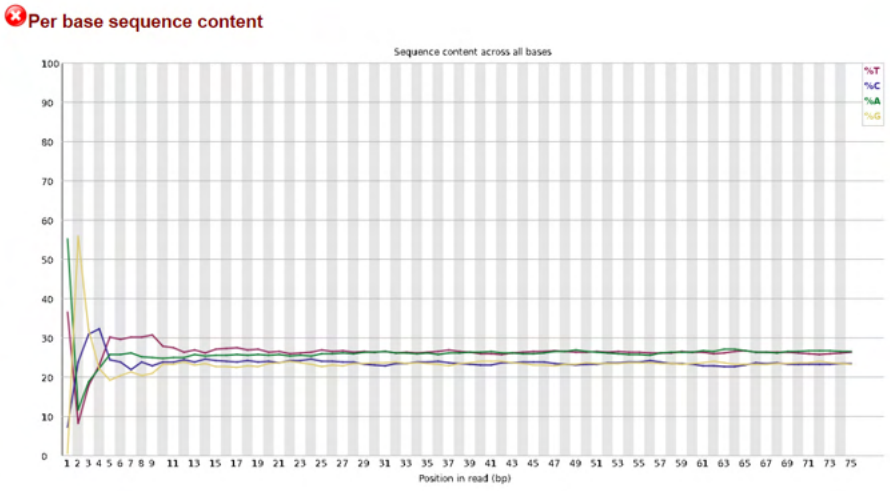
II. H3K27ac mark

We also managed to launch the nextflow pipeline (https://nf-co.re/chipseq) with the following command:

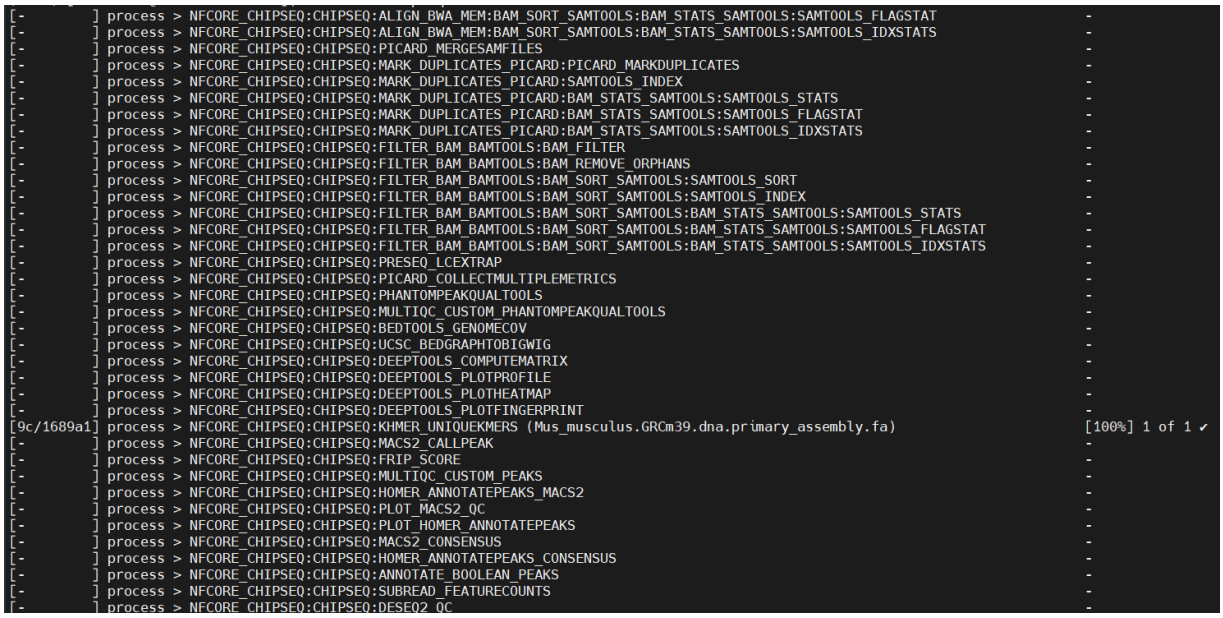
```
nextflow run ./chipseq/workflow/ --input samplesheet.csv --outdir out_chipseq \
--fasta /gss/home/l.sidorov/omics_final_project/omics_da_final_mbe/H3K27ac/Mus_musculus.GRCm39.dna.primary_assembly.fa.gz \
--gtf /gss/home/l.sidorov/omics_final_project/omics_da_final_mbe/H3K27ac/Mus_musculus.GRCm39.109.gtf.gz \
-profile singularity \
--read_length 75 \
--max_cpus 1 # on Arkuda only two CPUs were available at that moment
```

The following accesions were used in samplesheet: neuronal cells — SRR13911720, SRR13911721, SRR13911722; glial cells — SRR13911723, SRR13911724, SRR13911725.

We are still waiting for the output of the pipeline. It already gave us fastqc reports (sequencing was performed well, almost the point are green, except for Per base sequence content (below the figure contributes to the R1 from SRR13911720)):



Still, not much was performed yet, but it is running already for two days till this moment.



1. Kaul, A., Bhattacharyya, S. & Ay, F. Identifying statistically significant chromatin contacts from Hi-C data with FitHiC2. *Nat Protoc* **15**, 991–1012 (2020). https://doi.org/10.1038/s41596-019-0273-0

2. Ilya M Flyamer, Robert S Illingworth, Wendy A Bickmore (2020). Coolpup.py: versatile pile-up analysis of Hi-C data. Bioinformatics, 36, 10, 2980–2985.

ChiP-seq analysis

It is well-known that different types of neurons and glia displaying distinct gene expression and chromatin accessibility profiles. To address this issue, Hi–C may be used and its derivatives to build higher-order chromatin interaction maps. As Hi–C have been performed already we would like to see the results applying ChiP-seq analysis.

In this work we used the data obtained from in vitro differentiated neural cells and neural and non-neural cells collected from post-morten brain tissue of mice. The main difference is that *in vitro* cultured cells that mark early brain development compared to second ones [1, 2].

Here, we analyze epigenetic landscape at cellular resolution to capture how these types of cells are differed from each other. **The main goals of this part of our project:**

1. Investigating H3K9me3 tracks in neuronal and non-neuronal cells;
2. Investigating H3K27me3 profiles in in-vitro differentiated cortical neurons (CN) and in-vitro differentiated neural progenitors cells (NPC);
3. Investigating Ring1B profiles in in-vitro differentiated cortical neurons (CN) and in-vitro differentiated neural progenitors cells (NPC);

Investigating H3K9me3 tracks in neuronal and non-neuronal cells

Input data: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE168524. Raw reeds of the sorted cells (NeuN+ and NeuN-) obtained from mouse brain.

For NeuN+:

GSM5145419 WT NeuN+ H3K9me3 REP1 (SRR13911706)

GSM5145420 WT NeuN+ H3K9me3 REP2 (SRR13911707)

GSM5145421 WT NeuN+ H3K9me3 REP3 (SRR13911708)

GSM5145422 WT NeuN+ H3K9me3 REP4 (SRR13911709)

For NeuN-:

GSM5145423 WT NeuN- H3K9me3 REP1 (SRR13911710)

GSM5145424 WT NeuN- H3K9me3 REP2 (SRR13911711)

GSM5145425 WT NeuN- H3K9me3 REP3 (SRR13911712)

GSM5145426 WT NeuN- H3K9me3 REP4 (SRR13911713)

Downloading the data:

```
fastq-dump --gzip --split-files SRR13911706 SRR13911707 SRR1311708 SRR13911709
fastq-dump --gzip --split-files SRR13911710 SRR13911711 SRR13911712 SRR13911713
```

Mapping reads:

As a reference genome we took assembly of the mouse genome (mm39, Genome Reference Consortium Mouse Build 39 (GCA_000001635.9)). Link for downloading: https://hgdownload.soe.ucsc.edu/goldenPath/mm39/bigZips/mm39.fa.gz

```
# for paired-end data
bowtie2 -p 4 -x ~/project_10/data/bowtie2/mm39 -1 ~/project_10/data/download20230501/<SRA-id>_1.fastq.gz -2 ~/project_10/data/download20230
```

Converting .sam to .bam:

```
samtools view -S -b <SRA-id>.sam > <SRA-id>.bam
```

Sorting .bam files:

```
# sort bam file by position
samtools sort -o <SRA-id>.sorted.bam <SRA-id>.bam
```

Merging replicates in one .bam file:

We have got 4 replicates for NeuN+ and NeuN- cell types and needed to merge them in case of lots of replicates.

```
samtools merge -X NeuN+_H3K9me3_allReps.bam SRR13911706.sorted.bam SRR13911707.sorted.bam SRR13911708.sorted.bam SRR13911709.sorted.bam
samtools merge -X NeuN-_H3K9me3_allReps.bam SRR13911710.sorted.bam SRR13911711.sorted.bam SRR13911712.sorted.bam SRR13911713.sorted.bam
```

Marking read duplicates:

```
# run picard MarkDuplicates
PICARD=/usr/local/share/java/picard.jar
java -jar $PICARD MarkDuplicates -I <SRA-id>.sorted.bam -O <SRA-id>.markdup.bam -M <SRA-id>marked_dup_metrics.txt --REMOVE_DUPLICATES false
```

Filtering out "bad" reads:

```
# for paired-end data
samtools view -f 2 -F 1024 -q 30 -b <SRA-id>.markdup.bam > <SRA-id>.filt.bam
```

Indexing .bam files:

```
samtools index ~/project_10/data/filtered/<SRR_ID>.filt.bam
```

Creating a ChiP-seq tracks:

```
bamCoverage --bam ~/project_10/data/filtered/<SRR_ID>.filt.bam -o <SRR_ID>.filt.bw --binSize 10 --normalizeUsing RPGC --effectiveGenomeSize
```

In this article we do not have input files that is why we would not perform peak calling. We **have got two bigWig files** (NeuN+_H3K9me3_allReps.bw and NeuN-_H3K9me3_allReps.bw) which we will compare to each other [3].

Investigating H3K27me3 profiles in *in-vitro* differentiated cortical neurons (CN) and *in-vitro* differentiated neural progenitors cells (NPC)

Input data: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE96107. The cells were obtained from mouse and were differentiated in cortical neurons (CN) and neural progenitor cells (NPC) *in vitro*.

For CN:

GSM2533888 ChIP_CN_H3K27me3_1 (SRR5339985)

GSM2533889 ChIP_CN_H3K27me3_2 (SRR5339986)

GSM2533892 ChIP_CN_Input_1 (SRR5339989)

GSM2533893 ChIP_CN_Input_2 (SRR5339990)

For NPC:

GSM2533870 ChIP_NPC_H3K27me3_1 (SRR5339967)

GSM2533871 ChIP_NPC_H3K27me3_2 (SRR5339968)

GSM2533874 ChIP_NPC_Input_1 (SRR5339971)

GSM2533875 ChIP_NPC_Input_2 (SRR5339972)

Downloading the data:

```
fastq-dump --gzip --split-files SRR5339985 SRR5339986 SRR5339989 SRR5339990
fastq-dump --gzip --split-files SRR5339967 SRR5339968 SRR5339971 SRR5339972
```

Mapping reads:

As a reference genome we also took assembly of the mouse genome (mm39, Genome Reference Consortium Mouse Build 39 (GCA_000001635.9)). Link for downloading: https://hgdownload.soe.ucsc.edu/goldenPath/mm39/bigZips/mm39.fa.gz

```
# for paired-end data
bowtie2 -p 4 -x ~/project_10/data/bowtie2/mm39 -1 ~/project_10/data/download20230501/<SRA-id>_1.fastq.gz -2 ~/project_10/data/download20230
```

Converting .sam to .bam:

```
samtools view -S -b <SRA-id>.sam > <SRA-id>.bam
```

Marking read duplicates:

```
# sort sam file by position
samtools sort -o <SRA-id>.sorted.bam <SRA-id>.bam

# run picard MarkDuplicates
PICARD=/usr/local/share/java/picard.jar
java -jar $PICARD MarkDuplicates -I <SRA-id>.sorted.bam -O <SRA-id>.markdup.bam -M <SRA-id>marked_dup_metrics.txt --REMOVE_DUPLICATES false
```

Filtering out "bad" reads:

```
# for paired-end data
samtools view -f 2 -F 1024 -q 30 -b <SRA-id>.markdup.bam > <SRA-id>.filt.bam
```

Indexing .bam files:

```
samtools index ~/project_10/data/filtered/<SRR_ID>.filt.bam
```

Peak calling:

```
macs2 callpeak -t <name of experiment .bam file> \
-c <name of input .bam file> \
-f BAM -g 2652783500 -q 0.05 \
-n <name for the output peak file> \
--outdir <name of folder to store outputs>
```

For this data we have already had input files for each replicates and bigWig files, so, we could perform peak calling and we did not perform ChiP-seq track. **As a result we obtained two outputs for *in-vitro* differentiated neural progenitors cells**

(ChIP_NPC_H3K27me3_1_model.r, ChIP_NPC_H3K27me3_1_summits.bed, ChIP_NPC_H3K27me3_1_peaks.narrowPeak, ChIP_NPC_H3K27me3_1_peaks.xls, ChIP_NPC_H3K27me3_2_model.r, ChIP_NPC_H3K27me3_2_summits.bed, ChIP_NPC_H3K27me3_2_peaks.narrowPeak, ChIP_NPC_H3K27me3_2_peaks.xls).

For cortical neurons data the commands are still running. Some of them were failed due to no space was left on server and this process is time-consuming.

Investigating Ring1B profiles in *in-vitro* differentiated cortical neurons (CN) and in-vitro differentiated neural progenitors cells (NPC)

For CN:

GSM2533878 ChIP_CN_Ring1B_1 (SRR5339975)
GSM2533879 ChIP_CN_Ring1B_2 (SRR5339976)

GSM2533892 ChIP_CN_Input_1 (SRR5339989)

GSM2533893 ChIP_CN_Input_2 (SRR5339990)

For NPC:

GSM2533860 ChIP_NPC_Ring1B_1 (SRR5339957)
GSM2533861 ChIP_NPC_Ring1B_2 (SRR5339958)

GSM2533874 ChIP_NPC_Input_1 (SRR5339971)

GSM2533875 ChIP_NPC_Input_2 (SRR5339972)

The whole process of data processing for investigating Ring1B profiles the same as for H3K27me3 data analysis.

For this data we have already also had input files for each replicates and bigWig files, so, we could perform peak calling and we did not perform ChiP-seq track. Due to time-consuming processes the output files are still not obtained but in progress.

References:

1. Hu, B., Won, H., Mah, W. et al. Neuronal and glial 3D chromatin architecture informs the cellular etiology of brain disorders. Nat Commun 12, 3968 (2021). https://doi.org/10.1038/s41467-021-24243-0

2. Boshans, Linda L., et al. "The chromatin environment around interneuron genes in oligodendrocyte precursor cells and their potential for interneuron reprograming." Frontiers in Neuroscience 13 (2019): 829.

3. bigwigCompare — deepTools 3.5.0 documentation