Supplementary Material

Conv	**Act.**	**Conv**	**Act.**	**Conv**	**Act.**	**LSTM**	**Decis.**	**Corr.**
$64 \times 3 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$	relu	$8 \times 80 \times 1 \times 16$	relu	-	1	0.68
$64 \times 3 \times 1 \times 4$	exp.	$16 \times 8 \times 1 \times 64$	relu	$8 \times 80 \times 1 \times 16$	relu	-	1	0.67
$64 \times 3 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$	exp.	$8 \times 80 \times 1 \times 16$	relu	-	1	0.65
$64 \times 3 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$	relu	$8 \times 10 \times 1 \times 16$	relu	-	1	0.65
$64 \times 3 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$ $\text{dilation rate} : 2$	relu	$8 \times 20 \times 1 \times 16$ $\text{dilation rate} : 4$	relu	-	1	0.67
$64 \times 3 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$	relu	$8 \times 80 \times 1 \times 16$	relu	20	1	0.65
$64 \times 3 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$	relu	$8 \times 80 \times 1 \times 16$	relu	20	133	0.47
$64 \times 3 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$	relu	$8 \times 80 \times 1 \times 16$	relu	-	133	0.16
$64 \times 20 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$	relu	$8 \times 10 \times 1 \times 16$	relu	-	1	0.63
$64 \times 20 \times 1 \times 4$	relu	$16 \times 8 \times 1 \times 64$	relu	-	-	-	1	0.61
$64 \times 20 \times 1 \times 4$	relu	-	-	-	-	-	1	0.00

Table S1. Hyperparameters search. Several architecture were tested and compared using the correlation on the test set. The exponential activation is inspired from the work of Koo and Ploenzke 2020, the dilated convolution is inspired from the work of Kelley et al. 2018. The use of a final LSTM layer is inspired from the work of Di Gangi et al. 2018. Finally, predicting the nucleosome density covering a 133 bp long DNA sequence is inspired from Kelley et al. 2018 and Avsec et al. 2020.

References

Avsec Ž, Weilert M, Shrikumar A, Krueger S, Alexandari A, Dalal K, Fropf R, McAnany C, Gagneur J, Kundaje A, et al. 2020. Deep learning at base-resolution reveals cis-regulatory motif syntax. *bioRxiv.* 737981.

Di Gangi M, Bosco GL, and Rizzo R. 2018. Deep learning architectures for prediction of nucleosome positioning from sequences data. *BMC bioinformatics.* **19**: 418.

Kelley DR, Reshef YA, Bileschi M, Belanger D, McLean CY, and Snoek J. 2018. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research.* **28**: 739–750.

Koo PK and Ploenzke M. 2020. Improving representations of genomic sequence motifs in convolutional networks with exponential activations. *bioRxiv.*

Zhang J, Peng W, and Wang L. 2018. LeNup: learning nucleosome positioning from DNA sequences with improved convolutional neural networks. *Bioinformatics.* **34**: 1705–1712.

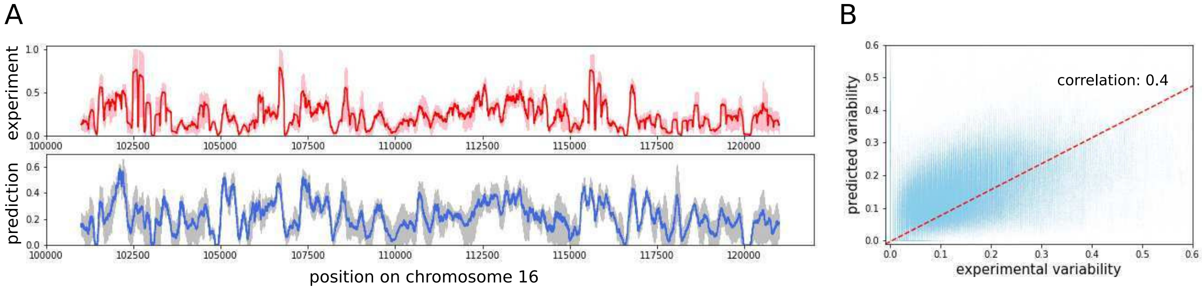

Figure S1. Predicted and experimental variability on four YPD NOCL replicates. (a) Maximal, mean and minimal experimental (red) and predicted (blue) density of nucleosome on a fragment of chromosome 16 for four YPD NOCL. The solid line represents the mean density and the width of the line represents the variability (max - min). (b) Scatter plot of the predicted variability at every point of chromosome 16 as a function of the experimental one. The variability is the difference between the maximal and the minimal nucleosome density for the four YPD NOCL replicates. The correlation between the variabilities is 0.4.

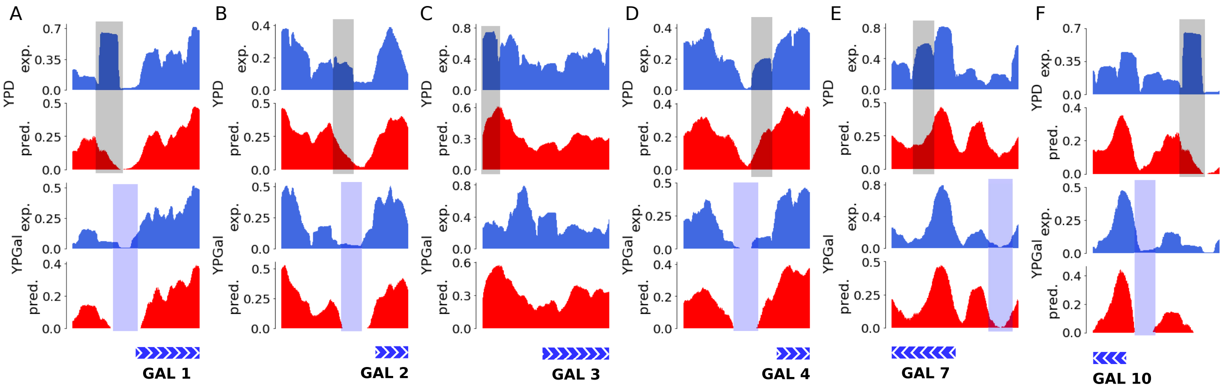

Figure S2. Predicted and experimental nucleosome density in a set of gal responsive promoter region. (a) Mean experimental density for four YPD NOCL replicates of the Kaplan's dataset (blue top), mean predicted density for four models independently trained on those replicates (red top), mean experimental density for two YPGal NOCL replicates (blue bottom) and mean predicted density for two models independently trained on those two replicates (red bottom) in the promoter region of GAL1. Gene position is sketched below the curves (blue rectangle with arrows corresponding to the strand). red YPGal specific enlargement of NDR are highlighted in blue, they are well predicted. YPD specific nucleosomes are highlighted in grey, they fail to be predicted. (b) Same for GAL2. (c) Same for GAL3. (d) Same for GAL4. (e) Same for GAL7. (f) Same for GAL10.

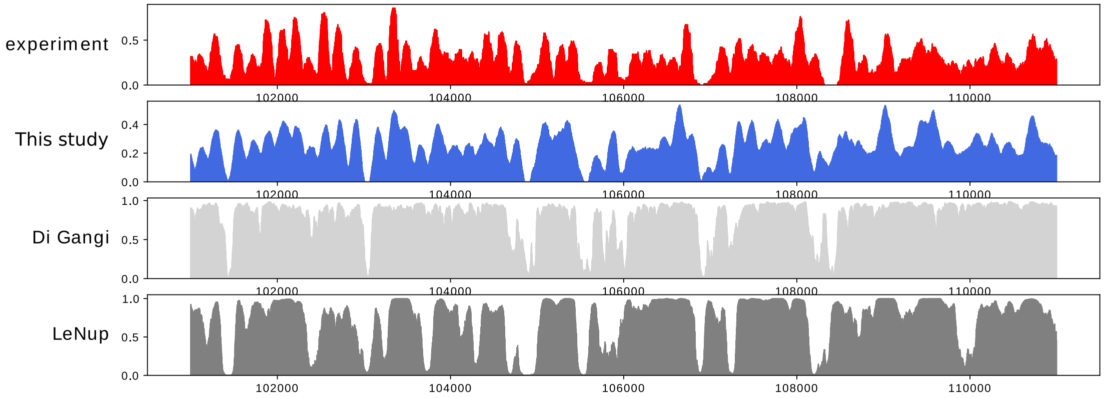

Figure S3. Comparison of deep learning methodologies on nucleosome density prediction. Comparison between the experimental density (red), the density predicted with our methodology (blue), with DiGangi et al. methodology Di Gangi et al. 2018 (light grey) and LeNup Zhang et al. 2018 methodology (grey) on a fragment of chromosome 16.

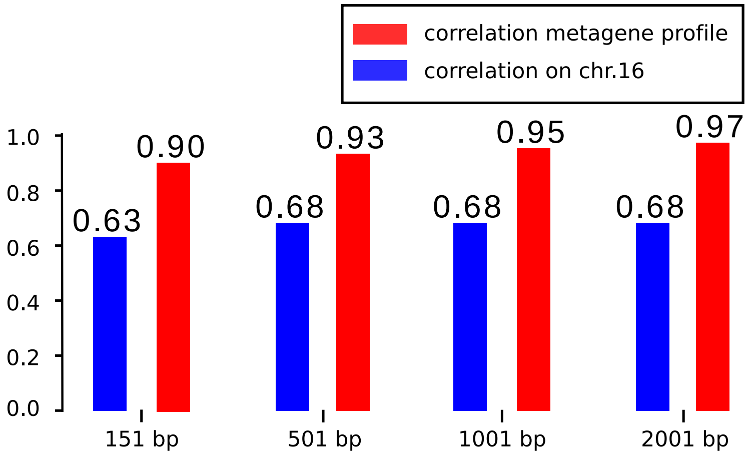

Figure S4. Quantitative evaluation of the input size effect. Correlation between the predicted and the experimental nucleosome density on the whole chromosome 16 (blue) and correlation between the average predicted and experimental density averaged on TSS regions (red). The predicted nucleosome densities are obtained with CNN models trained with different input length : 151bp, 501bp, 1001bp and 2001bp. All the other hyperparameters of the networks are the same.

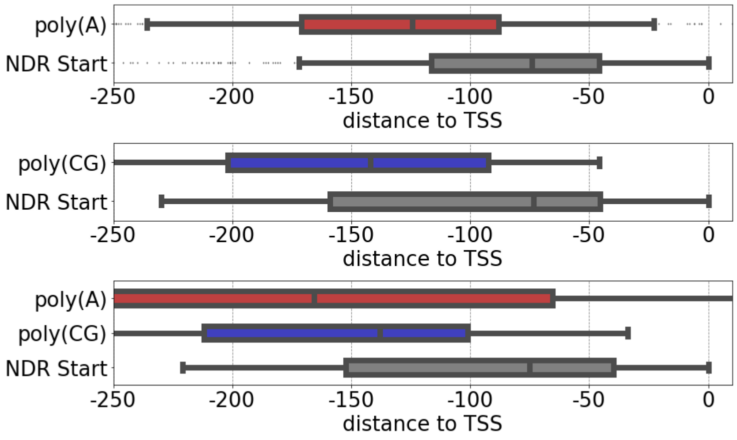

Figure S5. Position of detected sequence logos in NDR. Distribution of the position of poly(A), poly(CG) and NDR start with respect to the TSS for A-T NDR, CG NDR and A-T-CG NDR. Only poly(A) and poly(CG) are displayed since the conjugate motif poly(T) have the same distribution relative to the TSS as poly(A).

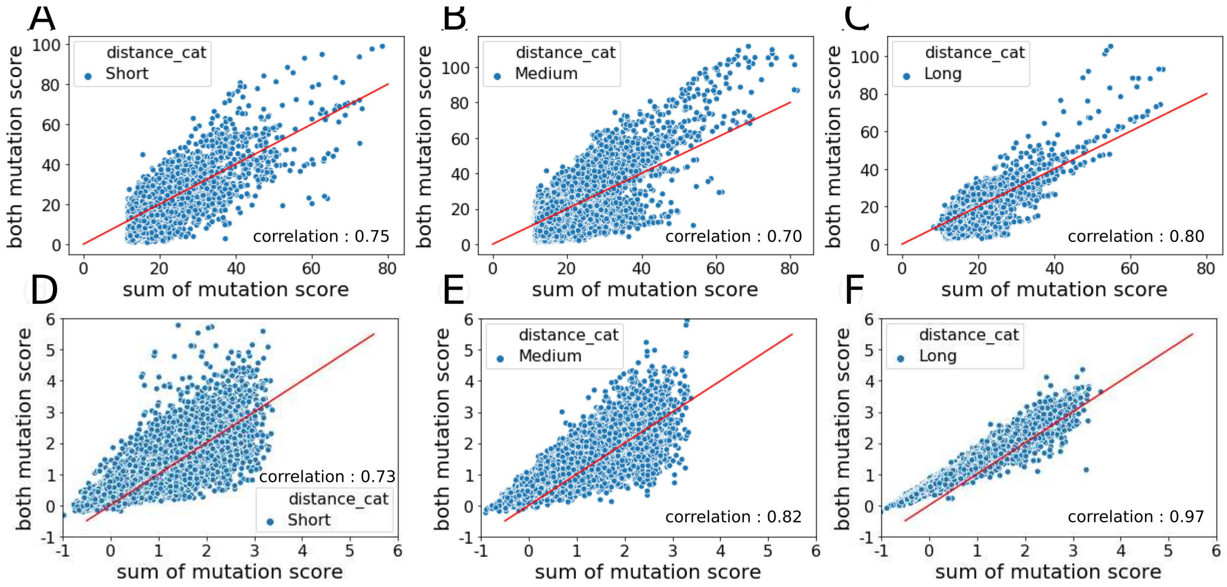

Figure S6. Comparison between the sum of two mutation scores and the mutation score obtained by mutating the nucleotides simultaneously. red(a) Scatter plot representing the mutation score obtained by mutating two nucleotides of high mutation score (> 5) in function of the sum of the mutation score associated to those nucleotides. The distance between the two nucleotides is less than 5 bp. (b) Same as (a) but the distance between the two nucleotides is comprised between 5 bp and 90 bp. (c) Same as (a) but for the distance between the two nucleotides is comprised between 90 bp and 500 bp. (d-e-f) Similar to (a-b-c) when the two nucleotides have a low mutation score (< 1).

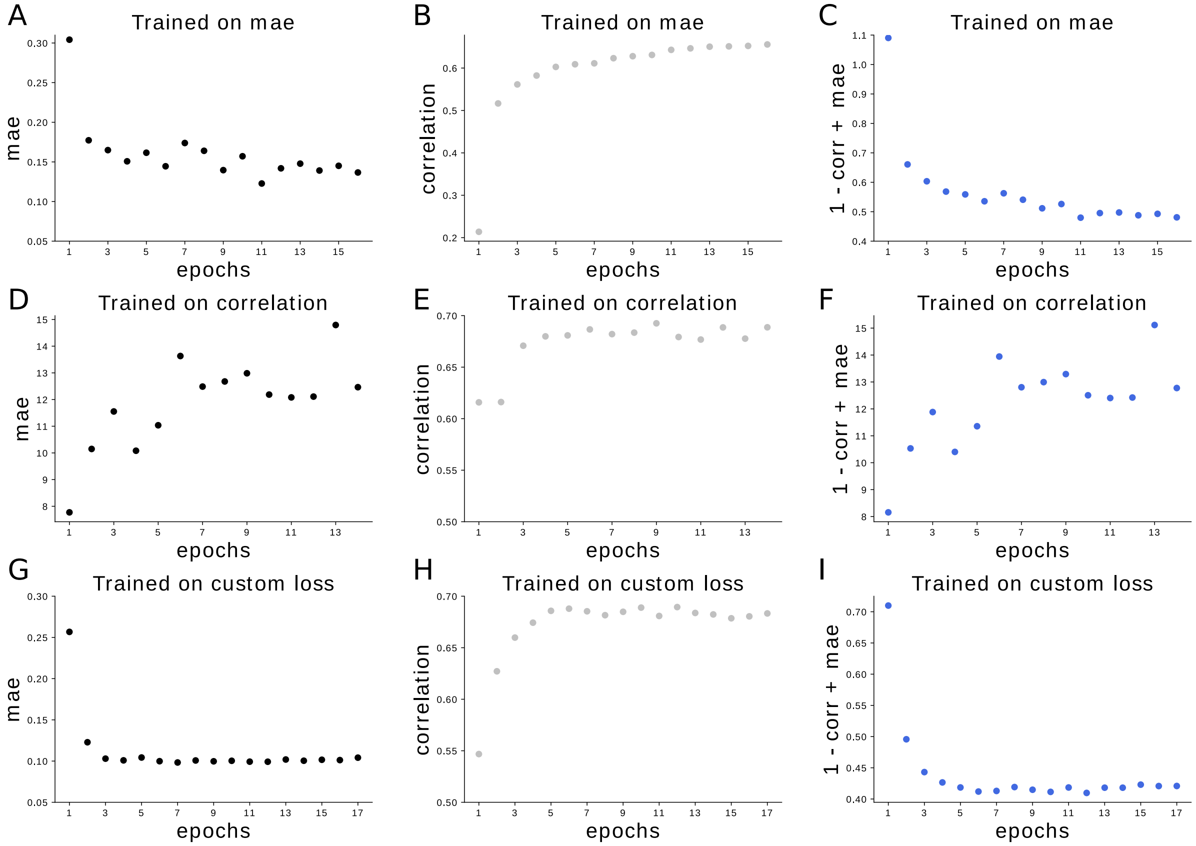

Figure S7. Evolution of the loss function, the correlation and the mean absolute error during training. (a-b-c) The model is trained using the mae as loss function. (a) Evolution of the mae during training, the minimal mae obtained is 0.11. (b) Evolution of the correlation during training, the maximum obtained is 0.64. (c) Evolution of our custom loss function during training, the minimum is 0.48. (d-e-f) The model is trained using the correlation as loss function. (d) Evolution of the mae during training, the mae is diverging. (e) Evolution of the correlation during training, the maximum obtained is 0.69. (f) Evolution of our custom loss function during training, diverging. (g-h-i) The model is trained using our custom loss function. (d) Evolution of the mae during training, the minimal mae obtained is 0.1. (e) Evolution of the correlation during training, the maximum obtained is 0.69. (f) Evolution of our custom loss function during training, the minimum obtained is 0.41.

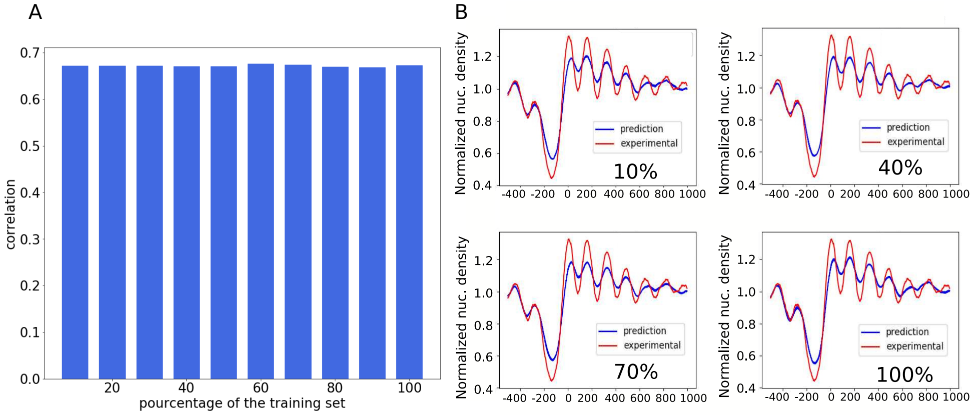

Figure S8. Influence of the training set size. (a) Correlation between the prediction and the experimental density on chromosome 16 for several model trained with different percentage of the training set. (b) Average nucleosome density in TSS region for model trained with chosen percentage of the training set.

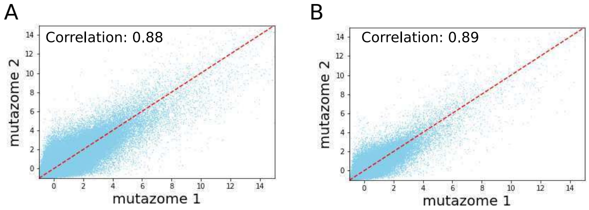

Figure S9. Evaluation of the reproducibility of the mutation score on chromosome 16. (a) Scatter plot representing two mutation scores on chromosome 16. Six models were trained independently on all the genome except chromosome 16, each score was obtained by averaging the scores obtained from three models.