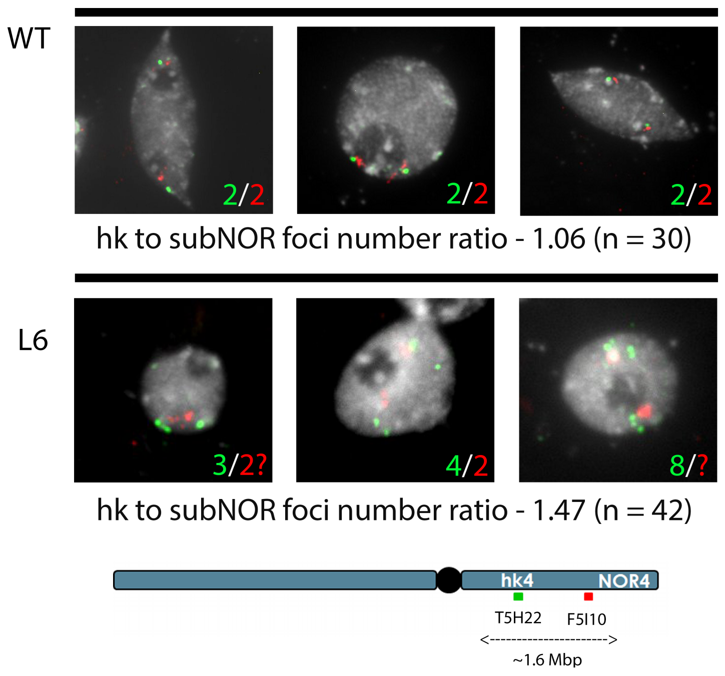
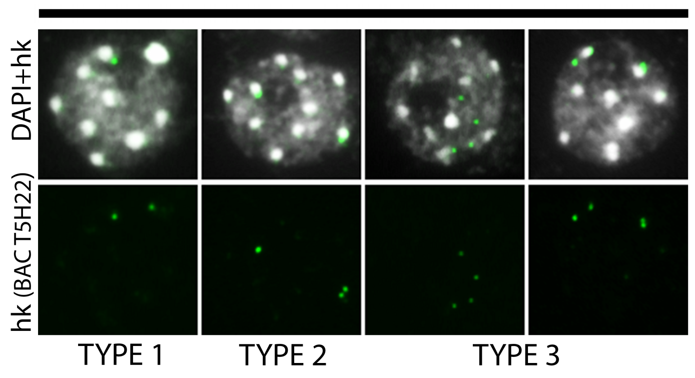
Supplemental Material document

A **LEAF NUCLEI**


WT
2/2
2/2
2/2
hk to subNOR foci number ratio - 1.06 (n = 30)
L6
3/2?
4/2
8/?
hk to subNOR foci number ratio - 1.47 (n = 42)
hk4
NOR4
T5H22
F5I10
<---------------------->
~1.6 Mbp

B **FLORAL BUD NUCLEI**

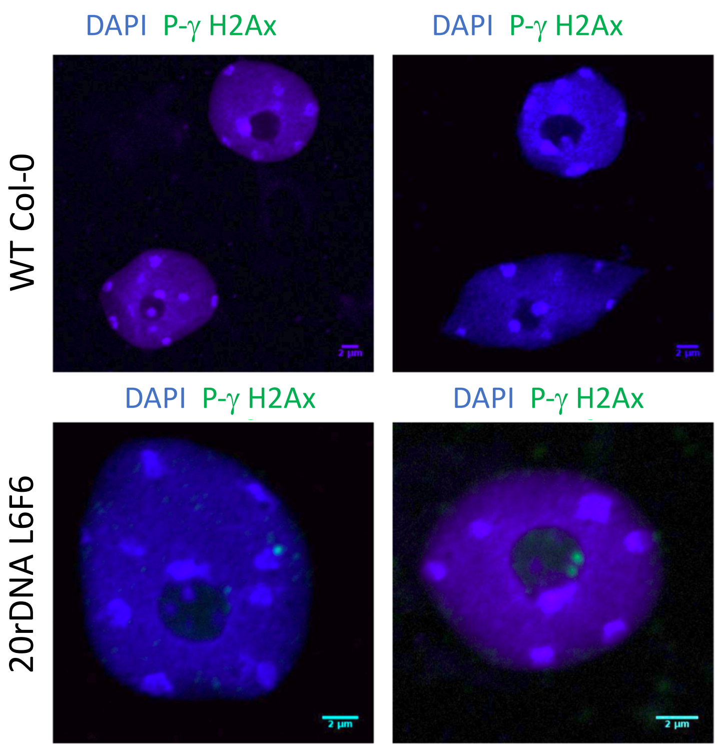


	TYPE 1	TYPE 2	TYPE 3	total
WT	77	0	0	77
L6	8	24	34	66
L9	56	2	0	58

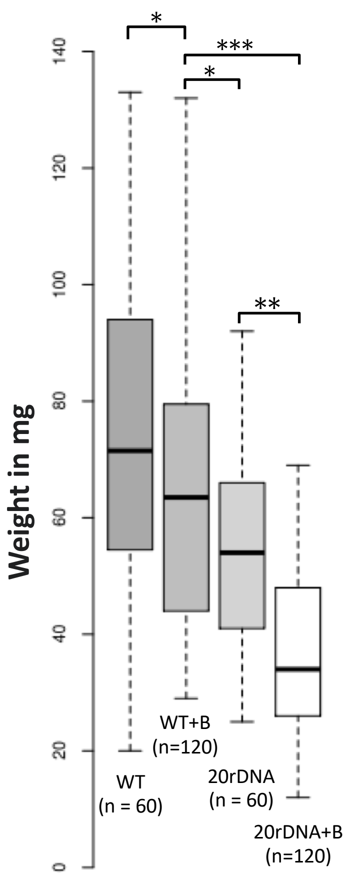
* in WT also some foci appear large, as if there were double dots, but not separated as in L6

Supplemental Fig S1: DNA-FISH analyses of TDDO4 in leaf and floral bud nuclei.
A-B. DNA-FISH analyses of a portion of the TDDO4 in leaf nuclei. **A.** Two *loci* present on the short arm of Chromosome 4, distant by 1.6 Mb are labeled. One is present on the TDDO4 (BAC T5H22 – green) and one is outside of the TDDO4 (BAC F5I10 – red). Three leaf nuclei from WT Col-0 and 20rDNA L6F6 are shown and the ratio of green versus red signals as well as its ratio noted in the figure. **B.** In that case, the TDDO4 locus (BAC T5H22 – green) was analyzed in floral bud nuclei and four representatives images are presented in the figure. DNA is labelled with DAPI stain (Top panel, with DAPI in grey). Between 58 and 77 nuclei per genotype (wild-type Col-0, rDNA20 L6 and L9) were analysed and the distribution of each nuclei found per genotype is presented in the table.

A



B

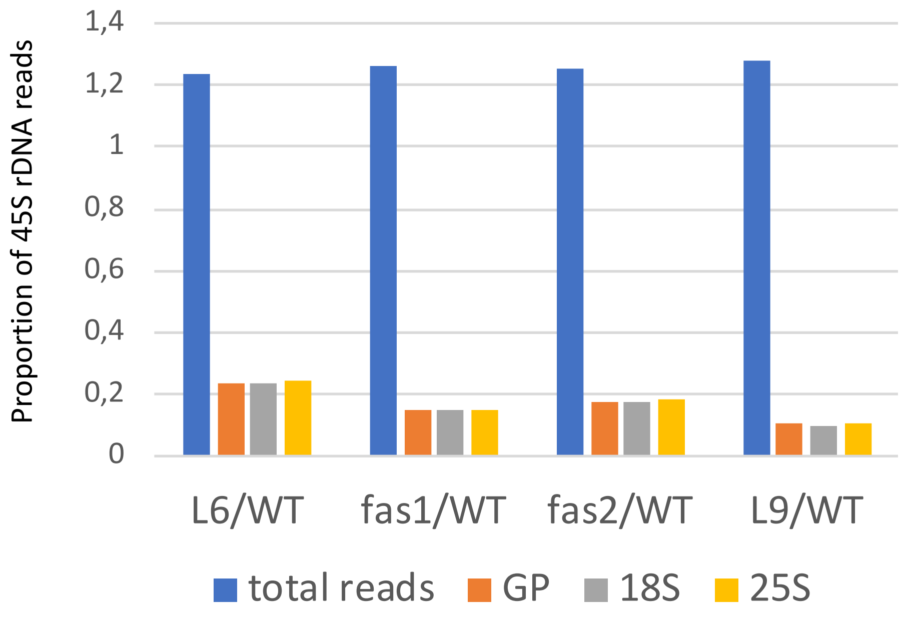


Paired Student's t test
(two sided, non equal variances)

* = p value < 0.001
** = p value < 0.00001 (e-5)
*** = p value < 0.0000001 (e-7)

Supplemental Fig S2: Signs of genomic instability in 20rDNA L6F6

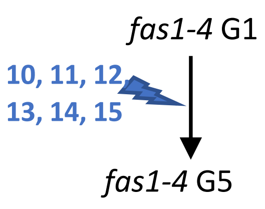
A. Representative confocal images showing the immunolocalization of the P-γH2Ax foci (green) in wild-type Col-0 (top panels) or in 20rDNA F6L6 (bottom panels) isolated nuclei. The Blue staining correspond to the DNA labeled with DAPI. **B.** Wild-type Col-0 and 20rDNA L6F6 seeds were grown for 18 days on an MS plate with or without Bleomycine at 10^{-5}M. Individual plant growth was then measured using their fresh weights.



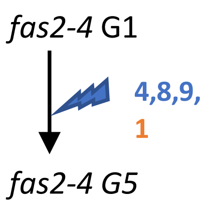
Supplemental Fig S3: Proportion of 45S rDNA in 20rDNA L6F6, L9F6 and in the **fas1-4** ***and*** **fas2-4** ***mutants***

Histogram displaying the relative proportion of long-reads obtained using nanopore sequencing that mapped again 45S rRNA genes.

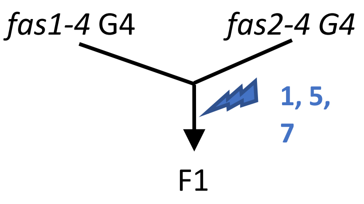
A



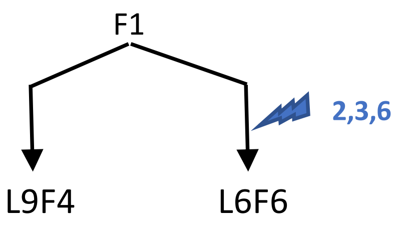
B



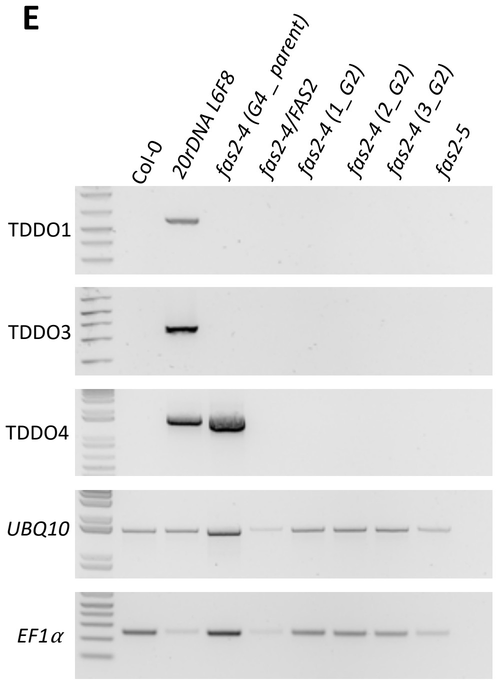
C



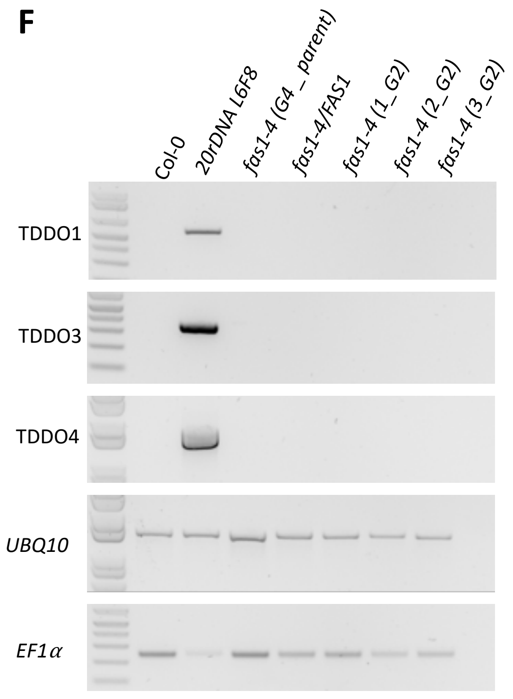
D



E

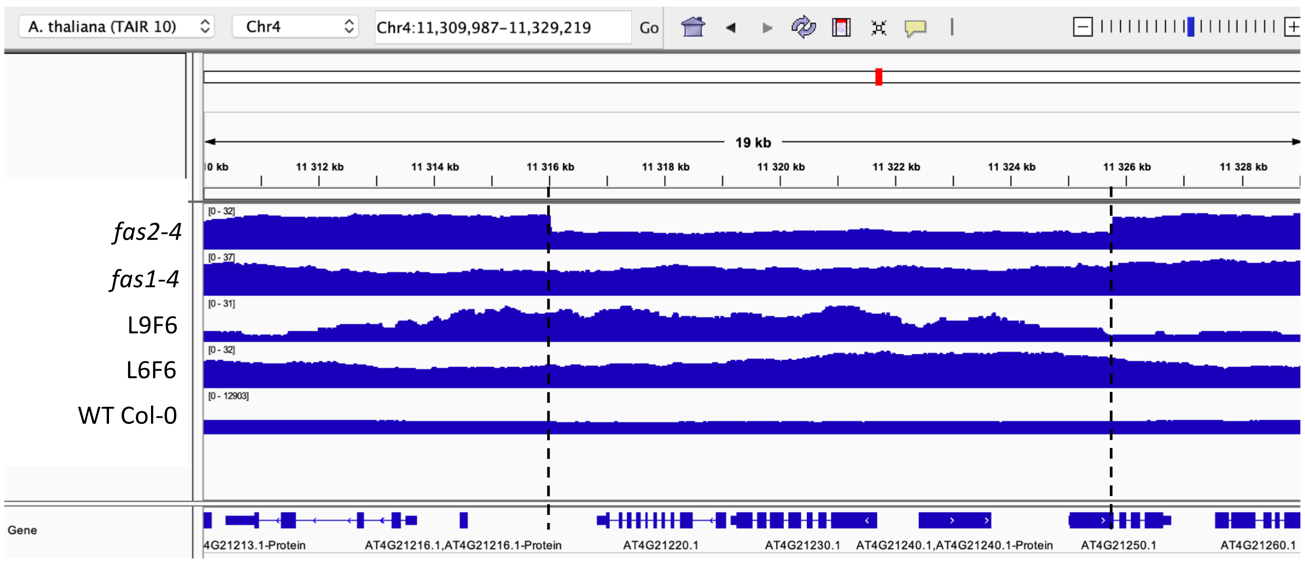


F



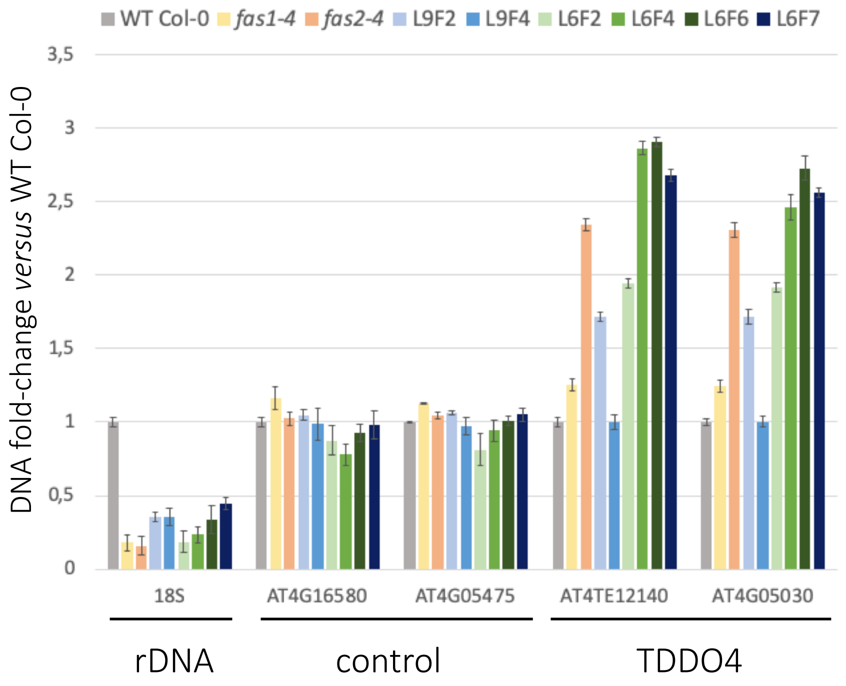
Supplemental Fig S4: Appearance of TDDOs in few generations

A-D Potential *scenarios* of TDDO apparition in *fas1-4, fas2-4*, in the F1 and from F1 to L6F6. No L9F6 specific TDDO were detected due to a lower sequence coverage obtained during the nanopore sequencing. **E-F.** PCR performed with primers flanking the breaking junctions of the TDDO1, TDDO3 and TDDO4 in the wild-type Col-0 and in a panel of *fas1* (**E**) or *fas2* (**F**) knock-out mutants. The loci encoding the elongation factor *EF1α* and *UBIQUITIN 10* (*UBQ10)* genes were used as a loading control.

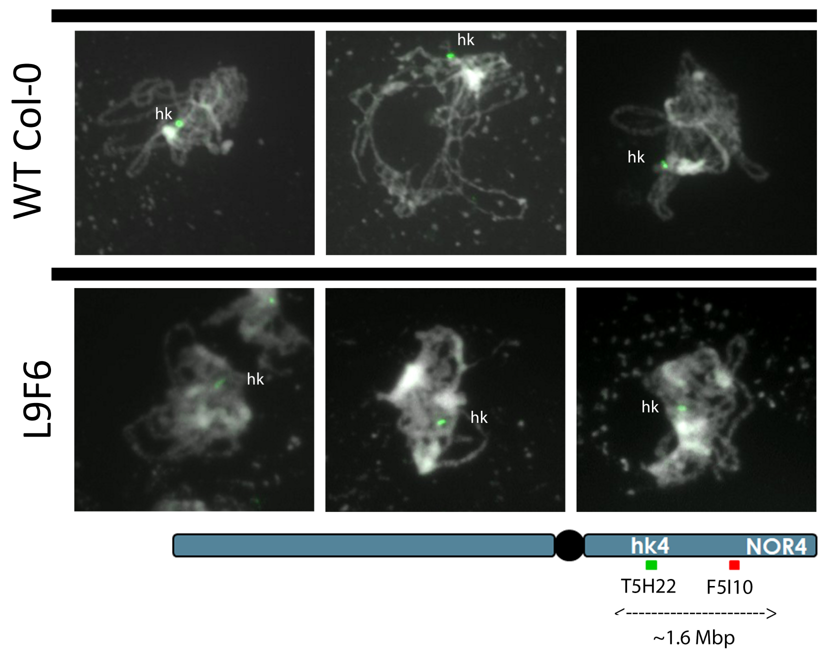


Supplemental Fig S5: Visualization of DEL1 in **fas2-4**
Screenshot obtained from the Integrative Genomic Viewer (IGV) software (Thorvaldsdottir et al. 2013) at the position of the deletion detected in *fas2-4* (G5). Read coverage from each genotype are displayed in blue. Deletion is located on Chromosome 4, from position 11,316,008 to 11,325,758 bp and is shown by the black dotted line.

A



B



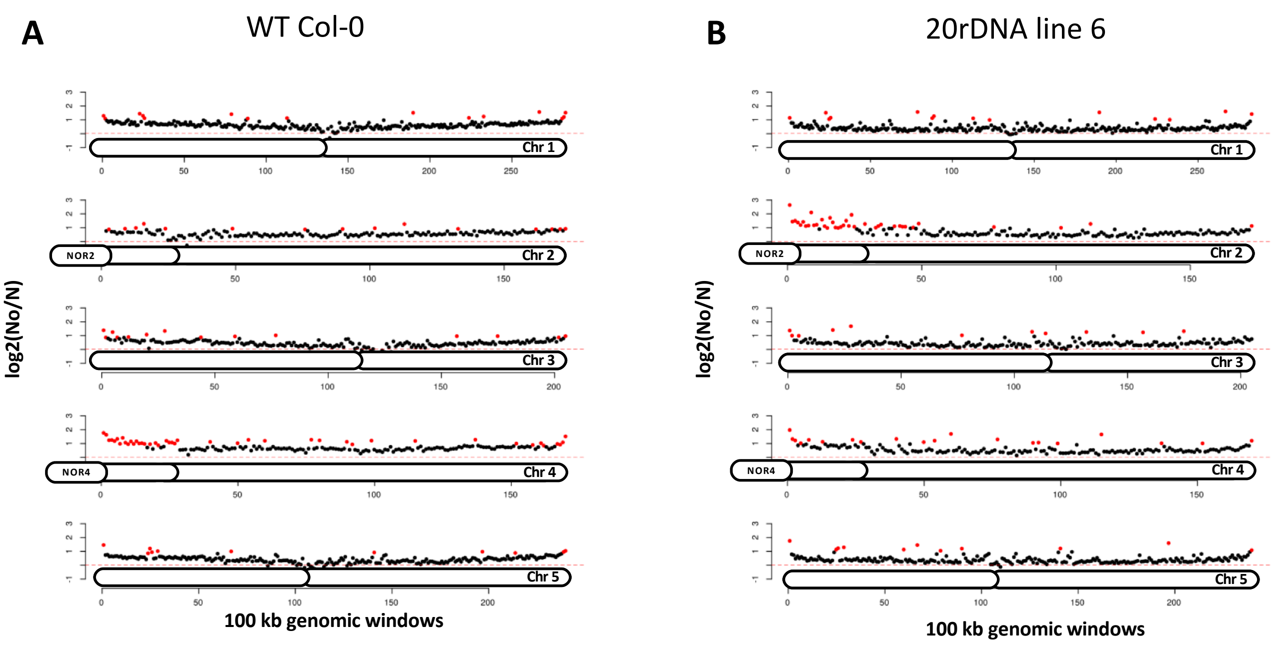
Supplemental Fig S6: Specific analyses of TDDO4

A. CNV of genes present or not in TDDO4 and of rRNA genes were determine by quantitative PCR. Their relative enrichment was determined in WT Col-0, the parent lines *fas1-4* and *fas2-4* and in L6 and L9 at several generations. RRNA genes CNVs was followed via probes amplifying the 18S, the loci AT4G05475 and AT4G16580 that are not duplicated are controls, while the *loci* AT4TE12140 and AT4G05030 allow the identification of the largest duplication TDDO4 **B**. DNA-FISH analyses of 2 loci present on the short arm of Chromosome 4 (kr4s), distant by 1.6 Mb: one present on the TDDO4 (BAC T5H22 – green) and one located outside the TDDO4 (BAC F5I10 – red). Three Pachytene chromosomes from WT Col-0 and in 20rDNA L9F6 are shown.

BORDER	Border description	position on chr	GENE name	Distribution in lines	TDDO or DEL size (bp)
TDDO1 LB	AT1G54770	Chr1:20,435,033	FCF2 pre-rRNA processing protein	L6, L9	202755
TDDO1 RB	AT1G55325	Chr1:20,637,788	Mediator 13 _ GCT	L6, L9	202755
TDDO2 LB	IGS	Chr2:6,345,231		L6	77901
TDDO2 RB	AT2G14930	Chr2:6,423,132	TE	L6	77901
TDDO3 LB	AT2G37520	Chr2:15,749,274	Acyl-CoA N-acyltransferase with RING/FYVE/PHD-type zinc	L6	355525
TDDO3 RB	AT2G38460	Chr2:16,104,799	IREG1	L6	355525
TDDO4 LB	AT4G02960	Chr4:1,313,756	AT4TE06710 (Copia)	L6, fas2	1453051
TDDO4 RB	AT4G05475	Chr4:2,766,807	RNI1-like	L6, fas2	1453051
TDDO5 LB	AT4G04950	Chr4:2,517,674	GRXS_17	L6, L9	96753
TDDO5 RB	IGS	Chr4:2,614,427		L6, L9	96753
TDDO6 LB	AT4G19860	Chr4:10,779,531	cytosolic calcium-independent phospholipase A.	L6	59638
TDDO6 RB	IGS	Chr4:10,839,169		L6	59638
TDDO7 LB	IGS	Chr5:12,489,606		L6, L9	99730
TDDO7 RB	AT5G33320	Chr5:12,589,336	CUE1	L6, L9	99730
TDDO8 LB	AT5G24260	Chr5:8,237,988	prolyl oligopeptidase family protein	fas2	286593
TDDO8 RB	AT5G24820	Chr5:8,524,581	Eukaryotic aspartyl protease family protein	fas2	286593
TDDO9 LB	AT4G25220	Chr4:12,921,881	glycerol-3-phosphate permease 2, G3Pp2	fas2	106828
TDDO9 RB	AT4G25515	Chr4:13,028,709	SEUSS-like 3, SLK3	fas2	106828
TDDO10 LB	IGS	Chr1:3,595,445		fas1	70624
TDDO10 RB	AT1G10970	Chr1:3,666,069	zinc transporter 4 (ZIP4)	fas1	70624
TDDO11 LB	AT2G03990	Chr2:1,266,696	TE	fas1	77270
TDDO11 RB	AT2G04060	Chr2:1,343,966	glycosyl hydrolase family 35 protein	fas1	77270
TDDO12 LB	AT4G18030	Chr4:10,014,227	S-adenosyl-L-methionine-dependent methyltransferases	fas1	57849
TDDO12 RB	AT4G18197	Chr4:10,072,076	purine permease 7 (PUP7)	fas1	57849
TDDO13 LB	AT5G33050	Chr5:12,406,977	TE (gypsy element)	fas1	175321
TDDO13 RB	IGS	Chr5:12,582,298		fas1	175321
TDDO14 LB	IGS	Chr5:14,178,149		fas1	64318
TDDO14 RB	AT5G36075	Chr5:14,242,467	TE (MuDR family)	fas1	64318
TDDO15LB	IGS	Chr5:26,169,422		fas1	151977
TDDO15 RB	IGS	Chr5:26,321,399		fas1	151977
DEL1 LB	IGS	Chr4:11,316,008		fas2	9750
DEL1 RB	AT4G21250	Chr4:11,325,758	Sulfite exporter TauE/SafE family protein;	fas2	9750

Supplemental Fig S7: Description of DEL and TDDO border

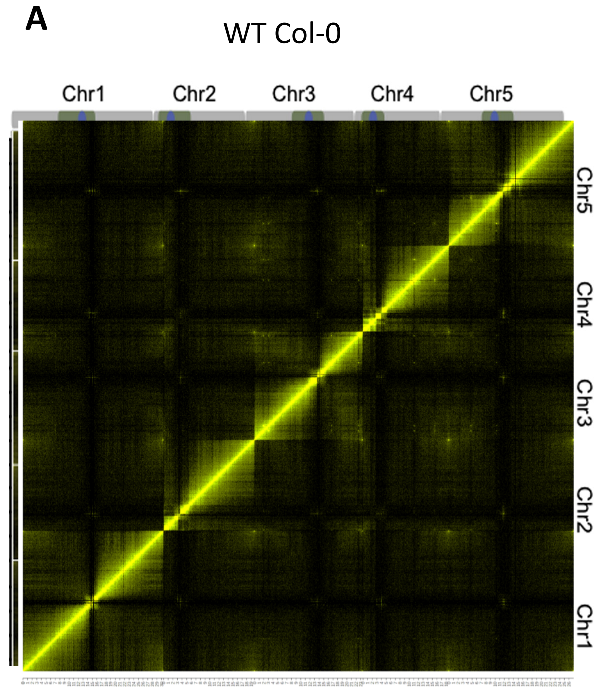
Table describing borders of each TDDO and DEL identified after long-read sequencing. TDDO = Tandem Duplication in Direct Orientation; DEL = Deletion; LB = left border; RB = right border; L6 = 20rDNA L6F6; L9 = 20rDNA L9F6; *fas1* = *fas1-4* (parental line G4 + 1 generation); *fas2* = *fas2-4* (parental line G4 + 1 generation); TE = Transposable Element.



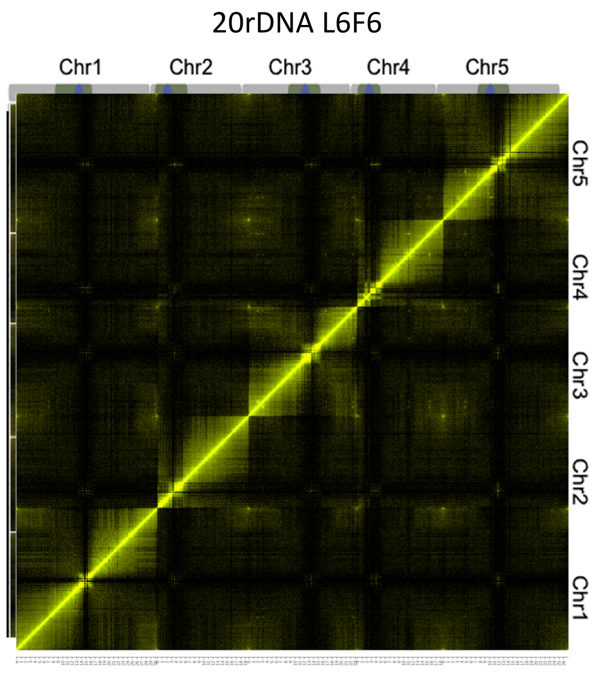
Supplemental Fig S8: NADs identification in wild-type Col-0 and in 20rDNA L6F6

A-B. Chromosome plots displaying the relative enrichment of a given genomic segment with the nucleolus in WT Col-0 (A) and in the 20rDNA L6F6 (B). The y-axis displays the fold change between the nuclear *versus* the nucleolar DNA. Each dot represents a 100kb window. Nucleolus-enriched genomic regions above the threshold (red-dotted line) are colored in red.

A

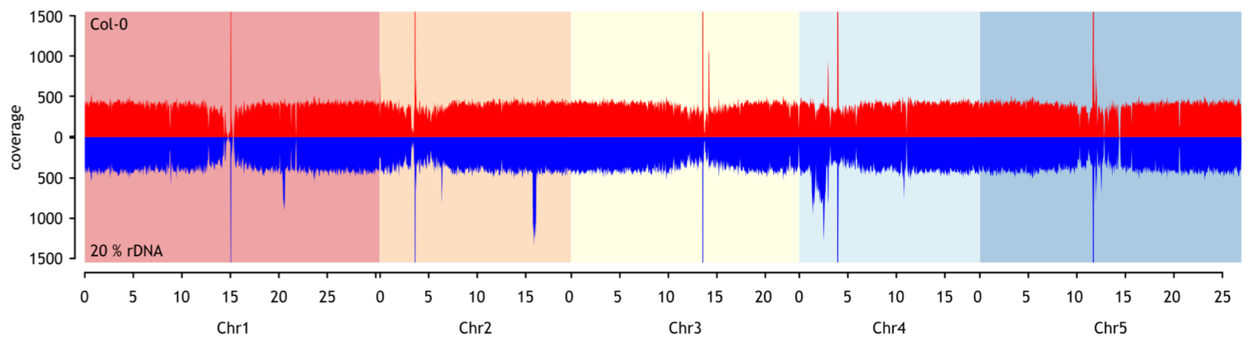


B

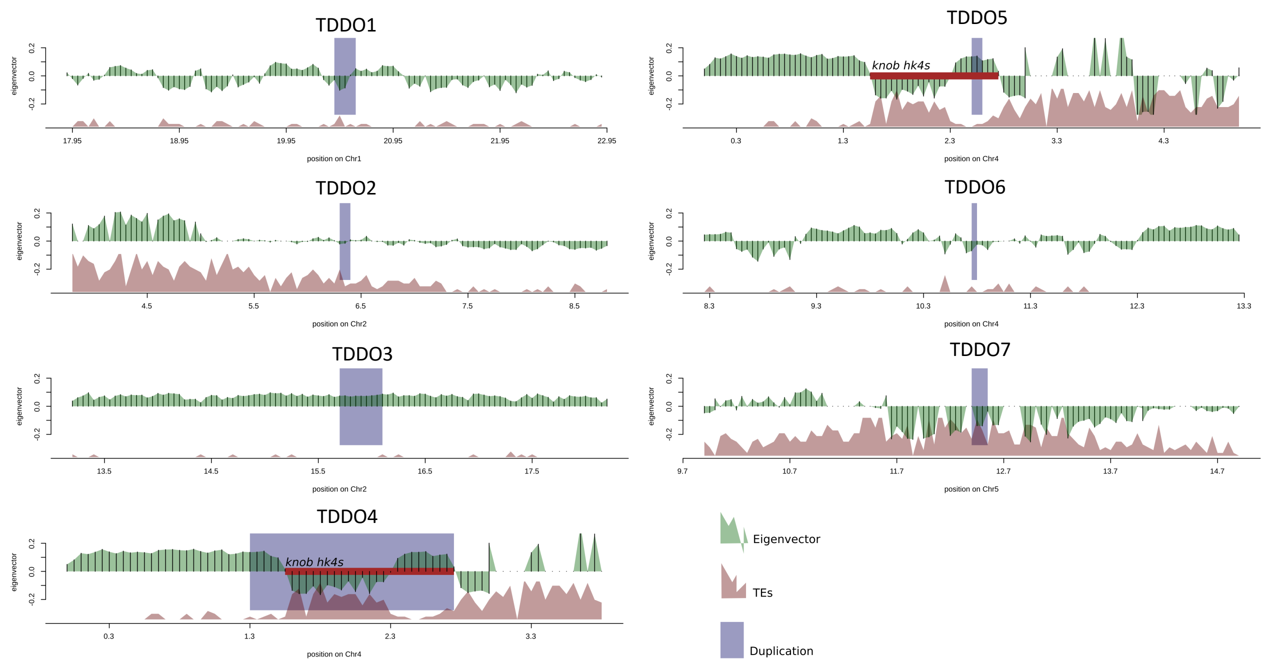


Supplemental Fig S9: Chromatin-chromatin interactions in wild-type Col-0 and in 20rDNA L6F6

A-B. HiC matrix displaying the chromatin-chromatin interactions identified by the chromosome capture conformation technique in wild-type Col-0 (A) and the 20rDNA L6F6 (B). Enriched interactions are displayed in yellow.

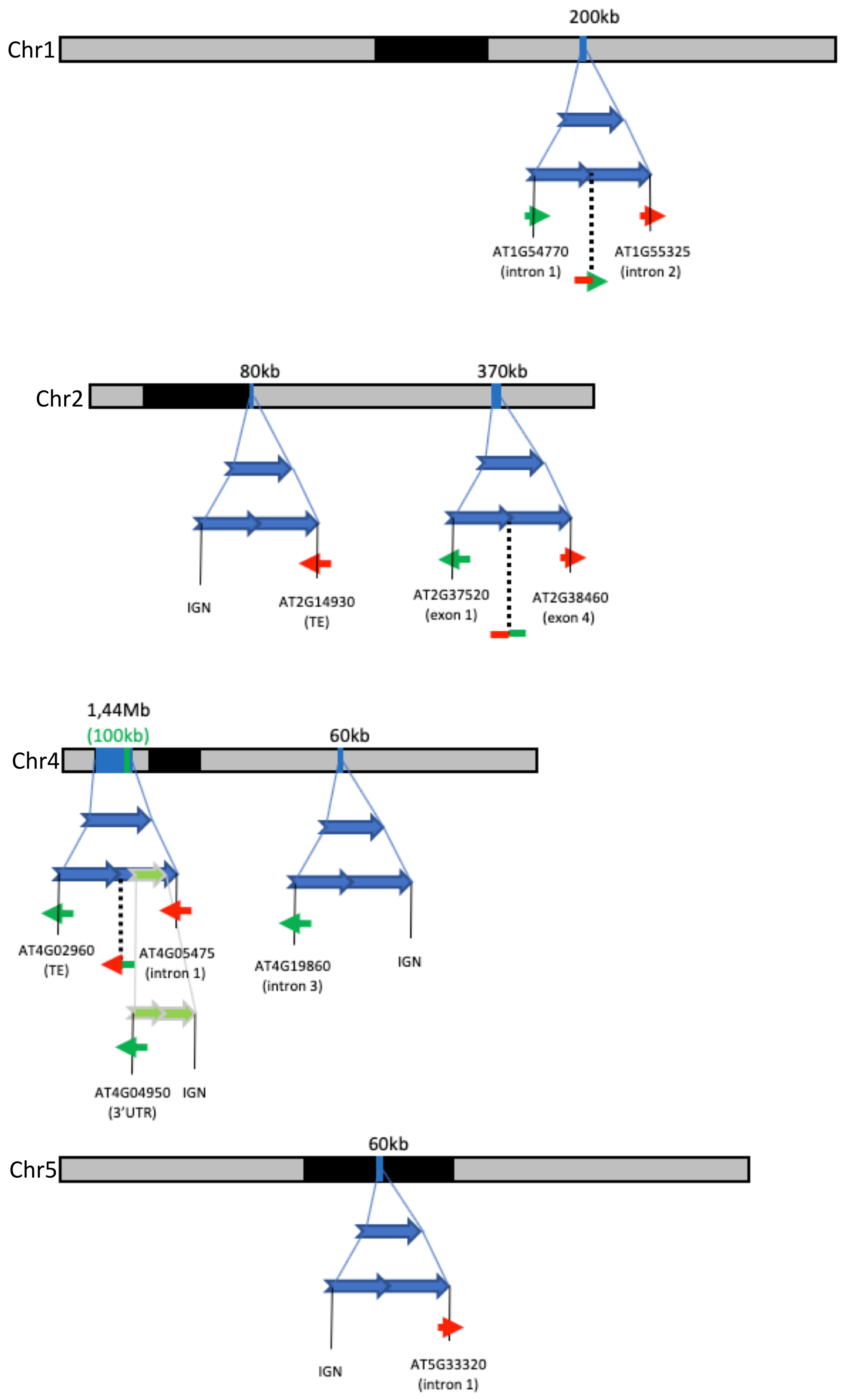


Supplemental Fig S10: Visualization of highly covered regions in L6F6 using Hi-C data
Summed coverage over triplicate Hi-C data sets in 50 kb bins in wild-type Col-0 (top - red) and in 20rDNA L6F6 (bottom – blue). The coverage was calculated based on unpaired single end raw Hi-C reads obtained by Illumina sequencing. The Y axis displays the position on chromosomes.

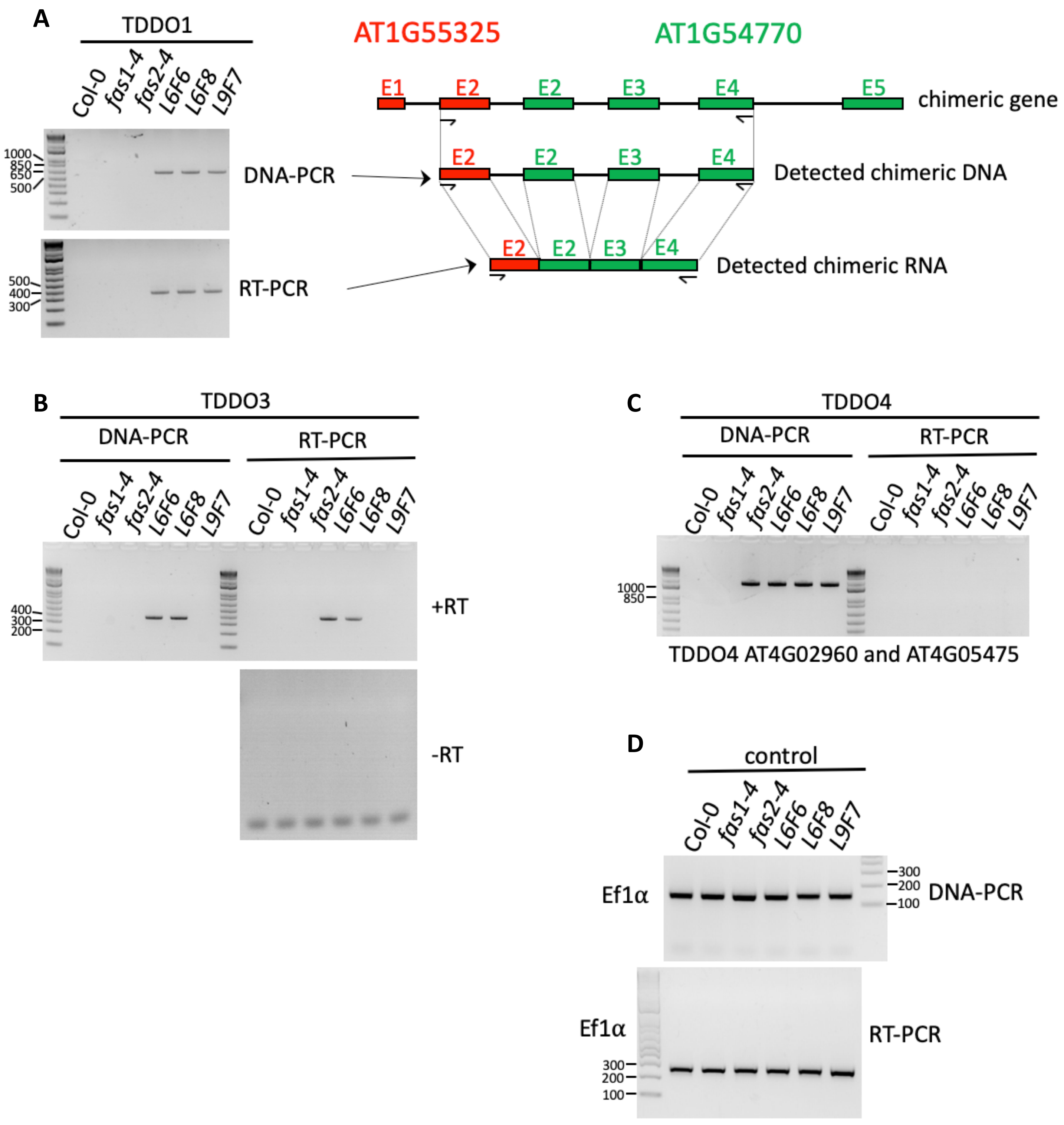


Supplemental Fig S11: Correlation between TDDO breakage points with the Eigenvector in L6F6

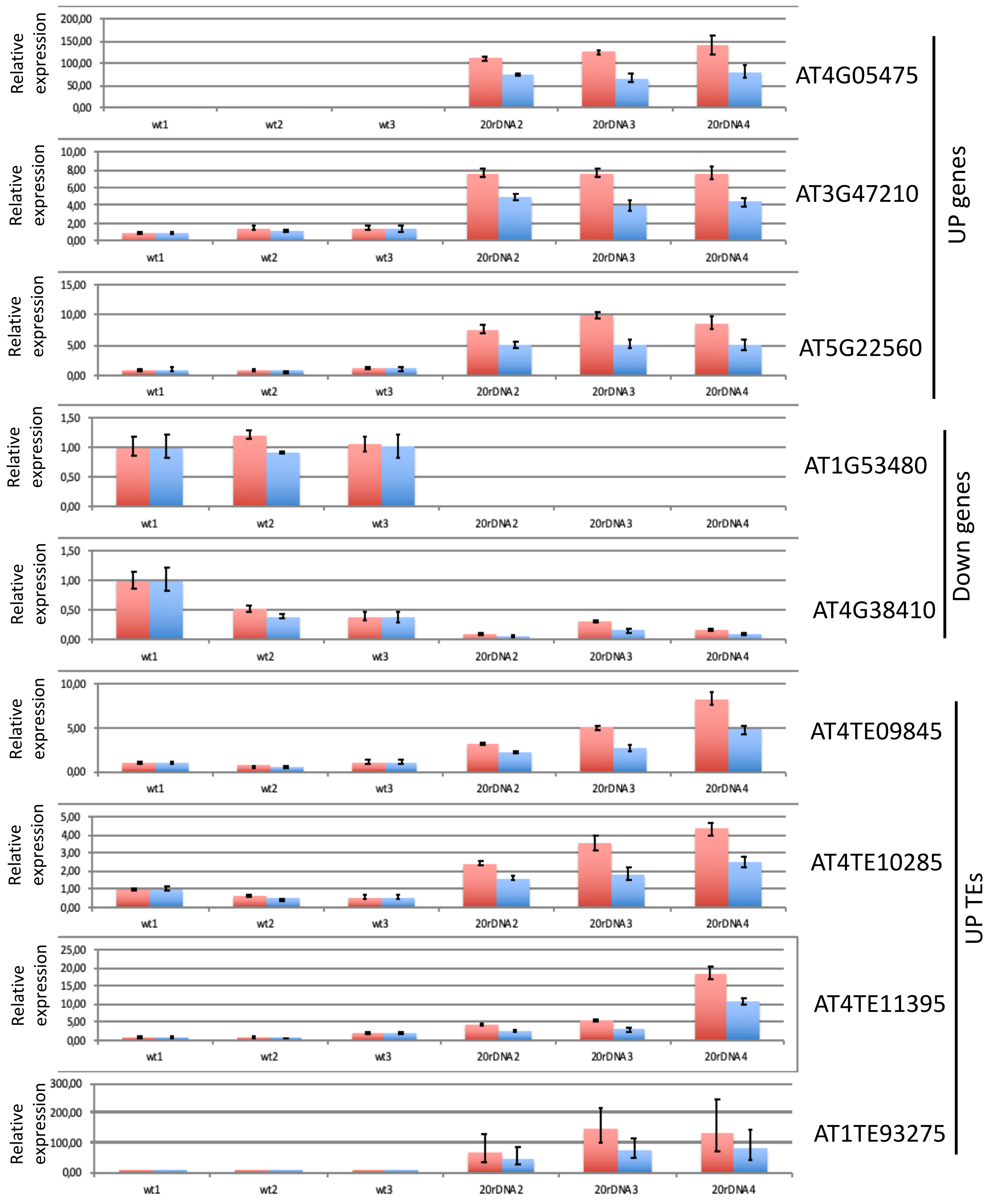
Eigenvector (in green) of the Col-0 wild-type Hi-C data set and annotation of the TDDO. TEs density is marked in red and the violet square show the TDDOs location.



Supplemental Fig S12: Rearrangements at TDDO breakpoint junctions
Distribution of the TDDOs along the *A. thaliana* chromosomes and analyses of the breaking junctions. Genes present in the breaking junction and their orientation are represented by an arrow in green (for the 5' of the TDDO) and in red (for the 3' of the TDDO).

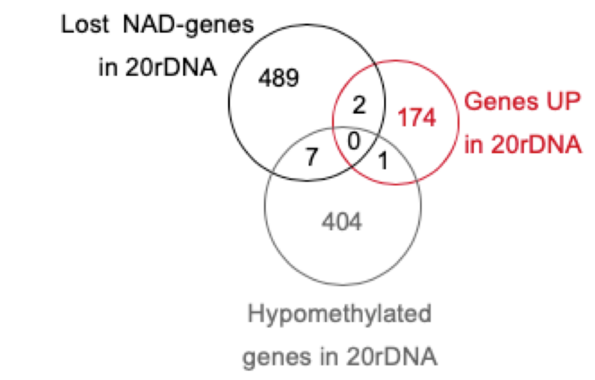


Supplemental Fig S13: Presence and expression of chimeric genes in* fas *mutants, L6 and L9 lines.
A-D. PCR performed with primers flanking the breaking junctions of the TDDO1 (**A**), TDDO3 (**B**) and TDDO4 (**C**) in the wild-type Col-0, the two mutants *fas1-4* and *fas2-4*, the 20rDNA lines L6 (generations F6 and F8) and L9 (generation F7). For the TDDO1, the cDNA amplified for shorter using cDNA versus gDNA template due to the correct splicing of the three introns, as confirmed by Sanger sequencing. The locus encoding the elongation factor *EF1α* was used as a loading control in (**D**).

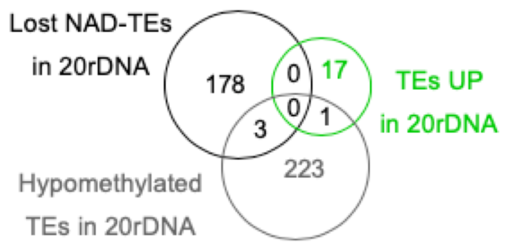


Supplemental Fig S14: Validation of gene expression changes by quantitative RT-PCR
Quantitative PCR analyses of differentially accumulated genes from three independent pool of wild-type Col-0 or 20rDNA L6F6. Relative accumulation each cDNA was analyzed against the *ACTIN7* (red bars) or the *HC7* (blue bars) household genes.

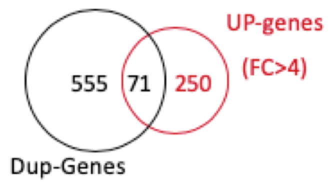
A



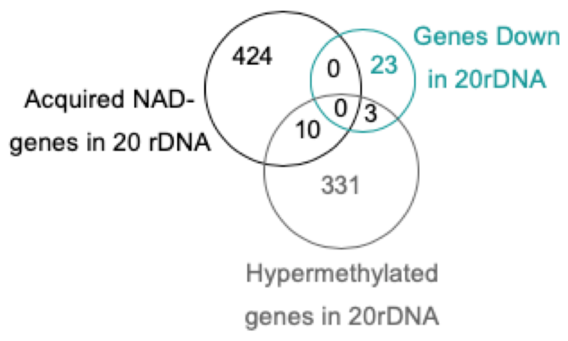
C



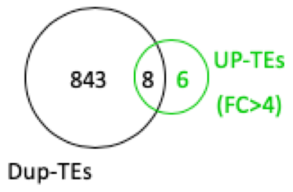
D



B

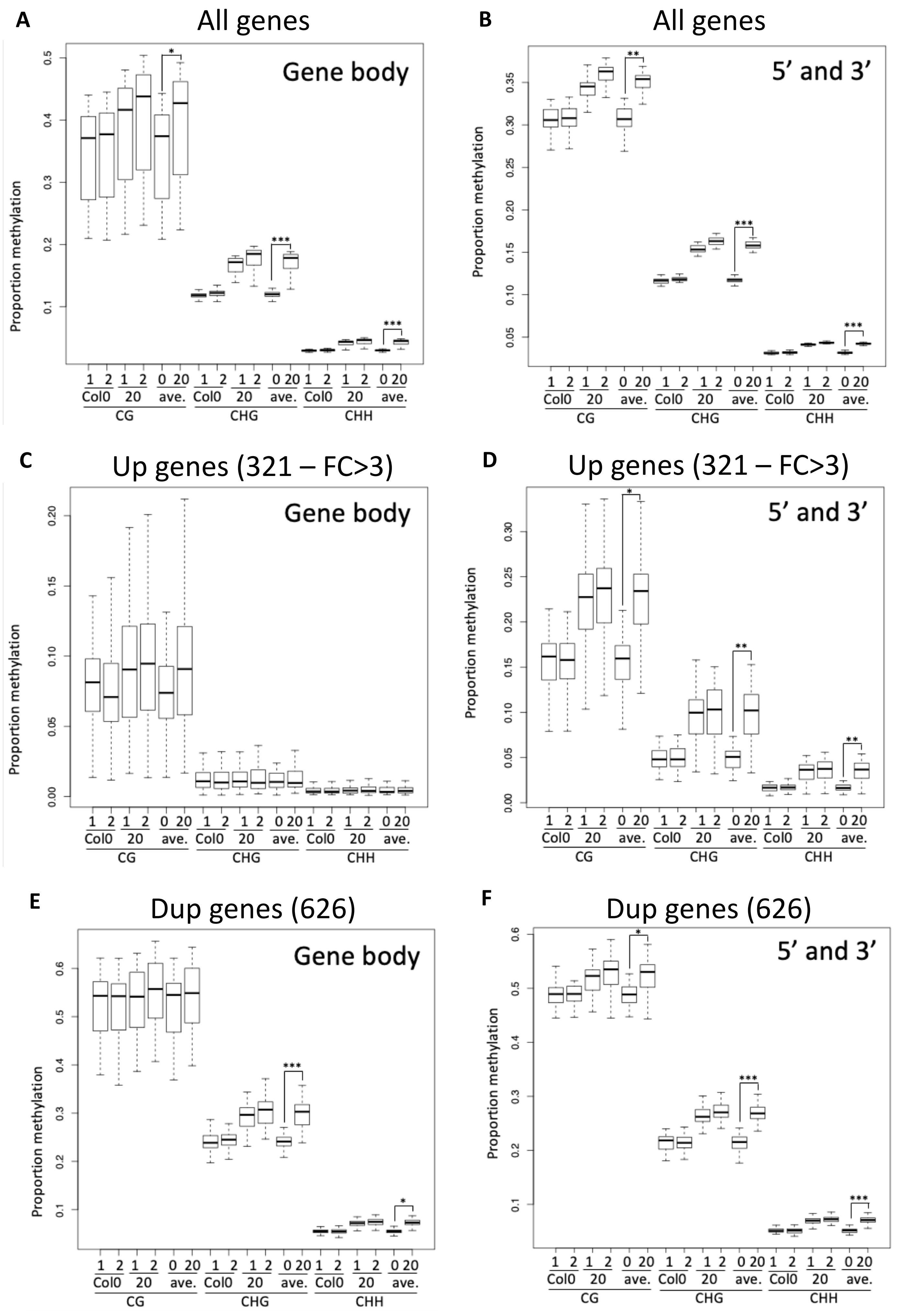


E



Supplemental Fig S15: Overlap between, nucleolar association, gene expression, DNA methylation and gene duplication

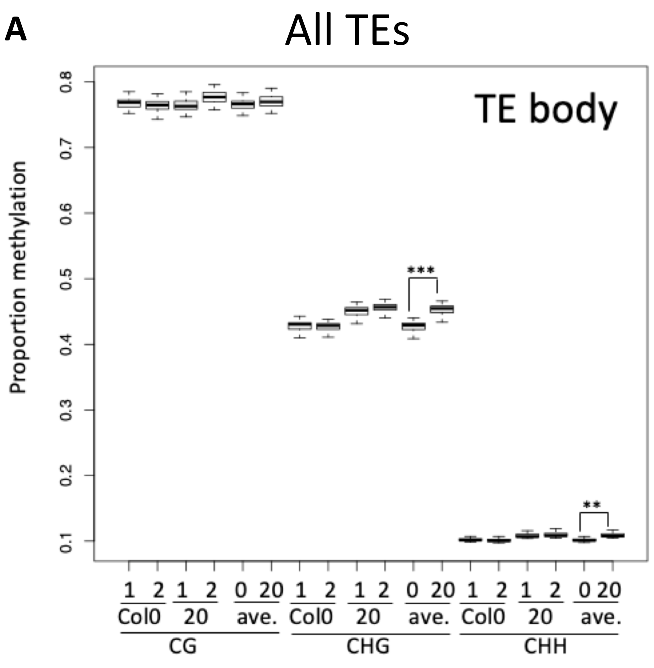
A. Venn diagram showing the relation between genes that lost their association with the nucleolus in the 20rDNA L6F6 (lost NAD-genes in 20rDNA), Genes that are upregulated in 20rDNA L6F6 with an adjusted *p*-value < 0.01 and a log2FoldChange > 2 (Genes UP in 20rDNA) and with hypomethylated genes in 20rDNA L6F6. **B**. Venn diagram showing the relation between genes that gain association with the nucleolus in the 20rDNA L6F6 (Acquired NAD-genes in 20rDNA), Genes that are down-regulated in 20rDNA L6F6 with an adjusted *p*-value < 0.01 and a log2FoldChange > 2 (Genes Down in 20rDNA) and with hypermethylated genes in 20rDNA L6F6. **C**. Venn diagram showing the relation between TEs that lost their association with the nucleolus in the 20rDNA L6F6 (lost NAD-TEs in 20rDNA), TEs that are upregulated in 20rDNA L6F6 with an adjusted *p*-value < 0.01 and a log2FoldChange > 2 (TEs UP in 20rDNA) and with hypomethylated TEs in 20rDNA L6F6. **D-E.** Venn diagrams representing the proportion of duplicated genes (D) or TEs (E) that accumulated a higher level of transcripts, using an adjusted *p* value < 0.01 and a log2FoldChange > 1.5



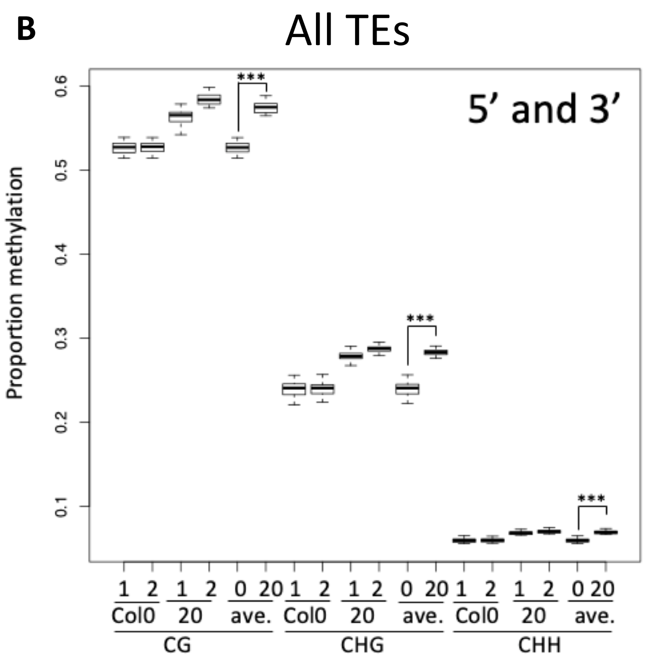
Supplemental Fig S16: Analyses of DNA methylation of genes in 20rDNA L6F6

A-F. Dot plots showing the relative methylation levels in CG, CHG and CHH contexts in wild-type Col-0 or in 20rDNA L6F6 genes. The analysis was performed by genome-wide bisulfite sequencing. Two Wild-type Col-0 (labelled "Col0") and two 20rDNA L6F6 (labelled "20") replicates (1 and 2) are shown. The average methylation between both replicates is shown (labelled "ave."). The analysis was performed with All genes (**A-B**), with up-regulated genes with an adjusted *p*-value < 0.01 and a log2FoldChange > 1.5 (**C-D**) and with the duplicated genes (**E-F**). For this analysis, we separated gene body sequences (**A, C** and **E**) from the flanking sequences of the genes (5' and 3' - **B**, **D** and **F**). *(p<0,00025); **(p<5.10^{-10}); ***(p<2.10^{-16}) calculated using a Wilcoxon test.

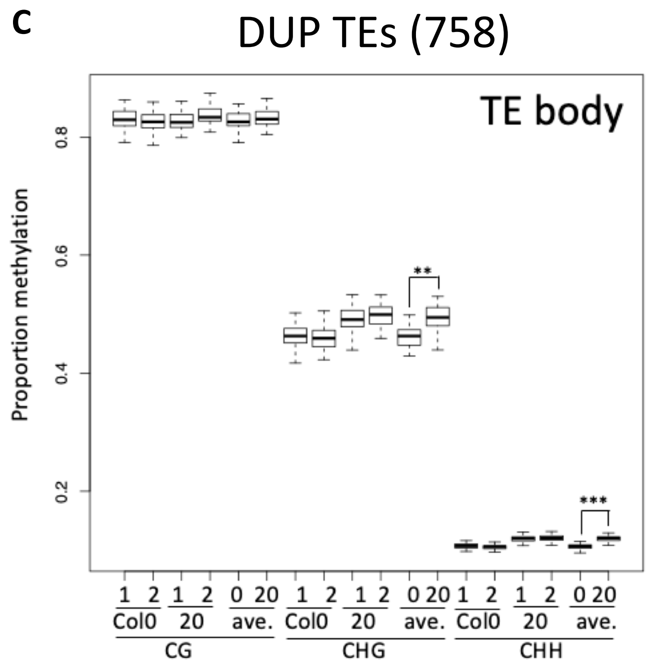
A



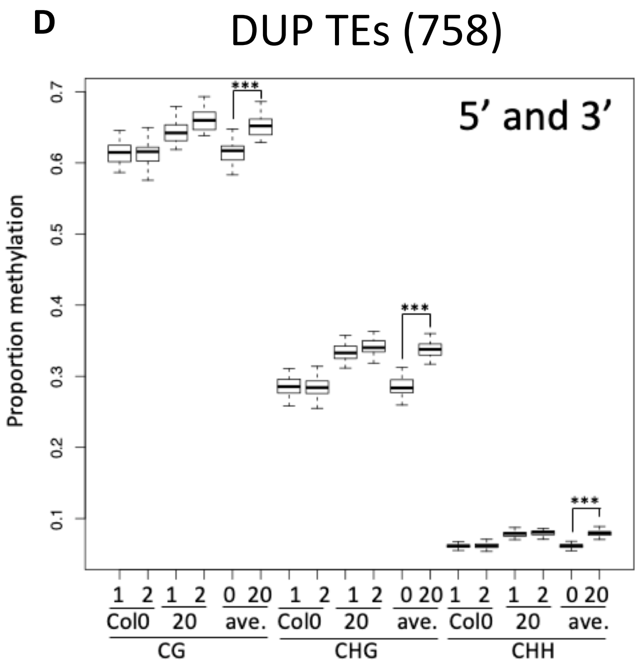
B



C



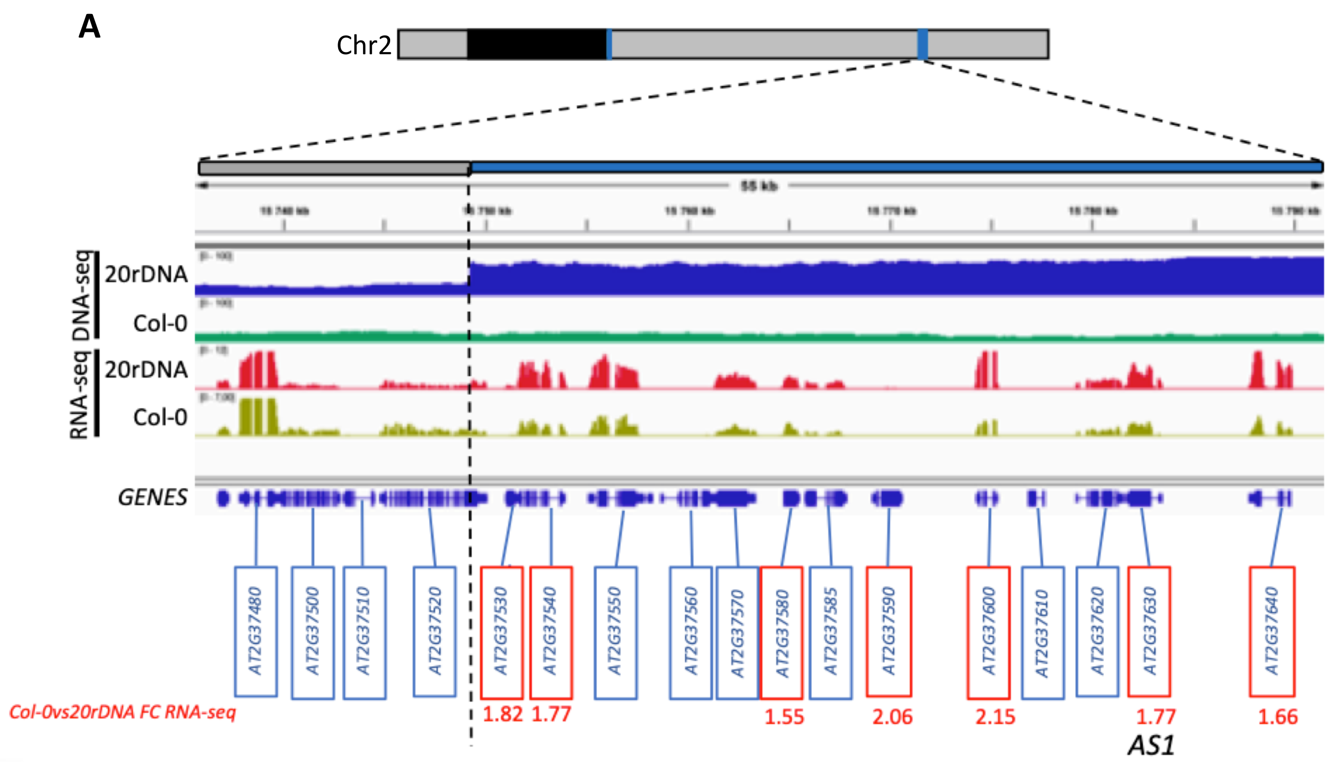
D



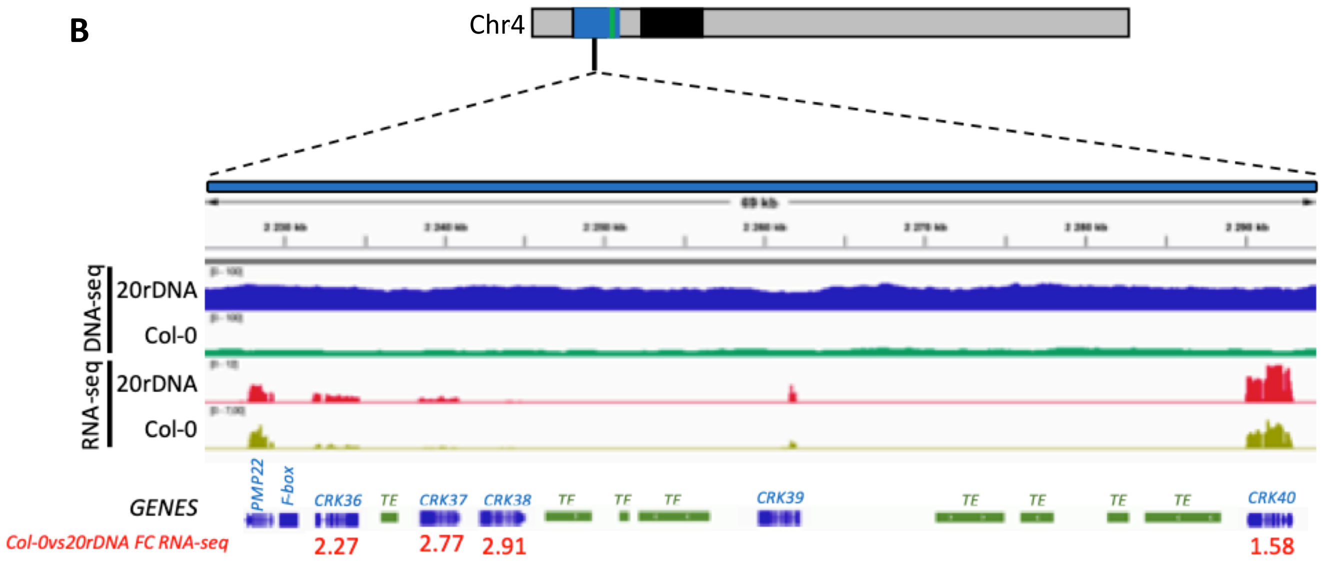
Supplemental Fig S17: Analyses of DNA methylation of TEs in 20rDNA L6F6

A-D. Dot plots showing the relative methylation levels in CG, CHG and CHH contexts in wild-type Col-0 or in 20rDNA L6F6. The analysis was performed by genome-wide bisulfite sequencing. Two Wild-type Col-0 (labelled “Col0”) and two 20rDNA L6F6 (labelled “20”) replicates (1 and 2) are shown. The average methylation between both replicates is shown (labelled “ave.”). The analysis was performed with All TEs (**A-B**) and with the duplicated TEs (**C-D**). For this analysis, we separated TE body sequences (**A** and **C**) from the flanking sequences of the TEs (5’ and 3’ – **B** and **D**). *($p<0,00025$); **($p<5.10^{-10}$); ***($p<2.10^{-16}$) calculated using a Wilcoxon test.

A

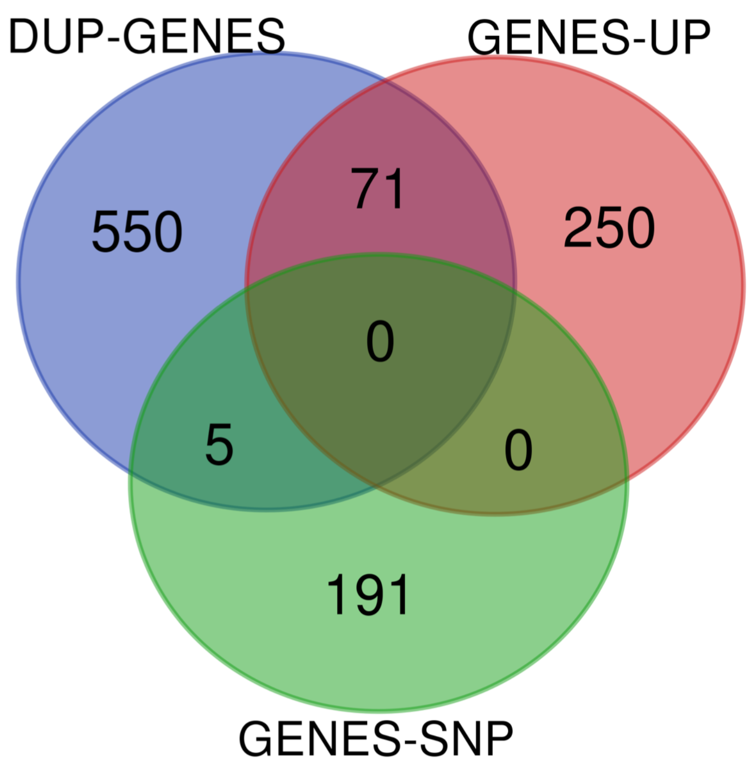


B

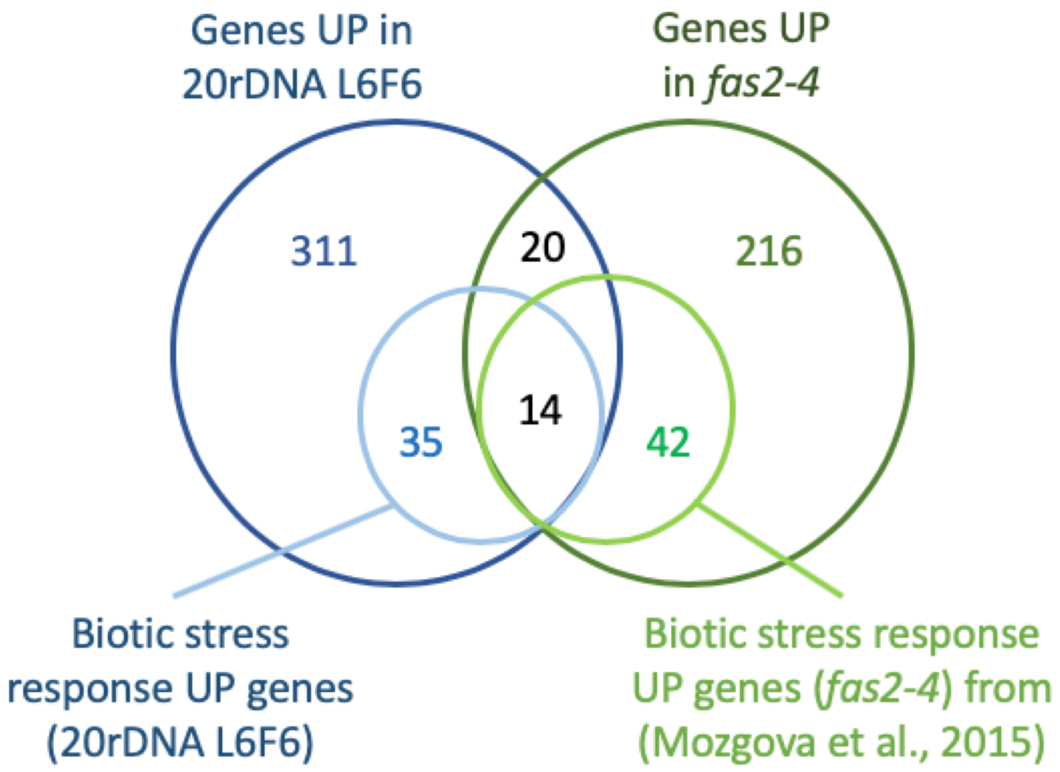


Supplemental Fig S18: Correlation between gene expression and CNV at TDDO3 and TDDO4

A-B. Genome browser (IGV (Thorvaldsdottir et al. 2013)) screenshot showing examples of Up-genes in 20rDNA L6F6 in duplicated regions. DNA coverage was obtained by nanopore sequencing for wild-type Col-0 (green) and for 20rDNA L6F6 (blue). Reads from RNA sequencing are shown for wild-type Col-0 (red) and for 20rDNA L6F6 (yellow). On the bottom, genes location is represented in blue and TEs in green. Gene names are surrounded in blue (non-significant) and in red (significant) when they are differentially accumulated transcripts with an adjusted *p*-value < 0.01 and a log2FoldChange > 1.5. This value is noted in red. Two locations are presented: the left border of the TDDO3 (**A**) and portion of the TDDO4 (**B**).



Supplemental Fig S19: overlap between small nucleotide polymorphism, duplicated and up-regulated genes
Venn diagram showing the relation between duplicated genes (DUP-GENES), up-regulated genes (GENES-UP) and genes with at least one small nucleotide polymorphism (GENES-SNP).


Genes UP in 20rDNA L6F6
Genes UP in *fas2-4*
311
20
216
35
14
42
Biotic stress response UP genes (20rDNA L6F6)
Biotic stress response UP genes (*fas2-4*) from (Mozgova et al., 2015)

Supplemental Fig S20: comparison of upregualed genes from our study and from (Mozgová et al. 2015)
Venn diagram showing the overlap between up regulated genes found in 20 rDNA L6F6 (our data – in blue) and up regulated genes identified in *fas2-4* (data from (Mozgová et al. 2015) – in green). Genes specifically implicated in biotic stress response are featured.

Supplemental Material and Methods

NADs identification

Nuclei and nucleoli were isolated as previously described (Pontvianne et al. 2016) using a S3 cell sorter (Biorad®). Sorted nuclei or nucleoli were treated with RNase A and proteinase K prior to purification and concentration of the DNA using a phenol/chloroforme purification, followed by two precipitation steps. DNA libraries were generated via the kit Nextera XT DNA sample preparation (Illumina®) according to manufacturer's instruction, and were then subjected to high throughput paired-end 2X150nt sequencing on a Next-seq 550 apparatus (Illumina ®) at the Bioenvironment sequencing platform (Perpignan University, Perpignan, France). NADs identification was then performed as described in (Carpentier et al. 2018).

Chromatin-chromatin interactions analyses

Hi-C samples were generated according to (Grob et al. 2014), using *HindIII* as a restriction enzyme. For both wild-type (Columbia-0 accession) and 20rDNA, Hi-C samples were generated in triplicates using 14-day-old seedling populations grown on MS culture media (22°C; 16h light/8h dark). Single ends of paired-end sequencing reads were trimmed to 35 bp using cutadapt (Martin 2011) and aligned separately using bowtie (Langmead et al. 2009) using the following parameters: bowtie -a -v 0 -m 1. After sorting and indexing using samtools (Li et al. 2009), single end alignments were further processed using HiCdat (Schmid et al. 2015). Before mapping read-pairs to genomic bins (10 kb and 50 kb), read pairs were filtered to remove pairs, which paired inwards with less than 1kb and outward with less than 25 kb. The resulting Hi-C matrices were further analyzed in R using HiCdatR package (Schmid et al. 2015) and other in-house R scripts. Hi-C matrices were normalized for sequencing depth (fragment counts per million). To normalize for coverage, we mapped single-end alignments to genomic bins (equal bin size than the corresponding Hi-C matrices) and created bed files indicating the coverage for each genomic bin. Subsequently, the bed files were normalized to obtain read per million values. Hi-C matrices were subsequently converted to coverage-normalized matrices using the R-base scale() function. Principal component analysis (PCA) were performed using f.principle.component.analysis() function from the HiCdatR package using either entire chromosomes or chromosome arms only, excluding pericentromeric regions (Chr1:13Mb-17Mb; Chr2:2Mb-5Mb; Chr3: 11.5Mb-15.5Mb; Chr4: 3Mb-5Mb; Chr5:10Mb;14Mb). Hi-C contact densities were assessed by calculating median interaction frequency of all contacts within 5 times the bin size of a given genomic bin. As input coverage-normalized Hi-C matrices were used. Significant differences between Hi-C data sets were determined by performing individual Student t-tests on each entry of the triplicate Hi-C matrices. Subsequently, we determined 5 categories of changes in contact frequencies (significantly less, less, no change, more, and significantly more). Significant difference matrices were generated using these categories and plotted with an according color code. All plots were generated using R-base (R Core Team 2018).

Transcriptome analyses

Total RNA was extracted from four pools of 3-week-old *Arabidopsis* plant leaf tissues of wild-type Col-0 or 20rDNA L6F6 using TRIzol reagent (MRC). Sequencing was performed by the Bioenvironment sequencing platform (Perpignan University, Perpignan, France) using a Nextseq 550 to generate 2 × 75-bp-long reads. Illumina reads from non-stranded, polyA+ RNA deep sequencing libraries were aligned to the *A. thaliana* TAIR10-annotated genome reference using HISAT 2 (Kim et al. 2015). Count the number of reads aligned to each genome coding sequences was performed with HTseq-count (Anders et al. 2015) and differential expression profile analyses with DESeq2 (Bioclite - R package) (Love et al. 2014).

Methylome analyses

Total genomic DNA was extracted from three pools of 3-week-old *Arabidopsis* plant leaf tissues of wild-type Col-0 or 20rDNA L6F6 using the Illustra Phytopure DNA extraction kit (GE Healthcare®, UK), following manufacturer's instructions. Bisulfite treatment, libraries preparation and sequencing were performed by the Novogene company (Hong-Kong) using the Illumina® technology. DMRs were identified with Bismark bisulfite mapper (Krueger and Andrews 2011), Methylkit (Akalin et al. 2012) and graph were generated using R (R Core Team 2018).

Immunostaining

Nuclei of *A. thaliana* plants were isolated and immunostained as described (Durut et al. 2014). An antibody targeting specifically the Phosphorylated γ-H2Ax was used at a dilution of 1:500 in PBS and applied onto a slide pre-coated with nuclei and incubated overnight at 4°C. The nuclei were labeled with anti-rabbit-Alexa Fluor 488 antibodies (Abcam) at a dilution of 1:1000, counterstained with DAPI (4', 6-diamidino-2-phenylindole), mounted using the Vectashield medium (Vector Laboratories), and analyzed in a Zeiss LSM 700 inverted confocal microscope.

Bleomycin assay

Sterile were sowed on Petri dishes containing medium ~ MS (Murashige and Skoog, Duchefa Biochemie M0231) 1X agar: 4.7g / L of MS medium, 5g / L of MES and 10g / L of Plant Agar (Gold Biotechnology) with a pH 5,7. For the treatment, media was supplemented with 1.5 ~ µg / mL bleomycin to obtain a 10^{-6}M concentration of bleomycin.

Nematode assay

For the infection assay, ten day-old *A. thaliana* seedlings aseptically grown on modified Knop's medium at 24°C under 16 h-light/8 h-dark were inoculated with surface-sterilized J2 *H. schachtii* nematodes (Baum et al. 2000). Four weeks after inoculation, adult females developing on each plant were counted, and the data were analyzed by a modified *t*-test using the Statistical Software Package SAS ($P<0.05$). Syncytium size measurements was performed 21 days after aseptic inoculation of *A. thaliana* with *H. schachtii* on modified Knop's in 100 mm petri dishes (Hewezi et al. 2008). For each line, 3 pool of 35 single-female syncytia were randomly selected, size was measured and average size for each line determined. Statistically significant differences were determined in a modified *t*-test using the statistical software package SAS ($P < 0.05$).

DC3000 infection assay and bacteria counting

Four plants per condition were dip-inoculated using *Pseudomonas* strain DC3000 at 5 x 10^7 cfu/ml (OD_{600} of 0.2 corresponds to 10 ^{8}cfu/ml) supplemented with 0.02 % Silwet L-77, and immediately placed in chambers with high humidity. Bacterial growth was determined 3 days after infection. For the quantification, infected leaves were harvested, washed for one minute in 70 % (v/v) EtOH and one minute in water. Leaf discs with a diameter of 5 mm have been excised, ground and homogenized in 200 μl of 10 mM $MgCl_2$. Each data point consists of four leaf discs. 10 μl of each homogenate were then plated undiluted and at different dilutions onto NYGA plates. Growth of bacteria was determined after 36 h of incubation at 28°C by colony counting. Statistical significance was tested with a Mann-Whitney non parametric *t*-test.

Fluorescence in situ hybridization (FISH), slide preparation, probe labelling

For nuclei isolation for FISH (T5H22, F5I10 ratio) 14 days-old seedlings of WT Col-0 and L6 (generation F6) were used. The seedlings were first fixed in 4% paraformaldehyde for 15 min at RT, chopped with a razor blade in ice cold nucleus isolation buffer (NIB - 0.5 M sucrose; 10 mM EDTA; 2.5 mM DTT; 100 mM KCl; 1 mM spermine; 4 mM spermidine in 10 mM Tris-Cl, pH = 9.5), and filtered through a 50 μm and 30 μm pore-sized disposable filters (CellTrics, Sysmex, Germany). Afterwards, the filtrate was supplemented with 1/10 volume of 10% Triton-X in NIB and centrifuged at 2000 x g for 10 minutes at 4°C. Pellet, containing nuclei, was then resuspended in 1x PBS and nuclei were dripped onto superfrost slides (10 μl / slide). After brief drying at 4°C, cells were fixed in chilled 3:1 methanol / acetic acid mixture for 10 minutes and air dried. After a rinse in 2x SSC, slides were incubated with RNase A (Dnase free, Applichem) in a humid chamber for 45 minutes at 37°C. Slides were then rinsed in 2xSSC and immersed in 10mM HCl for 5 minutes and subsequently in 10mM HCl with 8 μg/ml pepsin for 5 minutes (both at 37°C). After two washes in 2xSSC, slides were dehydrated in ethanol series (70%, 80%, 96%), 2 minutes each, and air-dried.

For pachytene chromosomes, entire inflorescences were fixed in ethanol:glacial acetic acid (3:1) overnight and stored in 70% ethanol at −20°C until use. WT Col-0, L6 (generation F4) and L9 (generation F6) were used. Selected immature flower buds were rinsed in distilled water (2 × 5 min) and in citrate buffer (10 mM sodium citrate, pH 4.8; 2 × 5 min), then anthers were dissected and incubated in an enzyme mix 0.15% each of cellulase (Onozuka R10, Serva), cytohelicase (Sigma), and pectolyase (Duchefa) in citrate buffer (10 mM sodium citrate buffer pH 4.5), at 4°C ON. Individual anthers were put on the microscope slide, disintegrated by the needle in a drop of citrate buffer. Then the suspension was softened by adding 15 to 30 µL of 50% acetic acid and spread by stirring with a needle on a hot plate at 50°C for 1 min. Chromosomes were fixed by adding 100 µL of ethanol:acetic acid (3:1) fixative, the slides were air dried, fixed in 4% formaldehyde dissolved in 2xSSC for 10 min, washed 3x5min in 2xSSC, and treated with RNase and pepsin (5-15min) as described above. After the pepsin treatment, slides were rinsed in 2xSSC, 3x5min and the fixation in paraformaldehyde was repeated, slides washed and dried in the ethanol series, 2 min each, air dried and hybridized with appropriate probes.

Hybridization conditions were used as follows: 50% formamide (Sigma)/10% dextran sulfate/2x SSC solution, denaturation for 3 minutes at 80 °C, overnight incubation at 37 ºC. Afterwards, slides were rinsed in 2x SSC at RT, then washed 2x5 min in 50% formamide in 2x SSC and 2x5 min in 2x SSC, (both at 42°C). Slides were blocked in 5% BSA in 4T buffer (4x SSC, 0.05% Tween-20) for 30 minutes at RT. After blocking, incubation with Alexa fluor 594-conjugated streptavidin (#S32356; Life technologies) and Alexa fluor 488-conjugated anti-digoxigenin (#IC75206; R&D systems) was performed for 45 minutes at RT. Antibodies were diluted in blocking buffer (1:200 and 1:10, respectively). Subsequently, slides were washed 2x5 min in 4T buffer and 2x5 min in 2xSSC buffer at 37°C with mild shaking.

Slides were stained with DAPI (4,6-diamidino-2-phenylindole, 2µg/ml) diluted in Vectashield (Vector laboratories), images were acquired using a Zeiss Axioimager Z1 or Olympus BX61, with appropriate filter set (AHF Analysentechnik, http://www.ahf.de/). Images were processed in Photoshop and Image J (https://imagej.nih.gov/ij/) softwares.

The BAC probes (T5H22, F5I10, available at www.arabidopsis.org) were kindly provided by T.Mandáková (CEITEC MU, Brno, CR). Approximately 1 μg of isolated BAC DNA (NucleoBond® Xtra Midi kit, Macherey-Nagel) were labeled by nick translation with biotin-dUTP or digoxigenin-dUTP (Jena Bioscience) using either commercial nick translation labelling kit (#07J00-001; Abbot - labelling was performed according to the manufacturer´s instructions using 5μl of the enzyme mix and 8h labelling time); or by home-made reaction mix (Mandakova and Lysak 2008), containing 5 µL of 10× NT buffer (0.5 M Tris-HCl, pH 7.5, 50 mM MgCl2, and 0.05% BSA), 10mM β-mercaptoethanol, 0.2mM each dATP, dCTP, dGTP and 40 μM dTTP (Jena Bioscience), 0.2 mM labelled-dUTP, 3 μL of DNase I (#10104159001, Roche ,diluted to 8 μg/ml) and 5U of DNA polymerase I (#EP0041,Fermentas).The nick translation mixture was incubated at 15°C for ~ 4h (or longer) to obtain a fragment length of ~200 to 500 bp. The reaction was stopped by adding 1 μL of 0.5 M EDTA, pH 8.0, and incubation at 65°C for 10 min.

Quantification of foci in isolated interphase nuclei

In the first analysis we counted the number of hk4 and subNOR4 signals in interphase nuclei and calculated their ratio. Two independent counts were performed and the ratio of hk4 to subNOR4 was evaluated foci in both wild-type cells and L6 cell line. To avoid conscious bias, one of the counts was blind and integrated all the foci into the analysis (singlet, doublet, partially split etc.) The results presented in Figure S1 are the average of the two counts.

The floral bud nuclei showed were 2, 3 or 4 of hk4 signals. The number and spatial distribution of these signals were evaluated. Total counts are shown in the table in (Figure S1).

SNP calling

Reads were mapped against the reference genome (*Arabidopsis* TAIR10) and single nucleotide polymorphisms called in Geneious (Bio- matters). Using R, single nucleotide polymorphisms were filtered for zygosity called based on the variant frequency provided by Geneious (>85% homozygous mutation) as previously described (Hristova et al. 2015).

Primer sequences used in this study:

Primer Name	sequence	use
AT1TE93275_F	CAACACAATTATGGGTCAGGT	qPCR Fig Suppl. 3
AT1TE93275_R	TCCAAGCCTCTTTCATTCTCA	qPCR Fig Suppl. 3

AT4G05475_F	CCAATGACGAACAAAGGAGTAA	qPCR Fig Suppl. 3
AT4G05475_R	TCACGAGAGTTTCAAGCAGT	qPCR Fig Suppl. 3
AT3G47210_F	AGCTTCTTCAACCTCTCAATG	qPCR Fig Suppl. 3
AT3G47210_R	GAAGTTTCTGTGTTTGACCCA	qPCR Fig Suppl. 3
AT5G22560_F	AAGAATGAATCTTGATGCAAGC	qPCR Fig Suppl. 3
AT1G53480_F	GCTCAGTGTTAAGAAGGTACAG	qPCR Fig Suppl. 3
AT1G53480_R	TCTTCACAACCAAGCTCGAAA	qPCR Fig Suppl. 3
AT4G38410_F	GAAGTTGCTGAAGATCAATGTG	qPCR Fig Suppl. 3
AT4G38410_R	CATCCTCATGGCCTATACCC	qPCR Fig Suppl. 3
AT4TE09845_F	CACGGTGGATGAATACATCAAA	qPCR Fig Suppl. 3
AT4TE09845_R	GAATCTACGCAAGCACTCTAA	qPCR Fig Suppl. 3
AT4TE10285_F	CGATCACATGAGTTAGCAACTT	qPCR Fig Suppl. 3
AT4TE10285_R	CCTATTGAGGCGAGGATTATCT	qPCR Fig Suppl. 3
AT4TE11395_F	CCCAATAAGGTAAATCTTCGGG	qPCR Fig Suppl. 3
AT4TE11395_R	CAACCAGAACCGAAACTGAC	qPCR Fig Suppl. 3
AT1G19250_FMO1_F	TTCTCCGACTTTCCATGGCC	RT-qPCR Figure 3
AT1G19250_FMO1_R	TGAGGAGTTTCGCCATCACC	RT-qPCR Figure 3
AT1G73805_SARD1_F	ACGAGGCTGCTCTTTTGACA	RT-qPCR Figure 3
AT1G73805_SARD1_R	TGTTTTGGTAGTGGCTCGCA	RT-qPCR Figure 3
AT1G75040_PR1_F	CCCGTCACTCTGGCTGAATT	RT-qPCR Figure 3
AT1G75040_PR1_R	CAGAGACACAGCCTGCGTAT	RT-qPCR Figure 3
AT1G80840_WRKY40_F	CAACCATCCAATGCCATCGC	RT-qPCR Figure 3
AT1G80840_WRKY40_R	GTAGTCACCGGCACAGTCAA	RT-qPCR Figure 3
AT3G50470_HR3_F	TCAAGACGGTCCAAGATGCA	RT-qPCR Figure 3
AT3G50470_HR3_R	GTTTCTGCGTCTGAGTTCCG	RT-qPCR Figure 3
AT4G23810_WRKY53_F	CTGTAGTCCCGGTGGCAAAT	RT-qPCR Figure 3
AT4G23810_WRKY53_R	TAGAACCTCCTCCATCGGCA	RT-qPCR Figure 3
AT5G64810_WRKY51_F	ACGGGTCATCGAGTTGCATT	RT-qPCR Figure 3
AT5G64810_WRKY51_R	GCTGCATCGTCACCATCTCT	RT-qPCR Figure 3
18S_F	CGTAGTTGAACCTTGGGATG	qPCR Figure 4
18S_R	CACGACCCGGCCAATTA	qPCR Figure 4
AT4G16580 F	TCTTGTTTCAAAGCGTTTCACT	qPCR Figure 4
AT4G16580 R	GATTCTTCCACGGTTAGGAC	qPCR Figure 4
AT4G05475 F	AATTGTTGCTAGCTATAACGC	qPCR Figure 4
AT4G05475 R	GAACAATGTGAGAACAAAGGAC	qPCR Figure 4
AT4TE12140_F	GGGTTCATATTCGGATCGGT	qPCR Figure 4
AT4TE12140_R	TCCGAACGGATCACAAATTTCA	qPCR Figure 4
AT4G05030 F	GGTAAGTATGAGGTGCGACAA	qPCR Figure 4
AT4G05030 R	TAGCATAGTCAAATGCCGGT	qPCR Figure 4
AT2G38250 F	TCAAGTGAGAATGGAGAGCG	qPCR Figure 4
AT2G38250 R	CTGCTCTTTCCTTCTCCCTA	qPCR Figure 4
AT2G38340 F	GTTTAGAGGTGTTCGACAAAGG	qPCR Figure 4

AT2G38340 R	AAGCCAAAGCCGTTTACTAC	qPCR Figure 4
3_fused-AT2G38460-AT2G37520	AAATCCAGTCGCTTCTCCGG	PCR Figure 5
5_fused-AT2G38460-AT2G37520	ATCTTGCGTTCCCGGACAAA	PCR Figure 5
5_fused_AT1G54770-AT1G55325	TGGTGGTCTACATAATGTTTCTTGG	PCR Figure 5
3_fused_AT1G54770-AT1G55325	CGCTTCCTGTAGAGAGCAGT	PCR Figure 5
5_fused-AT4G02960-AT4G05475	AACTTTTTGGACAACCGCCG	PCR Figure 5
3_fused-AT4G02960-AT4G05475	AGCCAGTGTTGGTTAGTCCG	PCR Figure 5
5_ef1alpha	CTAAGGATGGTCAGACCCG	PCR Figure 5
3_ef1alpha	CTTCAGGTATGAAGACACC	PCR Figure 5

Sanger sequences:

<u>Genomic DNA sequences:</u>

1. Consensus sequence obtained from the fused region of the genes AT1G54770 and AT1G55325

CTGAGCTGANTCCTGGCTTTGATAGAAGGTTTGATATGCCTGCTCCAACAATGAC
TCCTGAGTTGAAAAGAGATCTTCAGTTGCTTAAGTTGAGAACTGTAATGGATCCT
GCTCTACACTACAAGAAATCTGTGTCACGGTCTAAACTAGCTGAAAAATATTTCC
AGATTGGTACAGTAATTGAACCAGCTGAAGAGTTTTATGGGAGATTGACAAAGA
AGAATAGAAAAGCAACTCTTGCTGATGAGTTGGTTTCAGATCCTAAAACTGCTCT
TAC

Consensus sequence obtained for the fused region of the genes AT4G02960 and AT4G05475

TTCGTATCGTCTTGGGTGTGGCTGTTGATCGATCTTGGCCTATCCGTCAATTGGAT
GTCAATAATGCATTTCTATTACATATACTTTGTCATGGTTAAAGAAAACAAGTAG
TTAGGTTATTGGCATATATAAGAATATCATTATCTTATAGAACTCTCGTCATAGAC

Consensus sequence obtained for the fused region of the genes AT2G38460 and AT2G37520

GACAAAGCAAGTAGAAGGAAGAGATGCACCTGTTTCTGTTTCTATTGTTCCGGGA
ACAGAGGAGGGTTATACTGGAAACCCTCCGTCTAGAACCGGAATCTTGGTGATAC
TGGATAGAATGTCCAAGTCCTCTTTTGTTGGCGCATGGAGAATTTATTTCAATCAA
GAAGTTGTGGTAAAACATCACTCGACTGCTCTGTTTGTGAGCTACCTACAACGGA
TGCATCGTCACTTACAGTCTCTTCGGACCCAAAATCACTCCCGGAGCACTCAACT
AAACCGGATTTAACCGGCTGGCTAGA

<u>cDNA sequences:</u>

Consensus sequence obtained from cDNA of the genes AT1G54770 and AT1G55325

TCCTGGCTTTGATAGAAGGTTTGATATGCCTGCTCCAACAATGACTCCTGAGTTG
AAAAGAGATCTTCAGTTGCTTAAGTTGAGAACTGTAATGGATCCTGCTCTACACT
ACAAGAAATCTGTGTCACGGTCTAAACTAGCTGAAAAATATTTCCAGATTGGTAC
AGTAATTGAACCAGCTGAAGAGTTTTATGGGAGATTGACAAAGAAGAATAGAAA
AGCAACTCTTGCTGATGAGTTGGTTTCAGATCCTAAAACTGCTC

Supplemental code used to produce HiC figures

```
#GENERAL
library(HiCdatR)
library(plyr)
library(pastecs)

f.source.organism.specific.code("/home/stefan/CloudStation/Bioinformatik/HiCat/At_specific
_code.R")
InDir <- "/PATH/TO/HiCmatrices/"
plotDIR <- "/PATH/OUT/"

binSize <- 1e4
gffFileName <- "/PATH/to/TAIR10_GFF3_genes_transposons.gff"
gff <- gffRead(gffFileName, nrows = -1)
gff_genes_TE <- gff[gff[,3]=="gene" | gff[,3]=="transposable_element_gene"|
gff[,3]=="pseudogene",]
gff_genes_TE$name <- getAttributeField(gff_genes_TE$attributes, "Name")
TEs <- gff[gff[,3]=="transposable_element_gene",]
TEs$bin <- apply(TEs, 1, function(x)
f.translate.chrom.pos.to.index(x[1],as.numeric(x[4]),f.get.se.list(binSize),binSize))


DupList <- list(
                Dup1 = c("Chr1", 20435030, 20637825),
                Dup2 = c("Chr2", 6345226, 6423140),
                Dup3 = c("Chr2", 15749275, 16104798),
                Dup4 = c("Chr4", 1313751, 2766810),
                Dup5 = c("Chr4", 2517672, 2614425),
                Dup6 = c("Chr4", 10779530, 10839168),
                Dup7 = c("Chr5", 12439610, 12589340))

centromerePosList <- list(
  Chr1 = 15088987,
  Chr2 = 3608426.5,
  Chr3 = 13591999.5,
  Chr4 = 3956518.5,
  Chr5 = 11742754.5
)
chromLength <- list(
  Chr1 = 30427671,
  Chr2 = 19698289,
  Chr3 = 23459830,
  Chr4 = 18585056,
  Chr5 = 26975502#,
  #ChrM = 366924,
  #ChrC = 154478
)
chromCol <- list(
                    Chr1 = rgb(215,25,28, 100, max = 255),
                    Chr2 = rgb(253,174,97, 100, max = 255),
                    Chr3 = rgb(255,255,191, 100, max = 255),
                    Chr4 = rgb(171,217,233, 100, max = 255),
                    Chr5 = rgb(44,123,182, 100, max = 255)
)
chromosomes <- c("Chr1","Chr2","Chr3","Chr4","Chr5")
centromerePos <- list(
                         Chr1 = 15088987,
                         Chr2 = 3608426.5,
                         Chr3 = 13591999.5,
                         Chr4 = 3956518.5,
                         Chr5 = 11742754.5
                    )
periList <- list(
                    Chr1 = c(13e6,17e6),
                    Chr2 = c(2e6,5e6),
                    Chr3 = c(11.5e6,15.5e6),
                    Chr4 = c(3e6,5e6),
                    Chr5 = c(10e6,14e6)
)


sampleList_filtered <- list(
                    Col1 = paste("Col1_matrixR_filtered_",binSize/1000,"kb.txt", sep = ""),
                    Col2 = paste("Col2_matrixR_filtered_",binSize/1000,"kb.txt", sep = ""),
                    Col3 = paste("Col3_matrixR_filtered_",binSize/1000,"kb.txt", sep = ""),
                    rDNA1  =  paste("rDNA1_matrixR_filtered_",binSize/1000,"kb.txt",  sep  =
""),
                    rDNA2  =  paste("rDNA2_matrixR_filtered_",binSize/1000,"kb.txt",  sep  =
""),
                    rDNA3 = paste("rDNA3_matrixR_filtered_",binSize/1000,"kb.txt", sep = "")
)


SampleToColList <- list(
  Col1 = rgb(0,0,255,255,max=255),
  Col2 = rgb(0,50,200,255,max=255),
  Col3 = rgb(50,0,200,255,max=255),
  rDNA1 = rgb(255,0,0,255,max=255),
  rDNA2 = rgb(200,50,0,255,max=255),
  rDNA3 = rgb(200,0,50,255,max=255)
)

genoList <- list(
                        Col = c("Col1", "Col2", "Col3"),
                        rDNA = c("rDNA1", "rDNA2", "rDNA3")
)

#load the coverage data
cov <- read.table(paste(InDir,"cov_",binSize/1000,"kb.txt", sep = ""), sep = "\t", header =
TRUE)
colnames(cov)     <-     c(colnames(cov[1:4]),"Col1_R1",     "Col1_R2","Col2_R1",
"Col2_R2","Col3_R1",     "Col3_R2","rDNA1_R1",     "rDNA1_R2","rDNA2_R1",
"rDNA2_R2","rDNA3_R1", "rDNA3_R2")
cov <- cov[cov$chrom %in% chromosomes,]
covRPM <- as.data.frame(t((t(cov[5:ncol(cov)]) / colSums(cov[5:ncol(cov)]))*1e6))

#load the HiC data
Hi_C_list <- f.load.samples(dataDir = InDir,sampleToFiles = sampleList_filtered, binSize =
binSize, repetitions = 0)

#overview plots
for (name in names(sampleList_filtered)){
                        f.plot.XY.matrix(matrixToPlot = Hi_C_list[[name]], binSize = binSize,
axStep = 1e6, rDir=plotDIR, outfile = paste("Fred_filtered",name,binSize/1000,"kb",sep=""),
chromA = "ALL", startA = 0, endA = 0, chromB = "ALL", startB = 0, endB = 0, useLog =
TRUE,drawGrid = FALSE, doNorm = FALSE, doCor = FALSE, useSplineInterPol = TRUE)
}

#pool replicates and normalize to RPM
col_filtered <- Hi_C_list[["Col1"]]+Hi_C_list[["Col2"]]+Hi_C_list[["Col3"]]
colNorm_filtered                                                                      <-
col_filtered/(sum(Hi_C_list[["Col1"]]+Hi_C_list[["Col2"]]+Hi_C_list[["Col3"]]))*1e6
rDNA_filtered <- Hi_C_list[["rDNA1"]]+Hi_C_list[["rDNA2"]]+Hi_C_list[["rDNA3"]]
rDNANorm_filtered                                                                     <-
rDNA_filtered/(sum(Hi_C_list[["rDNA1"]]+Hi_C_list[["rDNA2"]]+Hi_C_list[["rDNA3"]]))
*1e6



#plot various kinds of differences
f.plot.signed.difference(col_filtered,     rDNA_filtered,     binSize=binSize,     rDir=plotDIR,
outfile="colvsrDNA_filtered_50kb",figureTitle = "hurz", filterZero = FALSE, filterThreshold
= 0.95,pValueThreshold = 0.05, randomizeDiff = FALSE)
f.plot.signed.difference(colNorm,     rDNANorm,     binSize=binSize,     rDir=plotDIR,
outfile="colvsrDNA_50kb",figureTitle = "hurz", filterZero = FALSE, filterThreshold =
0.95,pValueThreshold = 0.05, randomizeDiff = FALSE)

#normalize to RPM
HiClistNorm <- list()
for     (name     in     names(Hi_C_list)){HiClistNorm[[name]]     <-
(Hi_C_list[[name]]/sum(Hi_C_list[[name]]))*1e6}

#normalize HiCs for coverage
covNormList <- list()
```

```
for (name in names(HiClistNorm)[1:6]){
		covNormHiC <- scale(HiClistNorm[[name]], center=FALSE, scale=rowMeans(covRPM[,grep(name,colnames(covRPM))]))
		covNormHiC <- scale(t(covNormHiC), center=FALSE, scale=rowMeans(covRPM[,grep(name,colnames(covRPM))]))
		covNormHiC[is.na(covNormHiC)] <- 0 #in case coverage was 0
		covNormHiC[is.infinite(covNormHiC)] <- 0
		covNormHiC <- (covNormHiC/sum(covNormHiC))*1e6 # bring back to comparable levels (rpm)
		covNormList[[name]] <- covNormHiC
		f.plot.XY.matrix(matrixToPlot = covNormHiC, binSize = binSize, axStep = 1e6, rDir = plotDIR, outfile = paste("covNORM_notfiltered_",name,"_", binSize/1000,"kb",sep=""), chromA = "ALL", startA = 0, endA = 0, chromB = "ALL", startB = 0, endB = 0, useLog = TRUE,drawGrid = FALSE, doNorm = FALSE, doCor = FALSE, useSplineInterPol = TRUE)
}

rDNApool <- covNormList[["rDNA1"]] + covNormList[["rDNA2"]] + covNormList[["rDNA3"]]
Colpool <- covNormList[["Col1"]] +  covNormList[["Col2"]] + covNormList[["Col3"]]

f.plot.XY.matrix(matrixToPlot = rDNApool, binSize = binSize, axStep = 1e6, rDir = plotDIR, outfile = paste("Fred_covNORM_pool_rDNA_", binSize/1000,"kb",sep=""), chromA = "ALL", startA = 0, endA = 0, chromB = "ALL", startB = 0, endB = 0, useLog = TRUE,drawGrid = FALSE, doNorm = FALSE, doCor = FALSE, useSplineInterPol = TRUE)
f.plot.XY.matrix(matrixToPlot = Colpool, binSize = binSize, axStep = 1e6, rDir = plotDIR, outfile = paste("Fred_covNORM_pool_Col_", binSize/1000,"kb",sep=""), chromA = "ALL", startA = 0, endA = 0, chromB = "ALL", startB = 0, endB = 0, useLog = TRUE,drawGrid = FALSE, doNorm = FALSE, doCor = FALSE, useSplineInterPol = TRUE)


### PCA - prepare chromosome arms
PeriTab <- as.data.frame(rbind(c("Chr1", 13000001,17000000, "Peri1"),c("Chr2", 2000001,5000000, "Peri2"),c("Chr3", 11500001, 15500000, "Peri3"),c("Chr4", 3000001, 5000000, "Peri4"),c("Chr5", 10000001, 14000000, "Peri5")))
colnames(PeriTab) <- c("chrom", "start", "end", "name")
ChromArmTableExt <- as.data.frame(rbind(ChromArmTable, PeriTab), stringsAsFactor = FALSE)
ChromArmTableExt$start <- as.numeric(ChromArmTableExt$start)
ChromArmTableExt$end <- as.numeric(ChromArmTableExt$end)


SampleToColList <- list(
  Col1 = "red",
  Col2 = "orangered",
  Col3 = "violetred",
  rDNA1 = "navy",
  rDNA2 = "cornflowerblue",
  rDNA3 = "cyan"
)
```

```
##plotting for full chromosomes
#for (geno in names(genoList)[c(1,3)]){
			svg(paste(plotDIR,"PCASummary_",binSize/1000,"kb_Filterted_COLvs",g
eno,"_covNorm_chromArm.svg", sep=""),width=12,height=9)
			par(mfrow=c(5,2),mar=c(3, 4, 1, 2))

			for (name in ChromArmTable$name){
			#for (name in chromosomes){
				regionOI <- name
				plot(PCAlist[[names(PCAlist)[1]]][[regionOI]][,4],
PCAlist[[names(PCAlist)[1]]][[regionOI]][,2],type="n",
xlim=c(0,chromLength[["Chr1"]]),bty="n",ylim=c(-0.2,0.2),xlab="",ylab="")
				chrom <- substr(regionOI,1,4)
				if		(TEtab[TEtab[["chrom"]]==chrom,"numbTE"][1]		!=
0){TEtab[TEtab[["chrom"]]==chrom,"numbTE"][1] <- 0}

			polygon(TEtab[TEtab[["chrom"]]==chrom,"start"],(TEtab[TEtab[["chrom"]
]==chrom,"numbTE"]/50)-0.1,col=rgb(100,100,100,100,max=255),border=NA)

				lines(c(0, chromLength[[chrom]]),c(-0.1,-0.1), lwd = 3, col  = "orange")
				lines(c(periList[[chrom]][1],periList[[chrom]][2]),c(-0.1,-0.1), lwd = 3)
				points(centromerePosList[[chrom]],-0.1,
pch=16,cex=2,col="firebrick2")
				#comparison <- c(unlist(genoList[["Col"]]), unlist(genoList[[geno]]))
				comparison <- c(unlist(genoList[["Col"]]), unlist(genoList[["rDNA"]]))
				for (sample in comparison){
					if
(PCAlist[[sample]][[regionOI]][PCAlist[[sample]][[regionOI]][,4]==max(PCAlist[[sample]][
[regionOI]][,4]),2]<0){PCAlist[[sample]][[regionOI]][,2]				<-
PCAlist[[sample]][[regionOI]][,2]*-1}
					#used when replicates are fused
					#points(PCAlist[[sample]][[regionOI]][,4],
PCAlist[[sample]][[regionOI]][,2],type="l", col=SampleToColListFused[[sample]])
					#if(name=="Chr4")  {legend("topright",legend=names(PCAlist),
col=unlist(SampleToColListFused), pch=rep(16,length(names(PCAlist))),bty="n")}
					#used when replicates are there
					sub <- PCAlist[[sample]][[regionOI]]
					points(sub[,4],				sub[,2],type="l",
col=SampleToColList[[sample]])
#					pos <- as.data.frame(cbind(rep(0, nrow(sub)), sub[,4]))
#					pos[sub[,2] > 0,1] <- sub[sub[,2] > 0,2]
#					neg <- as.data.frame(cbind(rep(0, nrow(sub)), sub[,4]))
#					neg[sub[,2] < 0,1] <- sub[sub[,2] < 0,2]
#					polygon(c(pos[,2], rev(pos[,2])), c(pos[,1], rep(0, nrow(sub))),
col = "red")
#					polygon(c(neg[,2], rev(neg[,2])), c(neg[,1], rep(0, nrow(sub))),
col = "blue")
```

```
				if(name=="Chr4")	{legend("topright",legend=	comparison,
col=unlist(SampleToColList)[comparison], pch=rep(16,length(comparison)),bty="n")}
			}
		}
		dev.off()
#}
#########################################################
############detailed plots ############################
#########################################################

#select region of interest
chromOI <- "Chr4"
from <- 0e6
to <- 4e6

samples <- c("Col1", "Col2", "Col3", "rDNA1", "rDNA2", "rDNA3")
genotypes <- c("Col", "rDNA")

for (i in samples){ #for single files
  InDat <- paste(i,"_matrixR_10kb.txt", sep="")
  hicSubset <- f.minimal.HiC.loader(dataDir = InDir, fileName = InDat, chrom = chromOI,
from = from, to = to)
  hicNORM <- ((hicSubset)/sum(hicSubset))*1e6
  f.subset.hic(dataDir = InDir, fileName = inDat2, internalMatrix = hicNORM, chromA =
chromOI, startA = from, endA = to, binsize = binSize, geneOI = NULL, externalPlot = TRUE,
outDIR=plotDIR, outFile=paste(i,chromOI,from,to, "10kb_new", sep="_"), annotation =
gff_anno_oi, probes = NULL)
}

for (chr in chromosomes){
		chromOI <- chr
		from <- 0e6
		to <- chromLength[[chr]]

		fromIndex	<-	f.translate.chrom.pos.to.index(chromOI,	from,
f.get.se.list(binSize), binSize)
		toIndex <- f.translate.chrom.pos.to.index(chromOI, to, f.get.se.list(binSize),
binSize)
		RelCovRpm <- covRPM[fromIndex:toIndex,]

		hicNORMList <-list()
		for (geno in genotypes){ #for the pooled
		   hicSubsetList <- list()
		   for (rep in c(1:3)){
				InDat <- paste(geno,rep,"_matrixR_filtered_10kb.txt", sep="")
				hicSubsetList[[rep]] <- f.minimal.HiC.loader(dataDir = InDir,
fileName = InDat, chrom = chromOI, from = from, to = to)
		   }
		   cov <- rowSums(RelCovRpm[,grep(geno, colnames(RelCovRpm))])
			poolSubset <- hicSubsetList[[1]] + hicSubsetList[[2]] + 
hicSubsetList[[3]]
			covNormSubset <- scale(poolSubset, center=FALSE, scale = cov)
			covNormSubset <- scale(t(covNormSubset), center=FALSE, scale = 
cov)
			covNormSubset[is.na(covNormSubset)] <- 0 #in case coverage was 0
			covNormSubset <- (covNormSubset/sum(covNormSubset))*1e6 # bring 
back to comparable levels (rpm)
			source(externalFunctions)
			f.subset.hic(dataDir = InDir, fileName = NULL, internalMatrix = 
covNormSubset, chromA = chromOI, startA = from, endA = to, binsize = binSize, geneOI = 
NULL, externalPlot = TRUE, outDIR=plotDIR, outFile=paste(geno, 
"_pooled_",chromOI,"_",from,"_",to,"_",binSize/1000,"kb_NEW", sep=""), annotation = 
NULL, detailed = FALSE, probes = NULL, epiTrack = epiTrack, epiToPlot = c("H3K9me2", 
"transcription")) #gff_anno_oi
		}
}

#difference shown as ratio
diffHiC <- (hicNORMList[["rDNA"]]-hicNORMList[["Col"]])
diffHiCratio <- ((hicNORMList[["rDNA"]]+1)/(hicNORMList[["Col"]]+1))
diffHiCratio <- log2(diffHiCratio)
diffHiCratio[diffHiCratio > -1 & diffHiCratio < 1] <- 0
diffHiCratio[diffHiCratio <= -1] <- -1
diffHiCratio[diffHiCratio >= 1] <- 1
f.subset.hic(dataDir = InDir, fileName = inDat2, internalMatrix = diffHiCratio, chromA = 
chromOI, startA = from, endA = to, binsize = binSize, geneOI = NULL, externalPlot = TRUE, 
outDIR=plotDIR, outFile=paste("ratio",chromOI,from,to,"10kb", sep="_"), annotation = 
gff_anno_oi, probes = NULL, bindiff = FALSE, reldiff = TRUE)

#relative difference
diffHiCBinary <- diffHiC
diffHiCRel <- diffHiC
diffHiCRel[which(diffHiCRel<0)] <- diffHiCRel[which(diffHiCRel<0)]/(min(diffHiCRel)*-
1)
diffHiCRel[which(diffHiCRel>0)] <- diffHiCRel[which(diffHiCRel>0)]/(max(diffHiCRel))
f.subset.hic(dataDir = InDir, fileName = inDat2, internalMatrix = diffHiCRel, chromA = 
chromOI, startA = from, endA = to, binsize = binSize, geneOI = NULL, externalPlot = TRUE, 
outDIR=plotDIR, outFile=paste("rel_diff",chromOI,from,to,"10kb", sep="_"), annotation = 
gff_anno_oi, probes = NULL, bindiff = FALSE, reldiff = TRUE)

#binary difference
diffHiCBinary <- diffHiC
diffHiCBinary[which(diffHiCBinary<0)] <- -1
diffHiCBinary[which(diffHiCBinary>0)] <- 1
f.subset.hic(dataDir = InDir, fileName = inDat2, internalMatrix = diffHiCBinary, chromA = 
chromOI, startA = from, endA = to, binsize = binSize, geneOI = NULL, externalPlot = TRUE, 
outDIR=plotDIR, outFile=paste("binary_diff_",chromOI,from,to,"10kb", sep="_"), annotation 
= gff_anno_oi, probes = NULL, bindiff = TRUE, reldiff = FALSE)
```

```
######################## analysis differences with multiple t-tests
############################
library(qvalue) ## to get a better FDR estimation...

##illustrate the coverage
binSize <- 5e4

#to kill centromeres..
centromerePosList <- list(Chr1 = 15088987, Chr2 = 3608426, Chr3 = 13591999, Chr4 =
3956518, Chr5 = 11742754)
centromereIndex <- list()
for (name in names(centromerePosList)){centromereIndex[[name]] <-
c((f.translate.chrom.pos.to.index(name, centromerePosList[[name]], f.get.se.list(binSize),
binSize)-20):(f.translate.chrom.pos.to.index(name, centromerePosList[[name]],
f.get.se.list(binSize), binSize)+20)) }
centromereIndex <- as.numeric(unlist(centromereIndex))

cov50kb <-
read.table("/home/stefan/CloudStation/Oberassistenz/collaborations/FredPontvianne/HiCmatr
ices/cov_50kb.txt", sep = "\t", header = TRUE) # file made with HiCdat PRE "add tracks to
fragments"
colnames(cov50kb) <- c(colnames(cov50kb[1:4]),"Col1_R1", "Col1_R2","Col2_R1",
"Col2_R2","Col3_R1", "Col3_R2","rDNA1_R1", "rDNA1_R2","rDNA2_R1",
"rDNA2_R2","rDNA3_R1", "rDNA3_R2")
cov50kb <- cov50kb[cov50kb$chrom %in% chromosomes,]
covRPM <- as.data.frame(t((t(cov50kb[5:ncol(cov50kb)]) /
colSums(cov50kb[5:ncol(cov50kb)]))*1e6))

#plot the coverage
svg(file.path(plotDIR,"coverage_50kb.svg"),width=18,height=9)
YLIM <- c(0,1500)
par(mfrow = c(3,1))
plot(c(1:nrow(covRPM)), rowMeans(covRPM[,c(1:6)]), type = "n", ylim = YLIM, bty = "n",
col = "firebrick2", xaxt = "n", ylab = "coverage", main = "Col-0 WT")
polygon(c(1:nrow(covRPM),rev(1:nrow(covRPM))),c(rowMeans(covRPM[,c(1:6)]),rep(0,nro
w(covRPM))), col = "red", border  = NA)
for (chr in chromosomes){rect(xleft = f.get.se.list(binSize)[[chr]][1], ybottom = 0, xright =
f.get.se.list(binSize)[[chr]][2], ytop = 2000, border = NA, col = chromCol[[chr]])}
chromNameAt <- list()
for (name in names(centromerePosList)){chromNameAt[[name]] <-
f.translate.chrom.pos.to.index(name, centromerePosList[[name]], f.get.se.list(binSize),
binSize)}
text(x = unlist(chromNameAt), y = rep(1200, 5), labels = names(chromNameAt))
plot(c(1:nrow(covRPM)), rowMeans(covRPM[,c(7:12)]), type = "n", ylim = YLIM, bty = "n",
col = "orange", xaxt = "n", ylab = "coverage", main = "nuc1")
polygon(c(1:nrow(covRPM),rev(1:nrow(covRPM))),c(rowMeans(covRPM[,c(7:12)]),rep(0,nr
ow(covRPM))), col = "orange", border  = NA)
for (chr in chromosomes){rect(xleft = f.get.se.list(binSize)[[chr]][1], ybottom = 0, xright =
f.get.se.list(binSize)[[chr]][2], ytop = 2000, border = NA, col = chromCol[[chr]])}
```

```
plot(c(1:nrow(covRPM)), rowMeans(covRPM[,c(13:18)]), type = "n", ylim = YLIM, bty = "n",
col = "blue", xaxt = "n", ylab = "coverage", main = "20% rDNA")
polygon(c(1:nrow(covRPM),rev(1:nrow(covRPM))),c(rowMeans(covRPM[,c(13:18)]),rep(0,
nrow(covRPM))), col = "blue", border  = NA)
for (chr in chromosomes){rect(xleft = f.get.se.list(binSize)[[chr]][1], ybottom = 0, xright =
f.get.se.list(binSize)[[chr]][2], ytop = 2000, border = NA, col = chromCol[[chr]])}
xAt <- f.translate.chrom.pos.vector.to.index(c(rep("Chr1", 7), rep("Chr2", 4), rep("Chr3", 6),
rep("Chr4", 4), rep("Chr5", 6)), c(seq(0*1e6,30*1e6, by = 5*1e6), seq(0*1e6, 15*1e6, by =
5*1e6), seq(0*1e6, 25*1e6, by = 5*1e6), seq(0*1e6, 15*1e6, by = 5*1e6), seq(0*1e6, 25*1e6,
by = 5*1e6)), f.get.se.list(binSize), binSize)
axis(side  = 1, at = xAt, labels = c(seq(0,30, by = 5), seq(0, 15, by = 5), seq(0, 25, by = 5), seq(0,
15, by = 5), seq(0, 25, by = 5)))
dev.off()


HiClistNorm <- list()
for (name in names(Hi_C_list)){HiClistNorm[[name]] <-
(Hi_C_list[[name]]/sum(Hi_C_list[[name]]))*1e6}

covNormList <- list()
for (name in names(HiClistNorm)){
                    covNormHiC <- scale(HiClistNorm[[name]], center=FALSE,
scale=covRPM[,paste(name,"_R1", sep = "")])
                    covNormHiC <- scale(t(covNormHiC), center=FALSE,
scale=covRPM[,paste(name,"_R1", sep = "")])
                    covNormHiC[is.na(covNormHiC)] <- 0 #in case coverage was 0
                    covNormHiC <- (covNormHiC/sum(covNormHiC))*1e6 # bring back to
comparable levels (rpm)
                    covNormList[[name]] <- covNormHiC
                    f.plot.XY.matrix(matrixToPlot = covNormHiC, binSize = binSize, axStep =
1e6, rDir = plotDIR, outfile = paste("Fred_covNORM_",name, binSize,sep=""), chromA =
"ALL", startA = 0, endA = 0, chromB = "ALL", startB = 0, endB = 0, useLog = TRUE,drawGrid
= FALSE, doNorm = FALSE, doCor = FALSE, useSplineInterPol = TRUE)
}

#here we normalize that transinteraction appear stronger on the plot (each rectanlge is rpm)
for (name in names(covNormList)){
                    TransNorm <- covNormList[[name]]
                    for (chr in chromosomes){
                         for (chrom in chromosomes){
                              TempList <-
list(A=f.get.se.list(binSize)[[chr]],B=f.get.se.list(binSize)[[chrom]])
                              TransNorm[TempList$A,TempList$B] <-
(TransNorm[TempList$A,TempList$B]/sum(TransNorm[TempList$A,TempList$B]))*1e6
                              }
                    }
                    f.plot.XY.matrix(matrixToPlot = TransNorm*10, binSize = binSize, axStep
= 1e6, rDir = plotDIR, outfile = paste(name, "_", binSize/1000, "kb", "_trans-cov-Norm"),
chromA = "ALL", startA = 0, endA = 0, chromB = "ALL", startB = 0, endB = 0, useLog =
TRUE,drawGrid = FALSE, doNorm = FALSE, doCor = FALSE, useSplineInterPol = TRUE)
```

```
}


####### do the t-test


toCheck <- f.get.se.list(binSize)[["ALL"]]
diffHiCList <- list()
for (name in names(genoList)[3]){
#start_time <- Sys.time()
			ListOfMatrices1 <- list(covNormList[["Col1"]],covNormList[["Col2"]],covNormList[["Col3"]])
			ListOfMatrices2 <- list(covNormList[[genoList[[name]][1]]],covNormList[[genoList[[name]][2]]],covNormList[[genoList[[name]][3]]])
			diffHiCList[[name]] <- f.t.test.hiC.data.n3(x = ListOfMatrices1, y = ListOfMatrices2, ROI = toCheck, adjustP= FALSE, nthreads = 20)
			plotName <- paste("sign_diffence_filtered_noCovCorr",name,"vsCol_",binSize/1000,"kb", sep = "")
			svg(paste(plotDIR,plotName,".svg", sep = ""), width = 21, height = 21, pointsize = 18)
			#image((diffHiCList[[name]][["diffMat"]]),xlim=c(0,1.1), ylim = c(0,1.1), useRaster = TRUE, zlim = c(-2,2), col = c("red", rgb(240,128,128,150,max=255), "white", "lightblue1", "darkblue"), axes = FALSE) #prefect!!!! zlim is doing the job!!!!
			image((diffHiCList[[name]][["diffMat"]]),xlim=c(0,1.1), ylim = c(0,1.1), useRaster = TRUE, zlim = c(-2,2), col = c("red", "peachpuff", "white", "lightskyblue1", "navyblue"), axes = FALSE) #prefect!!!! zlim is doing the job!!!!
			#get the axis labels for one chrom
			# rangePlotted <- c(as.numeric(f.translate.index.to.chrom.pos(toCheck[1], f.get.se.list(binSize), binSize)[2]), as.numeric(f.translate.index.to.chrom.pos(toCheck[2], f.get.se.list(binSize), binSize)[2]))
			# axisRes <- 1e6
			# xLabs <- seq(rangePlotted[1],rangePlotted[2], by = axisRes)
			# xAt <- seq(0,(1/rangePlotted[2])*xLabs[length(xLabs)], length.out = length(xLabs))
			# axis(side = 1, at=xAt, labels = xLabs/1e6)
			# axis(side = 2, at=xAt, labels = xLabs/1e6)
			# legend(1,1, legend = c("sign_increase", "increase", "no change", "decrease", "sign_decrease"), col = c("red", rgb(240,128,128,150,max=255), "white", "lightblue1", "darkblue"), pch = 16, bty = "n", cex = 0.5)
			#for several chromosomes
			xLabs <- as.numeric(unlist(f.internal.axis.maker(binSize, axStep = 5e6, seList = f.get.se.list(binSize))))[1:26]
			xAt <- (1/f.get.se.list(binSize)[["ALL"]][2])*as.numeric(unlist(f.internal.axis.maker.on.index(binSize, axStep = 5e6, seList = f.get.se.list(binSize))))[1:26]
			axis(side = 1, at=xAt, labels = xLabs/1e6, cex.axis = 0.5)
			axis(side = 2, at=xAt, labels = xLabs/1e6, cex.axis = 0.5)
```

```
                    legend(1,1, legend = c("sign_increase", "increase", "no change", "decrease",
"sign_decrease"), col = c("red", rgb(240,128,128,150,max=255), "white", "lightblue1",
"darkblue"), pch = 16, bty = "n", cex = 0.5)
                    dev.off()
                    system(paste("rsvg-convert -p 300 -d 300 ",plotDIR,plotName,".svg", " >
",plotDIR,plotName,".png",sep=""))
}



#here we check which bins are significantly changed....
sign <- as.data.frame(which(abs(diffHiCList[["rDNA"]][["diffMat"]]) == 2, arr.ind =
TRUE))#2 and -2 are significant changes...
notsign <- as.data.frame(which(abs(diffHiCList[["rDNA"]][["diffMat"]]) == 1, arr.ind =
TRUE))#1 and -1 are non-significant changes...
totDiff <- colSums(abs(diffHiCList[["rDNA"]][["diffMat"]])) #total change is checked (sign
and insign..)
notvalid <- as.data.frame(which(abs(diffHiCList[["rDNA"]][["diffMat"]]) == 0, arr.ind =
TRUE))

signBin <- table(sign[,1])
notsignBin <- table(notsign[,1])
notvalidBins <- table(notvalid[,1])

SignBinTab                                                                          <-
as.data.frame(cbind(seq(1:f.get.se.list(binSize)[["Chr5"]][2]),rep(0,f.get.se.list(binSize)[["Chr
5"]][2]),rep(0,f.get.se.list(binSize)[["Chr5"]][2]),rep(0,f.get.se.list(binSize)[["Chr5"]][2])))

SignBinTab[as.numeric(names(signBin)),2] <- signBin
SignBinTab[as.numeric(names(notsignBin)),3] <- notsignBin
SignBinTab[as.numeric(names(notvalidBins)),4] <- notvalidBins

SignBinTab[,5] <- totDiff/rowSums(SignBinTab[,2:3])#we normalize to the number of valid
bins (no NA produced)
SignBinTab[is.na(SignBinTab[,5]),5] <- 0
SignBinTab[,6] <- SignBinTab[,2]/rowSums(SignBinTab[,2:3])
SignBinTab[is.na(SignBinTab[,6]),6] <- 0

#sign differences only
svg(paste(plotDIR,"sign_diffence_rDNAvsCol_50kb_perBin_CovCorr.svg", sep = ""), width
= 18, height = 9)
plot(SignBinTab[,1], SignBinTab[,6], type = "n", xlim = c(0,
f.get.se.list(binSize)[["Chr5"]][2]), xaxt = "n", xlab = "chromosomal position (Mb)", ylab =
"number of sign changed interactions")
for (chr in chromosomes){rect(xleft = f.get.se.list(binSize)[[chr]][1], ybottom = 0, xright =
f.get.se.list(binSize)[[chr]][2], ytop = 2000, border = NA, col = chromCol[[chr]])}
polygon(c(1:nrow(SignBinTab),rev(1:nrow(SignBinTab))),c(SignBinTab[,6],rep(0,nrow(Sign
BinTab))), col = "black", border  = NA)
xAt <- f.translate.chrom.pos.vector.to.index(c(rep("Chr1", 7), rep("Chr2", 4), rep("Chr3", 6),
rep("Chr4", 4), rep("Chr5", 6)), c(seq(0*1e6,30*1e6, by = 5*1e6), seq(0*1e6, 15*1e6, by =
5*1e6), seq(0*1e6, 25*1e6, by = 5*1e6), seq(0*1e6, 15*1e6, by = 5*1e6), seq(0*1e6, 25*1e6,
by = 5*1e6)), f.get.se.list(binSize), binSize)
axis(side  = 1, at = xAt, labels = c(seq(0,30, by = 5), seq(0, 15, by = 5), seq(0, 25, by = 5), seq(0,
15, by = 5), seq(0, 25, by = 5)))
dev.off()
system(paste("rsvg-convert -p 300 -d 300
",plotDIR,"sign_diffence_rDNAvsCol_50kb_perBin_CovCorr.svg", " >
",plotDIR,"sign_diffence_rDNAvsCol_50kb_perBin_CovCorr.png",sep=""))

#all differences
svg(paste(plotDIR,"all_diffence_rDNAvsCol_50kb_perBin_CovCorr.svg", sep = ""), width =
18, height = 9)
plot(SignBinTab[,1], SignBinTab[,5], type = "n", xlim = c(0,
f.get.se.list(binSize)[["Chr5"]][2]), xaxt = "n", xlab = "chromosomal position (Mb)", ylab =
"number of sign changed interactions")
for (chr in chromosomes){rect(xleft = f.get.se.list(binSize)[[chr]][1], ybottom = 0, xright =
f.get.se.list(binSize)[[chr]][2], ytop = 2000, border = NA, col = chromCol[[chr]])}
polygon(c(1:nrow(SignBinTab),rev(1:nrow(SignBinTab))),c(SignBinTab[,5],rep(0,length(Sig
nBinTab[,5]))), col = "black", border  = NA)
xAt <- f.translate.chrom.pos.vector.to.index(c(rep("Chr1", 7), rep("Chr2", 4), rep("Chr3", 6),
rep("Chr4", 4), rep("Chr5", 6)), c(seq(0*1e6,30*1e6, by = 5*1e6), seq(0*1e6, 15*1e6, by =
5*1e6), seq(0*1e6, 25*1e6, by = 5*1e6), seq(0*1e6, 15*1e6, by = 5*1e6), seq(0*1e6, 25*1e6,
by = 5*1e6)), f.get.se.list(binSize), binSize)
axis(side  = 1, at = xAt, labels = c(seq(0,30, by = 5), seq(0, 15, by = 5), seq(0, 25, by = 5), seq(0,
15, by = 5), seq(0, 25, by = 5)))
dev.off()
system(paste("rsvg-convert -p 300 -d 300
",plotDIR,"all_diffence_rDNAvsCol_50kb_perBin_CovCorr.svg", " >
",plotDIR,"all_diffence_rDNAvsCol_50kb_perBin_CovCorr.png",sep=""))




##########################################################
########## FUNCTIONS ####################################

gffRead <- function(gffFile, nrows = -1) {
  cat("Reading ", gffFile, ": ", sep="")
  gff = read.table(gffFile, sep="\t", as.is=TRUE, quote="",
               header=FALSE, comment.char="#", nrows = nrows,
               colClasses=c("character", "character", "character", "integer","integer","character",
"character", "character", "character"))
  colnames(gff) = c("seqname", "source", "feature", "start", "end", "score", "strand", "frame",
"attributes")
  cat("found", nrow(gff), "rows with classes:", paste(sapply(gff, class), collapse=", "), "\n")
  return(gff)
}
```

```
#### extract info from "attributes" field in gff files
getAttributeField <- function(x, field, attrsep = ";"){
  s = strsplit(x, split = attrsep, fixed = TRUE)
  sapply(s, function(atts){
    a = strsplit(atts, split = "=", fixed = TRUE)
    m = match(field, sapply(a, "[", 1))
    if (!is.na(m)){
      rv = a[[m]][2] }
    else {
      rv = as.character(NA) }
    return(rv)
  })
}


f.t.test.hiC.data.n3 <- function(x = ListOfMatrices1, y = ListOfMatrices2, ROI = toCheck,
adjustP= TRUE, nthreads = 2){#with triplicates
  require("parallel")
  dat <-
as.data.frame(cbind(as.vector(x[[1]][c(ROI[1]:ROI[2]),c(ROI[1]:ROI[2])]),as.vector(x[[2]][c(
ROI[1]:ROI[2]),c(ROI[1]:ROI[2])]),as.vector(x[[3]][c(ROI[1]:ROI[2]),c(ROI[1]:ROI[2])]),as
.vector(y[[1]][c(ROI[1]:ROI[2]),c(ROI[1]:ROI[2])]),as.vector(y[[2]][c(ROI[1]:ROI[2]),c(ROI
[1]:ROI[2])]),as.vector(y[[3]][c(ROI[1]:ROI[2]),c(ROI[1]:ROI[2])])))
  indexList <- split(c(1:nrow(dat)), c(1:nrow(dat)))
  startCalc <- Sys.time()
  pVec <- as.numeric(unlist(mclapply(indexList, function(x) t.test(dat[x,1:3],
dat[x,4:6])$p.value, mc.cores = nthreads)))
  endCalc <- Sys.time()
  endCalc - startCalc
  class(pVec)
  if (adjustP == TRUE ) {
    pVec[!is.na(pVec)] <- p.adjust(pVec[!is.na(pVec)], method = "BH")
  }
  pValMat <- matrix(pVec)
  pValMat[is.na(pValMat)] <- 1
  meanMat1 <-
matrix(rowMeans(cbind(as.vector(x[[1]][ROI[1]:ROI[2],ROI[1]:ROI[2]]),as.vector(x[[2]][R
OI[1]:ROI[2],ROI[1]:ROI[2]]),as.vector(x[[3]][ROI[1]:ROI[2],ROI[1]:ROI[2]]))), ncol =
length(ROI[1]:ROI[2]), nrow= length(ROI[1]:ROI[2]))
  meanMat2 <-
matrix(rowMeans(cbind(as.vector(y[[2]][ROI[1]:ROI[2],ROI[1]:ROI[2]]),as.vector(y[[2]][R
OI[1]:ROI[2],ROI[1]:ROI[2]]),as.vector(y[[2]][ROI[1]:ROI[2],ROI[1]:ROI[2]]))), ncol =
length(ROI[1]:ROI[2]), nrow= length(ROI[1]:ROI[2]))
  diffMat <- meanMat1 - meanMat2

  diffVec <- as.vector(diffMat)
  diffVec[diffVec < 0 & as.vector(pValMat) > 0.05] <- -1
  diffVec[diffVec > 0 & as.vector(pValMat) > 0.05] <- 1
  diffVec[diffVec < 0 & as.vector(pValMat) < 0.05] <- -2
  diffVec[diffVec > 0 & as.vector(pValMat) < 0.05] <- 2
  diffVec[diffVec == 0] <- 0
  diffMat <- matrix(diffVec, nrow = length(ROI[1]:ROI[2]), ncol = length(ROI[1]:ROI[2]))
  return(list(diffMat=diffMat, pValMat=pValMat))
}


f.subset.hic <- function(dataDir = NULL, fileName = NULL, internalMatrix = NULL, chromA
= chromOI, startA = from, endA = to, binSize = binSize, geneOI = "AT1G02580", externalPlot
= TRUE, outDIR=NULL, outFile=NULL, annotation = NULL, detailed = TRUE, probes =
probesTab, bindiff = FALSE, reldiff = FALSE, epiTrack = NULL, epiToPlot = c(5,6)){
  if(bindiff == TRUE & reldiff == TRUE){print("both, bindiff and reldiff is set on TRUE!!!")}
  library(HiCdatR)
  require("RColorBrewer")
  if (is.null(internalMatrix)){out <- f.minimal.HiC.loader(dataDir = dataDir, fileName =
fileName, chrom = chromA, from = startA, to = endA)}
  else {out <- internalMatrix}
  if (bindiff == TRUE | reldiff == TRUE){
    print(paste("bindiff=", bindiff, " ; reldiff=", reldiff, sep=""))
    HiCdata <- out
  }
  else {HiCdata <- log(out+1)}
  HiCdata[lower.tri(HiCdata)] <- NA
  nr <- nrow(HiCdata)
  nc <- ncol(HiCdata)
  d <- sqrt(nr^2 + nc^2) #pythagoras
  d2 <- 0.5 * d
  if (bindiff == TRUE){colMat <- matrix(surf.colors.bindiff(as.matrix(HiCdata)),nrow=nr-
1,ncol=nc-1)}##problem
  else if (reldiff == TRUE){colMat <-
matrix(surf.colors.reldiff(as.matrix(HiCdata))$colors,nrow=nr-1,ncol=nc-1)}##problem
  else {colMat <- matrix(surf.colors.light(as.matrix(HiCdata)),nrow=nr-1,ncol=nc-1)}
  raster <- as.raster(colMat)
  if (externalPlot == TRUE ) {pdf(paste(outDIR,outFile,".pdf", sep=""))}#best option to
export!!!
  baseline <- (nr-d2)*2 # this is half the height of the triangle down form the base of the
triangle...
  plot(NA, type="n", xlim=c(0, nc+d2), ylim=c(baseline , nr+d2), ylab="",yaxt="n", xaxt="n",
asp=1,bty="n", xlab="chromosomal postion (Mb)") #nc+d2: cause raster(angle=45) turns
around left corner, plot going up.... nr-d2: this is from the baseline of the triangel the same
distance down as to the top of the triangle... check it all out by using raster(angle=0)!!!!!
  if (externalPlot == TRUE) {
    addToExport <- (nc+d2)/2
    rasterImage(raster, xleft=0+d2, xright=nc+d2, ybottom=0+d2/2, ytop=nr+d2/2,
interpolate=FALSE, angle=45)
  }
  else {rasterImage(raster, xleft=0, xright=nc, ybottom=0, ytop=nr, interpolate=FALSE,
angle=45)}

  ##add features
  if (!is.null(annotation)){
```

```
  featureInInterval <- annotation[annotation$seqname == chromA & (annotation$start %in% c(startA:endA) | annotation$end %in% c(startA:endA)),]
  if (detailed == TRUE){
    features <- unique(annotation[["feature"]])
    NumbFeat <- length(features)*2 #for each feature two tracks (reverse and forward)
    FeatYpos <- seq(baseline, nr, length.out = NumbFeat+2)[2:(length(seq(baseline, nr, length.out = NumbFeat+2))-1)] # +2 we will not use the higest and lowest tracks keep distance to triangle..
    colPattern <- f.create.col.list(n = length(features), transparancy = 1, style="RdYlBu")
    featYposList <- list()
    for (j in c(1:length(features))){
      feat <- features[j]
      paste(feat,"Reverse",sep="")
      forwardYpos <- FeatYpos[c(TRUE,FALSE)][j]
      reverseYpos <- FeatYpos[c(FALSE,TRUE)][j]
      featYposList[[feat]] <- c(forwardYpos, reverseYpos, colPattern[j])
    }
    for (featType in unique(featureInInterval$feature)) {
      sub <- featureInInterval[featureInInterval$feature == featType,]
      for (i in c(1:nrow(sub))) {
        if (sub$strand[i] == "+") {
          featureStart <- f.translate.to.triangel.coord(sub$start[i]-startA)/binSize
          featureEnd <- f.translate.to.triangel.coord(sub$end[i]-startA)/binSize
          arrows(x0 = featureStart, y0 = as.numeric(featYposList[[featType]][1]), x1 = featureEnd, y1 = as.numeric(featYposList[[featType]][1]), length = 0.05, angle = 30, code = 2, col = featYposList[[featType]][3], lwd = 1)
        }
        if (sub$strand[i] == "-") {
          featureStart <- f.translate.to.triangel.coord(sub$start[i]-startA)/binSize
          featureEnd <- f.translate.to.triangel.coord(sub$end[i]-startA)/binSize
          arrows(x0 = featureStart, y0 = as.numeric(featYposList[[featType]][2]), x1 = featureEnd, y1 = as.numeric(featYposList[[featType]][2]), length = 0.05, angle = 30, code = 1, col = featYposList[[featType]][3], lwd = 1)
        }
      }
    }
    for (feat in features){text(0, mean(c(as.numeric(featYposList[[feat]][1]),as.numeric(featYposList[[feat]][2]))), labels=feat, adj=0, cex=0.5)}
    if (!is.null(geneOI)) { #label a specific gene
      LabelX <- round(mean(c(featureInInterval[featureInInterval$name == geneOI,"start"],featureInInterval[featureInInterval$name == geneOI,"end"])))
      LabelX <- f.translate.to.triangel.coord(LabelX-startA)/binSize
      text(LabelX, mean(c(as.numeric(featYposList[[featureInInterval[featureInInterval$name == geneOI,"feature"]]][1]), as.numeric(featYposList[[featureInInterval[featureInInterval$name == geneOI,"feature"]]][2]))), labels=geneOI, cex = 0.5)
    }
    if (reldiff == TRUE){
      cuts <- surf.colors.reldiff(as.matrix(HiCdata))$interval
```

```
        legendCol <- surf.colors.reldiff(as.matrix(HiCdata))$ColLegend
        legend("topright",cuts,col=legendCol,pch=16,bty="n",cex=0.5)
      }
    }
    ##for chromosome wide - make densities for features... (binned at 50 kb)
    else {
      featureInInterval$bin <- f.translate.chrom.pos.vector.to.index(featureInInterval$seqname, 
rowMeans(featureInInterval[,c("start","end")]), f.get.se.list(5e4), 5e4)
      startBIN <- f.translate.chrom.pos.to.index(chromA, startA, f.get.se.list(5e4), 5e4)
      endBIN <- f.translate.chrom.pos.to.index(chromA, endA, f.get.se.list(5e4), 5e4)
      binsTAB <- as.data.frame(cbind(c(startBIN:endBIN),rep(0, 
length(c(startBIN:endBIN))),rep(0, length(c(startBIN:endBIN)))))
      genes <- featureInInterval[featureInInterval$feature == "gene",]
      TE <- featureInInterval[featureInInterval$feature %in% 
c("transposable_element_gene","transposable_element"),]
      genesBin <- ddply(genes, "bin", function(x) nrow(x))
      TEsBin <- ddply(TE, "bin", function(x) nrow(x))
      binsTAB[binsTAB[,1] %in% genesBin[,1],2] <- genesBin[,2]
      binsTAB[binsTAB[,1] %in% TEsBin[,1],3] <- TEsBin[,2]
      binsTAB[,2] <- (binsTAB[,2]/mean(binsTAB[,2]))*15
      binsTAB[,3] <- (binsTAB[,3]/mean(binsTAB[,3]))*15
      binsTAB[,1] <- 
f.translate.to.triangel.coord(f.translate.index.vector.to.chrom.pos(binsTAB[,1], 
f.get.se.list(5e4), 5e4)[,2]/binSize)
      polygon(x = c(binsTAB[,1], rev(binsTAB[,1])), y = 
c(baseline+(binsTAB[,2]*5),rep(baseline, nrow(binsTAB))), border = NA, col = "blue")
      polygon(x = c(binsTAB[,1], rev(binsTAB[,1])), y = c(baseline-
(binsTAB[,3]*3),rep(baseline, nrow(binsTAB))), border = NA, col = "red")

    }
  }
  ##add epiTrack
  if (!is.null(epiTrack)){
    print("two epiTracks are allowed")
    startBIN <- f.translate.chrom.pos.to.index(chromA, startA, f.get.se.list(binSize), binSize)
    endBIN <- f.translate.chrom.pos.to.index(chromA, endA, f.get.se.list(binSize), binSize)
    epiTrack <- epiTrack[startBIN:endBIN,]
    xVal <- 
f.translate.to.triangel.coord(f.translate.index.vector.to.chrom.pos(c(1:nrow(epiTrack)), 
f.get.se.list(binSize), binSize)[,2]/binSize)
    polygon(x = c(xVal, rev(xVal)), y = c(baseline+(epiTrack[,epiToPlot[1]]),rep(baseline, 
length(xVal))), border = NA, col = "blue")
    polygon(x = c(xVal, rev(xVal)), y = c(baseline-(epiTrack[,epiToPlot[2]]),rep(baseline, 
length(xVal))), border = NA, col = "red")
    text(x = 0, y = (baseline + baseline/5), labels = epiToPlot[1], adj = c(0,1))
    text(x = 0, y = (baseline - baseline/5), labels = epiToPlot[2], adj = c(0,0))
  }
  ##add probes
  if (!is.null(probes)){
    probesInInterval <- probes[probes$chrom == chromA & (probes$start %in% c(startA:endA)
| probes$end %in% c(startA:endA)),]
    probesY <- baseline
    for (j in c(1:nrow(probesInInterval))){
      probesStart <- f.translate.to.triangel.coord(probesInInterval$start[j]-startA)/binSize
      probesEnd <- f.translate.to.triangel.coord(probesInInterval$end[j]-startA)/binSize
      lines(c(probesStart,probesEnd), c(probesY,probesY), col = "firebrick2")
    }
  }
  realAxPos <- pretty(c(startA:endA),n=10)
  xaxAt <- f.translate.to.triangel.coord(c(realAxPos - startA))/binSize
  axis(side = 1, at = xaxAt, labels = realAxPos)
  if (externalPlot == TRUE) {dev.off()}
}


f.minimal.HiC.loader <- function(dataDir = inDir, fileName = inDat, chrom = chromOI, from
= startA, to = endA){
  library(HiCdatR)
  start <- f.translate.chrom.pos.to.index(chrom, from, f.get.se.list(binSize), binSize)
  end <- f.translate.chrom.pos.to.index(chrom, to, f.get.se.list(binSize), binSize)
  inFile <- file.path(dataDir, fileName)
  temp <- read.table(text = system(paste("awk '{if ($1 >=",start, "&& $1 <=",end, "&& $2
>=",start, "&& $2 <=",end,") {print $0}}'", inFile, sep=" "), intern = TRUE), sep = '\t',
col.names = c("binA", "binB", "count"))
  out <- matrix(0, nrow=(end-start)+1, ncol=(end-start)+1)
  temp$binA <- (temp$binA-start)+1
  temp$binB <- (temp$binB-start)+1
  numberInteractions <- 0
  numberInteractions <- numberInteractions+sum(temp$count)
  toAdd <- out[cbind(temp$binA,temp$binB)] + temp$count
  out[cbind(temp$binA,temp$binB)] <- toAdd
  out[cbind(temp$binB,temp$binA)] <- toAdd
  out <- as.matrix(out)
  return(out)
}
```

References

Akalin A, Kormaksson M, Li S, Garrett-Bakelman FE, Figueroa ME, Melnick A, Mason CE. 2012. methylKit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles. *Genome Biol* **13**: R87.
Anders S, Pyl PT, Huber W. 2015. HTSeq--a Python framework to work with high-throughput sequencing data. *Bioinforma Oxf Engl* **31**: 166–169.
Baum TJ, Wubben MJ, Hardyy KA, Su H, Rodermel SR. 2000. A Screen for Arabidopsis thaliana Mutants with Altered Susceptibility to Heterodera schachtii. *J Nematol* **32**: 166–173.
Carpentier M-C, Picart-Picolo A, Pontvianne F. 2018. A Method to Identify Nucleolus-Associated Chromatin Domains (NADs). *Methods Mol Biol Clifton NJ* **1675**: 99–109.
Durut N, Abou-Ellail M, Pontvianne F, Das S, Kojima H, Ukai S, de Bures A, Comella P, Nidelet S, Rialle S, et al. 2014. A duplicated NUCLEOLIN gene with antagonistic activity is

required for chromatin organization of silent 45S rDNA in Arabidopsis. *Plant Cell* **26**: 1330–1344.
Grob S, Schmid MW, Grossniklaus U. 2014. Hi-C analysis in Arabidopsis identifies the KNOT, a structure with similarities to the flamenco locus of Drosophila. *Mol Cell* **55**: 678–693.
Hewezi T, Howe P, Maier TR, Hussey RS, Mitchum MG, Davis EL, Baum TJ. 2008. Cellulose binding protein from the parasitic nematode Heterodera schachtii interacts with Arabidopsis pectin methylesterase: cooperative cell wall modification during parasitism. *Plant Cell* **20**: 3080–3093.
Hristova E, Fal K, Klemme L, Windels D, Bucher E. 2015. HISTONE DEACETYLASE6 Controls Gene Expression Patterning and DNA Methylation-Independent Euchromatic Silencing. *Plant Physiol* **168**: 1298–1308.
Kim D, Langmead B, Salzberg SL. 2015. HISAT: a fast spliced aligner with low memory requirements. *Nat Methods* **12**: 357–360.
Krueger F, Andrews SR. 2011. Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinforma Oxf Engl* **27**: 1571–1572.
Langmead B, Trapnell C, Pop M, Salzberg SL. 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* **10**: R25.
Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinforma Oxf Engl* **25**: 2078–2079.
Love MI, Huber W, Anders S. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* **15**: 550.
Mandakova T, Lysak MA. 2008. Chromosomal phylogeny and karyotype evolution in x=7 crucifer species (Brassicaceae). *Plant Cell* **20**: 2559–2570.
Martin M. 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* **17**: 3.
Pontvianne F, Boyer-Clavel M, Saez-Vasquez J. 2016. Fluorescence-Activated Nucleolus Sorting in Arabidopsis. *Methods Mol Biol Clifton NJ* **1455**: 203–211.
R Core Team. 2018. R : A language and environment for statistical computing.
Schmid MW, Grob S, Grossniklaus U. 2015. HiCdat: a fast and easy-to-use Hi-C data analysis tool. *BMC Bioinformatics* **16**: 277.