Supplementary Information

Supplementary Tables and Legends

Supplementary Table 1. Comparison of vbSPT and Spot-On; Effect of HMM (vbSPT) on filtering out bound population. Summary of Spot-On inferred parameters for each condition: Diffusion Coefficients and Bound and Free Fractions. Please see attached spreadsheet for Supplementary Table 1.

Supplementary Table 2. Parameters for the simulations

Description	Parameters	Value
Polymer persistence length	l_0	600nm
Diffusion constant of the free protein	D_F	$5\mu m^2/\text{sec}$
Diffusion constant of the protein inside a zone (TTZ or PTZ)	D_Z	$2.5\mu m^2/\text{sec}$
Diffusion constant of the protein inside a CBS	D_{CBS}	$0.5\mu m^2/\text{sec}$
Trapping radius of CBS/TTZ	$\varepsilon_{CBS}\ /\varepsilon_{TTZ}$	$30\text{nm}/200\text{nm}$
Release radius of the protein from CBS/TTZ	$a_{CBS}\ /a_{TTZ}$	$40\text{nm}/210\text{nm}$
Characteristic time bound at the cognate binding site. Randomized from a Poisson distribution with mean τ_{CBS}	τ_{CBS}	1 minutes
Spring constant	k	$\frac{3k_BT}{(0.2l_0)^2}Nm^{-1}$
Monomer-monomer LJ distance	σ	l_0
Radius of the nucleus	A	$5\mu\text{m}$
Fraction of cognate binding sites	f_{CBS}	0.02
Fraction of the Power-law-distributed-Trapping Zones out total zone number	f_{PT}	(Figure4G-I): 0.2 (red,black)
Monomer number/ total number of zones	N	550
Size distribution of the Power-law Trapping Zones (PTZ)	$P_{PTZ}(\varepsilon)\sim A_1\varepsilon^{-\gamma}$	(Figure4G-I): $\gamma = 0.5$ The PTZ can have size between 100nm and 800nm. A_1 is a normalization coefficient
Release radius of the protein from the PTZ. Different for each PTZ, depending on its size	a_{PTZ}	$a_{PTZ} = \varepsilon_{PTZ} + 10\text{nm}$

CBS/TTZ		
Trapping probability to the Transiently Trapping Zone (TTZ)	$P_{trap,TTZ}$	(**Fig. 2d-h**): 0.99 (red), 0.1 (black) (**Fig. 3g-i**): 0.99 (red-wt), 0.2 (black-ΔRBR$_{i}$-CTCF), 0.01 (green ΔZF-CTCF)
Trapping probability to the cognate binding site (CBS)	$P_{trap,CBS}$	Equal to $P_{trap,TTZ}$
Trapping probability to the Power-law Trapping Zone (PTZ)	$P_{trap,PTZ}$	(**Fig. 3g-i**): 1 (red-wt), 1 (black-ΔRBR$_{i}$-CTCF), 0.01 (green ΔZF-CTCF)
Trapping exponent of the PTZs $P_{trap,PTZ}(\varepsilon)\sim\varepsilon^{-\delta}$	δ	(**Fig. 3g-i**): 0.5
Exit probability when hitting the boundary of a zone	P_{exit}	0.2
On-Rate to bind inside a TTZ or a PTZ	k_{on}	(**Fig. 2d-h**): 70msec$_{-1}$ (**Fig. 3g-i**): 1000msec$_{-1}$
Off-Rate to be released from a binding site inside a TTZ	$k_{off,TTZ}$	200 msec$_{-1}$
Off-Rate to be released from a binding site inside a PTZ	$k_{off,PTZ}$	2000 msec$_{-1}$

Supplementary Table 3. Non-Gaussian parameter (NGP). The non-Gaussian parameter (NGP)[1] given by $NGP = \left\langle \left(\Delta x\right) \right\rangle^4 / 3 \left\langle \left(\Delta x\right)^2 \right\rangle^2 - 1$ estimated for both the experimental data and the simulation corresponding to **Supplementary Figure 12**. NGP is 1 for the Laplace distribution and 0 for a Gaussian.

Δt sec	mESC C59 wt-Halo-mCTCF	mESC ΔZF-Halo-mCTCF (OE)	Model: wt	Model: ΔZF $P_{trap,TTZ} = 0.01$ $P_{trap,PTZ} = 0.01$
0.0045	0.3872	0.0517	1.1315	0.0344
0.0075	0.5421	0.0449	1.5436	0.0463
0.0135	0.6201	0.0881	0.8799	0.0603
0.0225	0.3568	0.1322	1.1153	0.0725
0.0292	0.4447	0.1258	1.1402	0.0801
0.0385	0.3978	0.1414	1.3389	0.0720

Supplementary Figures and Legends

Least squares power law fit: MSD = $4D\tau^{\alpha} + 4\sigma^2$ (σ: localization error); only first 50% of points used in the fitting; MSD calculated from HMM-classified "free" population
Code availability: https://gitlab.com/anders.sejr.hansen/anisotropy/blob/master/ProcessPlotFit_MSD.m

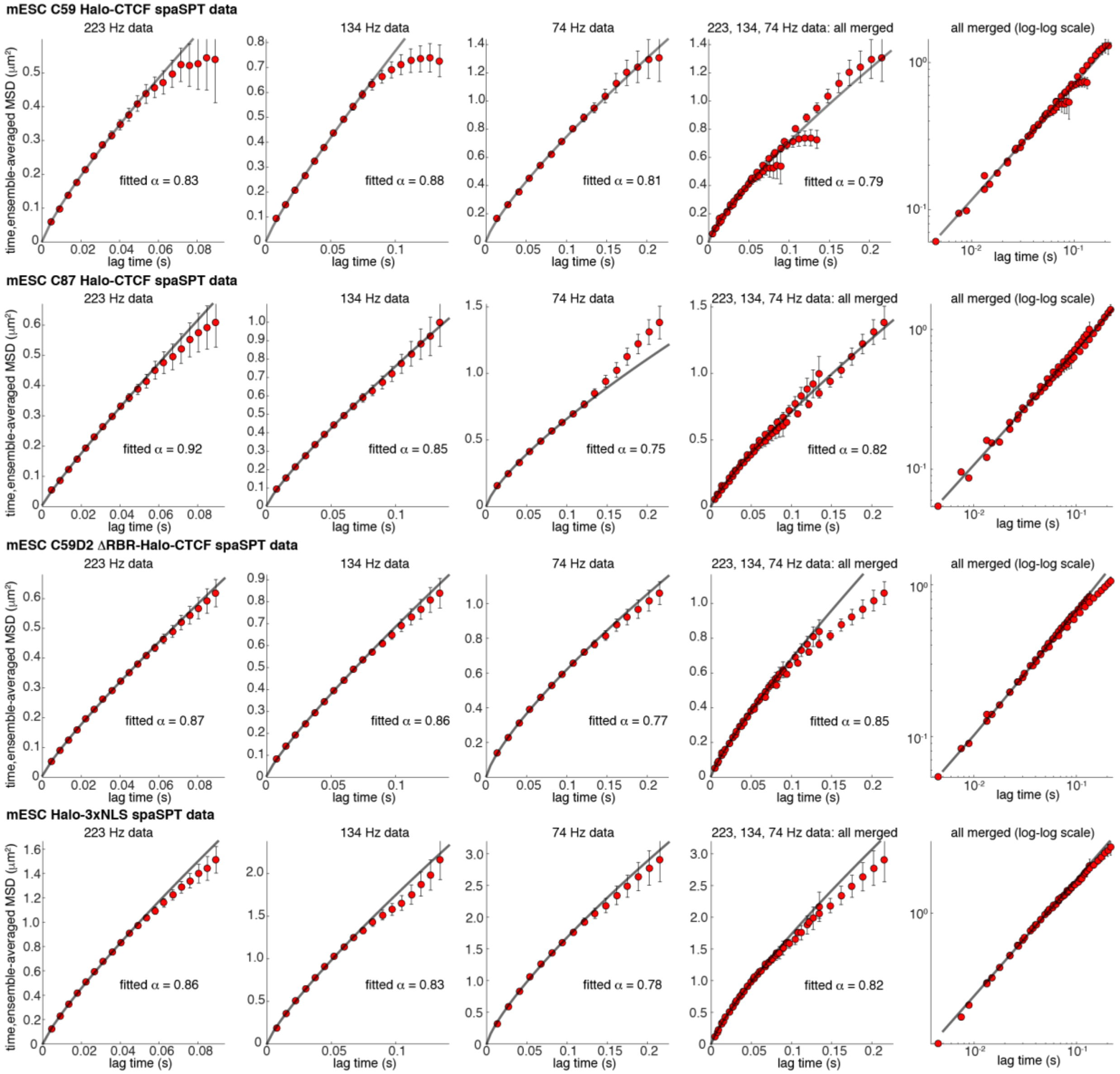


Table quantifying sensitivity of inferred α to fraction of points used in the fitting

		Percentage of data points used in fit (min=4)								
	Frame rate	20%	30%	40%	50%	60%	70%	80%	90%	100%
mESC C59 Halo-CTCF	223 Hz	0.85	0.89	0.87	0.83	0.81	0.79	0.77	0.73	0.69
	134 Hz	0.81	0.86	0.88	0.88	0.87	0.81	0.76	0.70	0.67
	74 Hz	0.75	0.77	0.79	0.81	0.81	0.83	0.83	0.81	0.78
	all merged	0.84	0.83	0.83	0.92	0.76	0.75	0.76	0.77	0.83
mESC C87 Halo-CTCF	223 Hz	0.86	0.93	0.93	0.85	0.90	0.88	0.85	0.85	0.82
	134 Hz	0.82	0.85	0.86	0.75	0.82	0.81	0.83	0.84	0.86
	74 Hz	0.73	0.74	0.76	0.83	0.77	0.82	0.84	0.89	0.91
	all merged	0.81	0.86	0.86	0.85	0.82	0.81	0.85	0.84	0.85
mESC C59D2 ΔRBR Halo-CTCF	223 Hz	0.86	0.89	0.88	0.87	0.85	0.85	0.84	0.84	0.83
	134 Hz	0.83	0.84	0.86	0.86	0.85	0.83	0.82	0.82	0.81
	74 Hz	0.77	0.77	0.77	0.77	0.76	0.74	0.74	0.73	0.72
	all merged	0.84	0.84	0.85	0.85	0.85	0.83	0.80	0.77	0.72
mESC Halo-3xNLS	223 Hz	0.90	0.87	0.87	0.86	0.83	0.82	0.81	0.80	0.79
	134 Hz	0.90	0.86	0.84	0.83	0.80	0.78	0.75	0.76	0.78
	74 Hz	0.86	0.84	0.79	0.78	0.76	0.74	0.74	0.74	0.74
	all merged	0.88	0.87	0.85	0.82	0.81	0.79	0.77	0.77	0.76

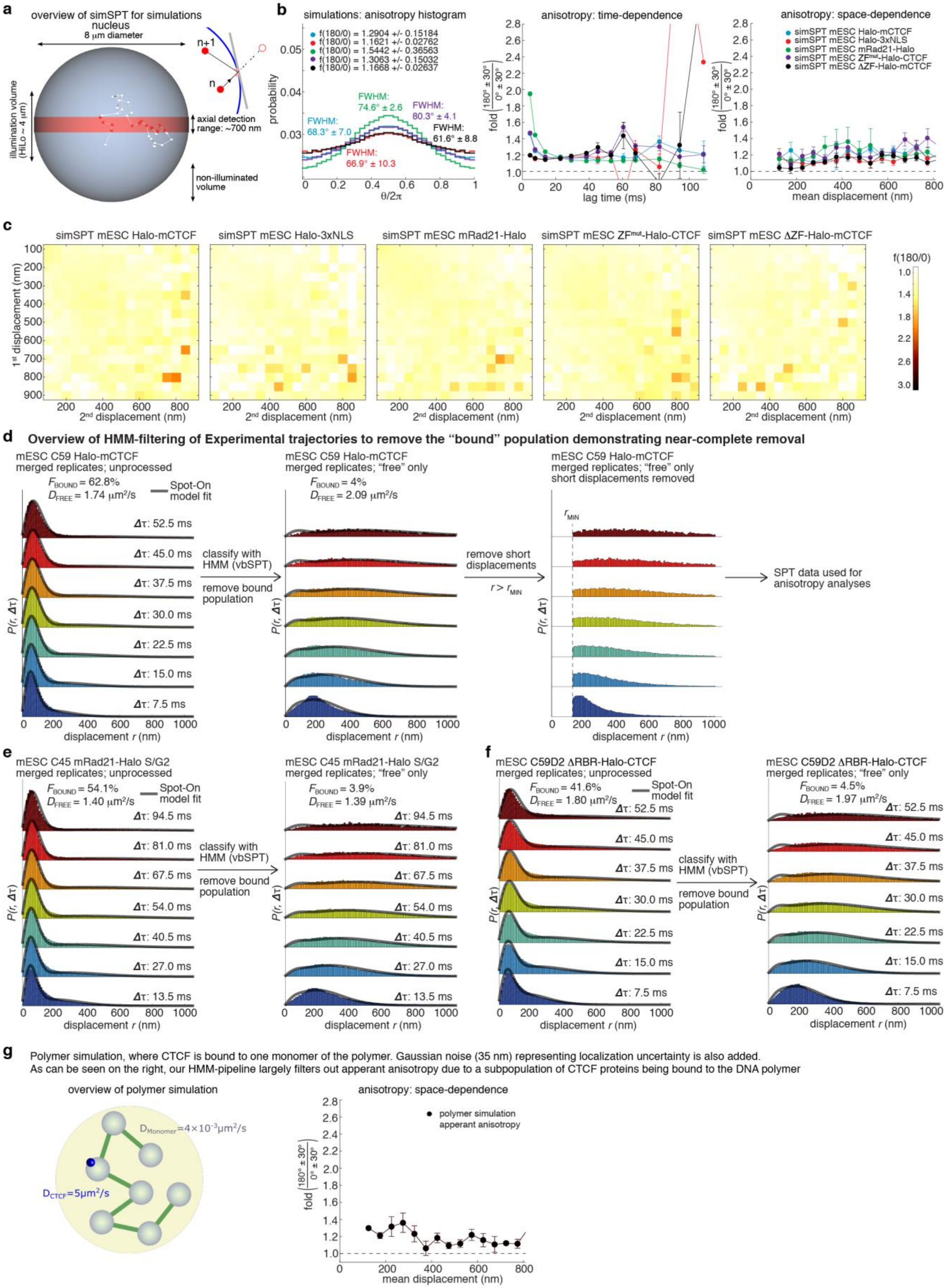
Supplementary Figure 1. MSD-analysis and anomalous CTCF diffusion.

Top: Mean-square displacement (MSD) analysis for mESC C59 Halo-CTCF (top row), mESC C87 Halo-CTCF (2nd row), mESC C59D2 ΔRBR$_i$-Halo-CTCF (3rd row) and mESC Halo-3xNLS (4th/bottom row). The data for each cell line was analyzed using code available at https://gitlab.com/anders.sejr.hansen/anisotropy. Briefly, using an HMM we filter out the bound population, such that the MSD is only calculated for the diffusing subpopulation. spaSPT[2,3] was performed at three frame rates (4.477 ms per frame or ~223 Hz, 7.477 ms per frame or ~134 Hz, and 13.477 ms per frame or ~74 Hz) and the MSD-fit is shown to each of the three datasets in the first three columns. Briefly, a power-law accounting for localization error was fit and the anomalous exponent, α, inferred. As suggested[4,5], we only used the first 50% of the data for the fitting. Error bars show standard deviation (from subsampling 50% of the data using 25 iterations) and the number of timepoints was limited to 20, 18, and 16 for 223 Hz, 134 Hz, and 74 Hz, respectively. The 4th column shows the MSD-fit with all the data merged. We show all four versions in part to emphasize that the inferred anomalous exponent, α, is quite sensitive to the data processing and therefore should be interpreted cautiously. Nevertheless, we believe the merged fit (column 4) most robustly represents the actual value. Finally, the 5th column shows the same data at a log-scale.

Bottom: Table quantifying the robustness of inferring the anomalous exponent, α, as a function of fitting parameters. As discussed previously[4,5], it can make sense to use only the more statistically relevant first fraction of the MSD-curve for fitting and we thus varied from 20% to 100%, which part of the MSD-curve we used in the fit. As expected, perhaps in part due to collisions with the nuclear envelope at long lag times, the estimated α-value decreases as we consider longer and longer lag times. As also pointed out previously, interpreting time- and ensemble-averaged MSD-curves based on multi-population SPT data is highly challenging[6]. Therefore, we suggest that the α-values reported here should be interpreted very cautiously.

Brownian motion simulations with realistic gaussian-distributed localization errors (σ_{1D} = 35 nm), with realistic HILO-photobleaching confined inside 8 μm spherical nucleus.
Simulated 500,000 trajectories at 223, 134, 74 Hz using simSPT: https://gitlab.com/tjian-darzacq-lab/simSPT; Representative command-line simulation:

```
>> ./simSPT -D1=0.02 -D2=2.5 -p1=0.65 -p2=0.35 -sigma=0.035 -dt=0.004477 -n_traj=500000 -file=./simulations/20171023_sim_mESC_Halo-mCTCF_223Hz_pooled_QC_CD2.csv -seed=0
```



d Overview of HMM-filtering of Experimental trajectories to remove the "bound" population demonstrating near-complete removal

g Polymer simulation, where CTCF is bound to one monomer of the polymer. Gaussian noise (35 nm) representing localization uncertainty is also added.
As can be seen on the right, our HMM-pipeline largely filters out apperant anisotropy due to a subpopulation of CTCF proteins being bound to the DNA polymer

Supplementary Figure 2. Control simulations demonstrating that apparent anisotropy is not an artifact of localization uncertainty, insufficient filtering of the bound population or polymer fluctuations of CTCF bound to the chromatin fiber.

Please note: Panels a, b, c and g show simulated data, whereas d, e and f show experimental data.

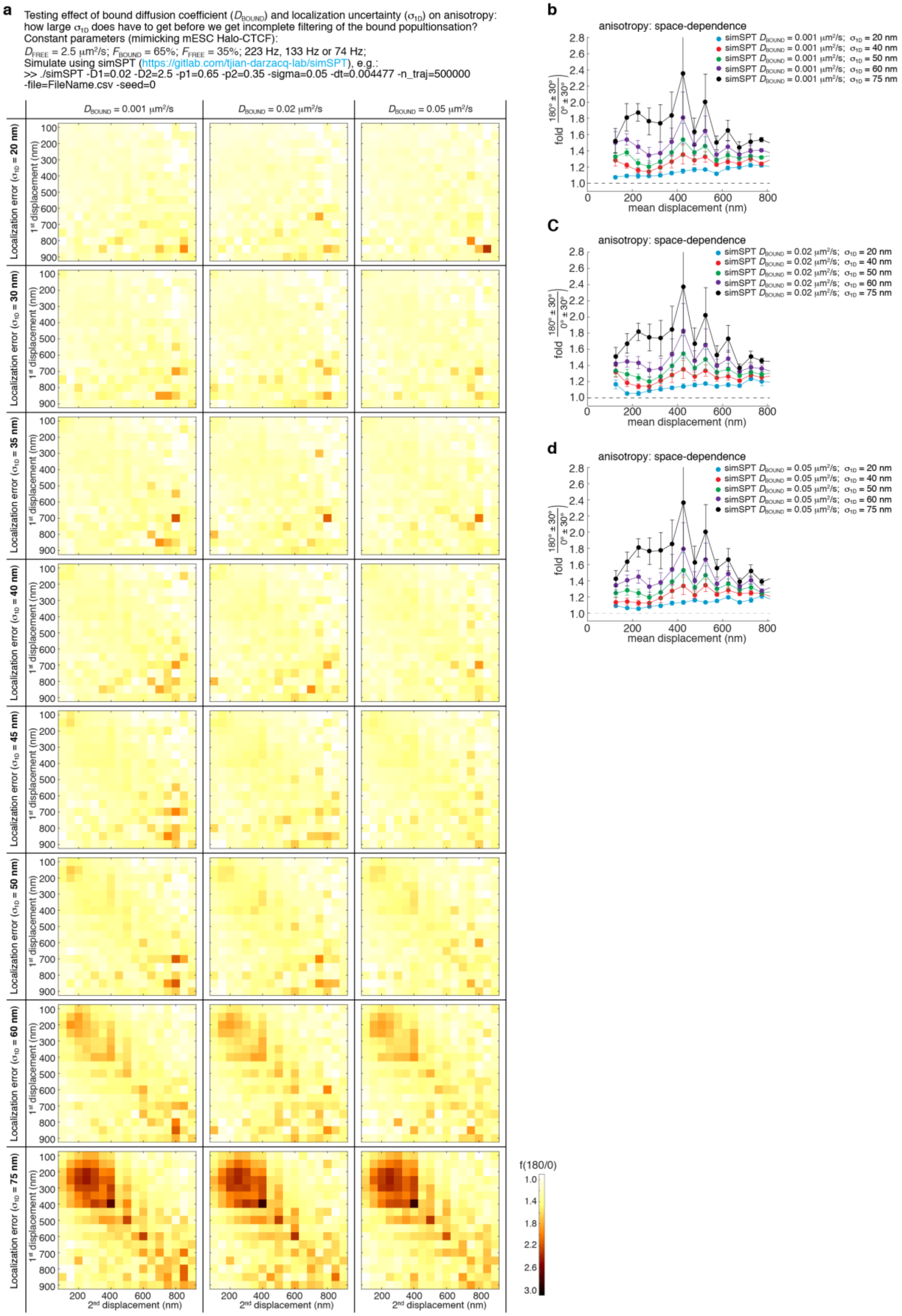
(**a**) simSPT simulations. Brownian motion with a mixture of bound and free subpopulations subject to significant localization error (std=35 nm) confined inside the nucleus can result in "apparent" anisotropy, since a bound molecule will appear to take steps back and forth around its true localization because of localization uncertainty. The purpose of these simulations was to assess whether or not our analysis pipeline fully filters this out. We performed these simulations using simSPT[2], which simulates experimentally realistic HILO SPT data. For each condition (e.g. C59 Halo-CTCF at 223 Hz), we "matched" the parameters (F_{BOUND}, D_{FREE}, σ) to the data (parameters obtained from analysis with Spot-On[2]) and then simulated 500,000 trajectories and analyzed them the same way as the experimental data.

(**b**) Analysis of simSPT simulations. Representative plots of fold-anisotropy, $f_{180/0}$, at the bulk level (left), as a function of the lag time (middle) and as a function of the mean displacement length (right) for the indicated "matched" experiments: mESC Halo-CTCF (F_{BOUND}=0.65; D_{FREE}=2.5 $\mu m^2/s$), mESC Halo-3xNLS (F_{BOUND}=0.10; D_{FREE}=6.0 $\mu m^2/s$), mESC Rad21-Halo G1 (F_{BOUND}=0.50; D_{FREE}=1.5 $\mu m^2/s$), mESC ZF_{mut}-Halo-CTCF (F_{BOUND}=0.30; D_{FREE}=2.5 $\mu m^2/s$), mESC ΔZF-Halo-CTCF (F_{BOUND}=0.05; D_{FREE}=5.5 $\mu m^2/s$). As can be seen, simulated Rad21 is mildly anisotropic at ~223 Hz, but otherwise, the analysis pipeline successfully filters out "apparent" anisotropy. This is from n=1 simulation and based on 500,000 simulated trajectories. Error bars show standard deviation from 50 subsamplings with replacement using 50% of the data and centre values show value using 100% of the data.

(**c**) Anisotropy heatmaps. Plot of $f_{180/0}$ as a function of the length of the first and second displacement. Most notably, the anisotropy "bump" around 200 nm is fully absent in the simulations.

(**d-e**) Reliability of HMM-pipeline (vbSPT) to remove the bound population. To remove the bound population and perform anisotropy analysis, our pipeline takes 2 steps (**Fig. 1b**): first HMM-mediated classification into Free and Bound states. We remove the Bound state. Second, we remove displacements below a certain minimum threshold, r_{min}. To test if the HMM efficiently removes the bound population, we applied it to our entire experimental SPT data (full results in **Supplementary Table 1**). Here we show a few examples. To test if the bound fraction was removed, we ran Spot-On before and after filtering. If the bound fraction is efficiently removed, then the Spot-On inferred bound fraction should be close to zero after filtering. Indeed this was the case: the bound fraction was minimal after HMM-filtering as shown in (**d-f**) and in **Supplementary Table 1**.

(**g**) Polymer simulation. Fold anisotropy of the diffusion protein is estimated in a simulation where the polymer is diffusing. The simSPT analysis above (**a-c**) was applied to estimate the fold anisotropy of the protein in a simulation where the polymer is also diffusing, but simSPT models the polymer using Brownian motion. We tested if our HMM-pipeline would filter out the motion of protein cognately bound to chromatin, which is due to chromatin dynamics. Left: Chromatin is described as a chain of 50 monomers (blue spheres) that interact with the potential described in the Methods (see "Brownian simulation of chromatin and the protein"). Each monomer corresponds to a cognate binding site (CBS) to which the protein binds with a characteristic time of 1 minute. There are no TTZs or PTZs monomers in the simulation. An additional Gaussian noise with standard deviation of 35 nm and zero mean was added to the position of the protein in the simulation to account for the localization error. The diffusion coefficient of the protein is $D_{CTCF} = 5\mu m^2/s$ while the diffusion coefficient of the polymer (chromatin) is the empirically estimated diffusion coefficient of a fluorescently chromatin locus on the X chromosome in mESC ($D_{Monomer} = 4 \times 10^{-3}\mu m^2/s$ [7]). Effectively, there are two subpopulations of CTCF in the simulation. One is freely diffusing and the other is bound to the polymer Right: The fold anisotropy of the protein as a function of the mean step size following the HMM pipeline. Our HMM-pipeline largely filters out any apparent anisotropy coming from the DNA polymer fluctuations. The residual value of the anisotropy (~1.2) is scale-free, and similar in value to that of the free, Halo-3xNLS (**Fig. 1c**). This rules out that the observed anisotropy could be due to polymer motion. The data shown in (**g**) is the result of n=100 independent simulations. In each simulation, the trajectory of CTCF was recoded until its 10000's capture event. Error bars in (**g**) show standard deviation from 50 subsamplings with replacement using 50% of the data and centre values show value using 100% of the data.



Testing effect of bound diffusion coefficient (D_{BOUND}) and localization uncertainty (σ_{1D}) on anisotropy: how large σ_{1D} does have to get before we get incomplete filtering of the bound popultionsation?
Constant parameters (mimicking mESC Halo-CTCF):
D_{FREE} = 2.5 μm²/s; F_{BOUND} = 65%; F_{FREE} = 35%; 223 Hz, 133 Hz or 74 Hz;
Simulate using simSPT (https://gitlab.com/tjian-darzacq-lab/simSPT), e.g.:

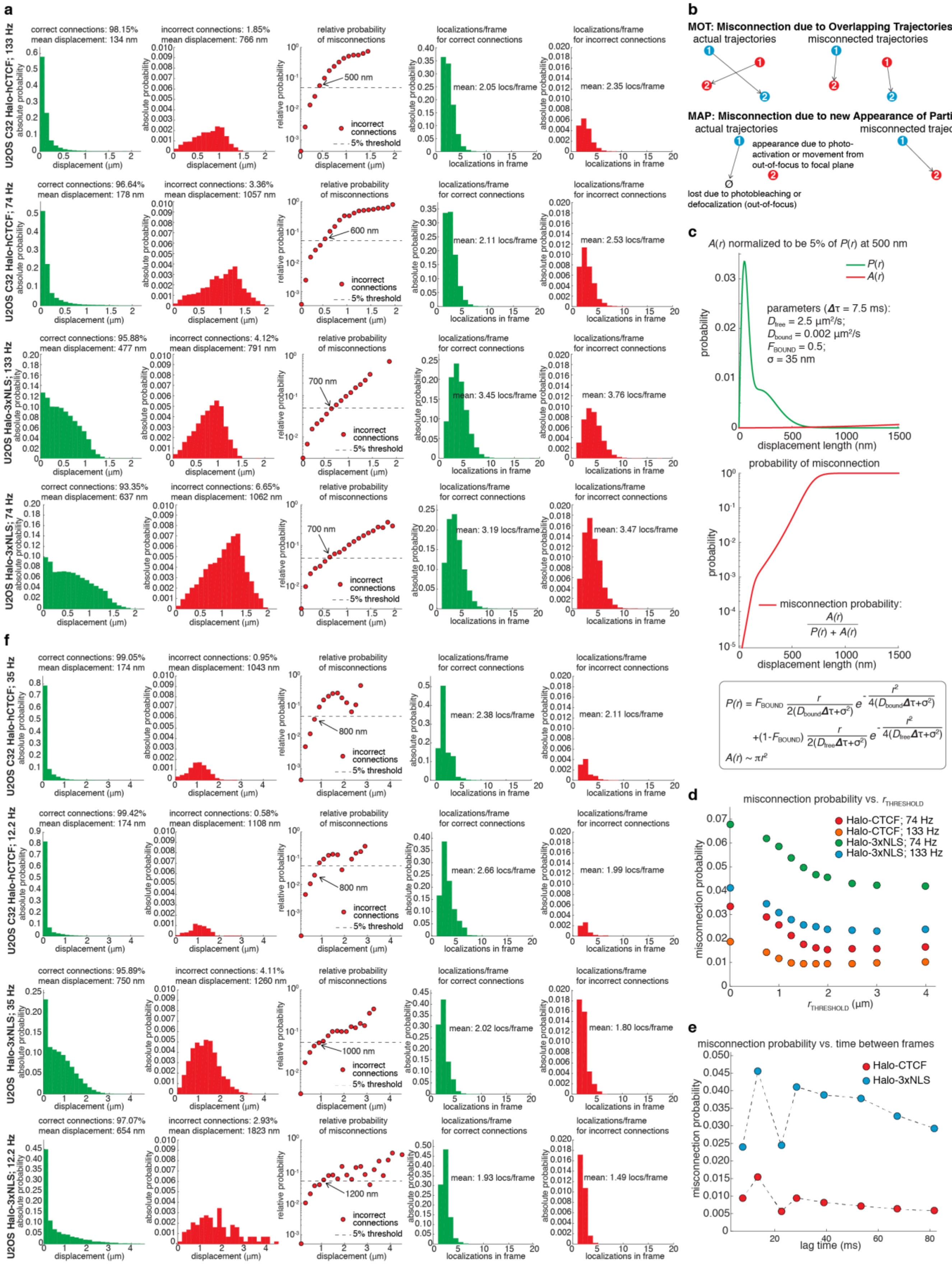
```
>> ./simSPT -D1=0.02 -D2=2.5 -p1=0.65 -p2=0.35 -sigma=0.05 -dt=0.004477 -n_traj=500000
-file=FileName.csv -seed=0
```

a
D_{BOUND} = 0.001 μm²/s
D_{BOUND} = 0.02 μm²/s
D_{BOUND} = 0.05 μm²/s
Localization error (σ_{1D} = 20 nm)
Localization error (σ_{1D} = 30 nm)
Localization error (σ_{1D} = 35 nm)
Localization error (σ_{1D} = 40 nm)
Localization error (σ_{1D} = 45 nm)
Localization error (σ_{1D} = 50 nm)
Localization error (σ_{1D} = 60 nm)
Localization error (σ_{1D} = 75 nm)
1st displacement (nm)
100
200
300
400
500
600
700
800
900
2nd displacement (nm)
200
400
600
800
f(180/0)
1.0
1.4
1.8
2.2
2.6
3.0
b
anisotropy: space-dependence
simSPT D_{BOUND} = 0.001 μm²/s; σ_{1D} = 20 nm
simSPT D_{BOUND} = 0.001 μm²/s; σ_{1D} = 40 nm
simSPT D_{BOUND} = 0.001 μm²/s; σ_{1D} = 50 nm
simSPT D_{BOUND} = 0.001 μm²/s; σ_{1D} = 60 nm
simSPT D_{BOUND} = 0.001 μm²/s; σ_{1D} = 75 nm
fold $\frac{180° \pm 30°}{0° \pm 30°}$
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0
200
400
600
800
mean displacement (nm)
C
anisotropy: space-dependence
simSPT D_{BOUND} = 0.02 μm²/s; σ_{1D} = 20 nm
simSPT D_{BOUND} = 0.02 μm²/s; σ_{1D} = 40 nm
simSPT D_{BOUND} = 0.02 μm²/s; σ_{1D} = 50 nm
simSPT D_{BOUND} = 0.02 μm²/s; σ_{1D} = 60 nm
simSPT D_{BOUND} = 0.02 μm²/s; σ_{1D} = 75 nm
fold $\frac{180° \pm 30°}{0° \pm 30°}$
mean displacement (nm)
d
anisotropy: space-dependence
simSPT D_{BOUND} = 0.05 μm²/s; σ_{1D} = 20 nm
simSPT D_{BOUND} = 0.05 μm²/s; σ_{1D} = 40 nm
simSPT D_{BOUND} = 0.05 μm²/s; σ_{1D} = 50 nm
simSPT D_{BOUND} = 0.05 μm²/s; σ_{1D} = 60 nm
simSPT D_{BOUND} = 0.05 μm²/s; σ_{1D} = 75 nm
fold $\frac{180° \pm 30°}{0° \pm 30°}$
mean displacement (nm)

Supplementary Figure 3. Additional control simulations demonstrating that apparent anisotropy peak at ~200 nm is not an artifact of localization uncertainty or chromatin mobility.

(**a**) simSPT simulations processed through anisotropy pipeline. simSPT simulates Brownian motion (**Supplementary Fig. 2a**) under realistic HiLo-experimental conditions subject to localization uncertainty and both a bound and free population. Here, we tested how the diffusion coefficient of the chromatin bound population (D_{BOUND} = {0.001, 0.02, 0.05} μm^2/s; shown as different columns) and the 1-dimensional localization error or uncertainty (defined as the Gaussian standard deviation: σ_{1D} = {20, 30, 35, 40, 45, 50, 60, 75} nm; shown as different rows), affect our anisotropy analyses with the other parameters chosen to approximately fit Halo-CTCF in mESCs (as shown in the plot). As can be seen, the bound diffusion coefficient makes a minimal contribution to apparent anisotropy. But predictably, at very high localization uncertainty, apparent movement due to localization errors becomes poorly distinguishable from true movement of the free population and thus is no longer filtered out efficiently, which causes apparent anisotropy. However, this only happens at high localization error $\sigma_{1D} \geq 60$ nm, which is much higher than our experimental uncertainty of 35 nm.

(**b-d**) Apparent anisotropy as a function of mean displacement length. For different bound diffusion coefficients, we plotted how the apparent anisotropy scales mean displacement length as a function of the 1D localization error σ_{1D}. As can be seen, at high enough localization uncertainty, we do get apparent anisotropy, but it does not manifest itself as a sharp peak ~200 nm as seen for CTCF in our experimental data and this only occurs at high localization error $\sigma_{1D} \geq 60$ nm, which is much higher than our experimental uncertainty of 35 nm. (**b-d**) are from n=1 simulation and based on 500,000 simulated trajectories. Error bars show standard deviation from 50 subsamplings with replacement using 50% of the data and centre values show value using 100% of the data.



Supplementary Figure 4. 2-color spaSPT experiments to estimate the misconnection probability during tracking.

(**a**) Misconnection probability for U2OS Halo-CTCF and U2OS Halo-3xNLS. Cells were labeled ~1:1 with the distinguishable PA-JF$_{549}$ and PA-JF$_{646}$ dyes[8]. This way, we can directly observe tracking misconnections (e.g. green-to-red or red-to-green). We correct for the ~2-fold undercounting. We imaged at a ~2-3-fold higher density than normal (~2-3 particles per frame per nucleus) to estimate the worst-case level of tracking errors. Histograms of displacement lengths are shown for correct connections (1st column) and misconnections (2nd column). The 3rd column shows that the misconnection probability increases *exponentially* with displacement length. Histograms of number of localizations per frame for correct and incorrect connections are shown in column 4 and 5, respectively.

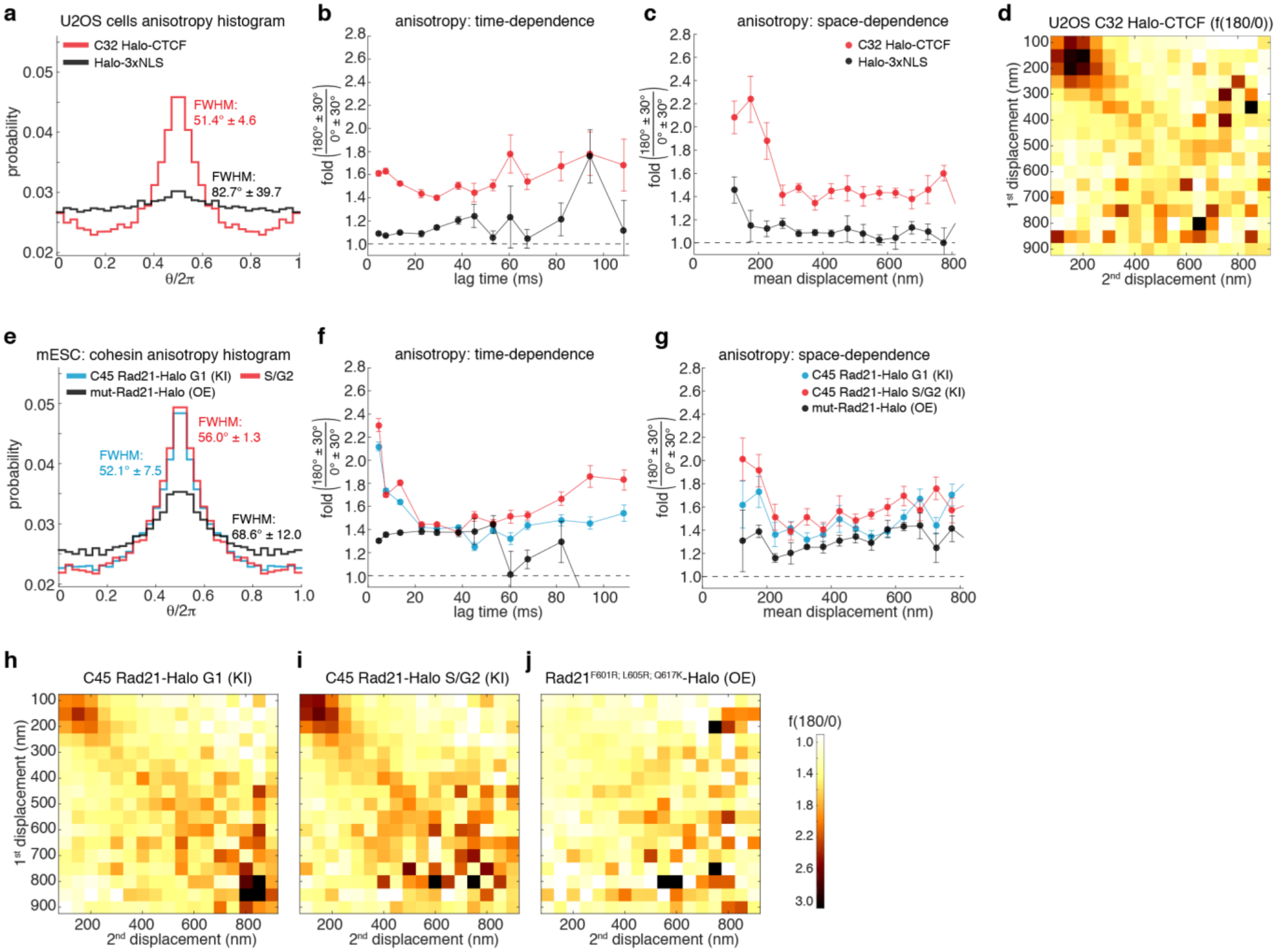
(**b**) Schematics. Schematics illustrating two types of misconnections: Misconnection due to overlapping trajectories (MOT) and Misconnection due to new Appearance of Particle (MAP).

(**c**) Why tracking errors increase exponentially. "back-of-the-envelope" illustration of why tracking errors increase exponentially with displacement length. Although the misconnection probability presumably scales with the area of a circle (radius square), because correct connections decrease exponentially with displacement length, misconnection probability will increase exponentially with displacement length.

(**d**) Close encounter filter. To minimize misconnections, we applied a close encounter filter, $r_{\text{THRESHOLD}}$, such that whenever two particles come closer than $r_{\text{THRESHOLD}}$ they will be cut short. Choosing $r_{\text{THRESHOLD}}$ = 2 μm optimally reduced the misconnection probability.

(**e**) How misconnections scale with lag time.

(**f**) Misconnection probability for U2OS Halo-CTCF and U2OS Halo-3xNLS. Same as (**A**), but for temporally subsampled data (35 Hz, 12.2 Hz).



Supplementary Figure 5. Anisotropy for CTCF in human U2OS cells and cohesin in mESCs.

Anisotropy plots for C32 Halo-CTCF and Halo-3xNLS in human U2OS cells showing fold-anisotropy, $f_{180/0}$, at the bulk level (**a**), as a function of the lag time (**b**) and as a function of the mean displacement length (**c**).

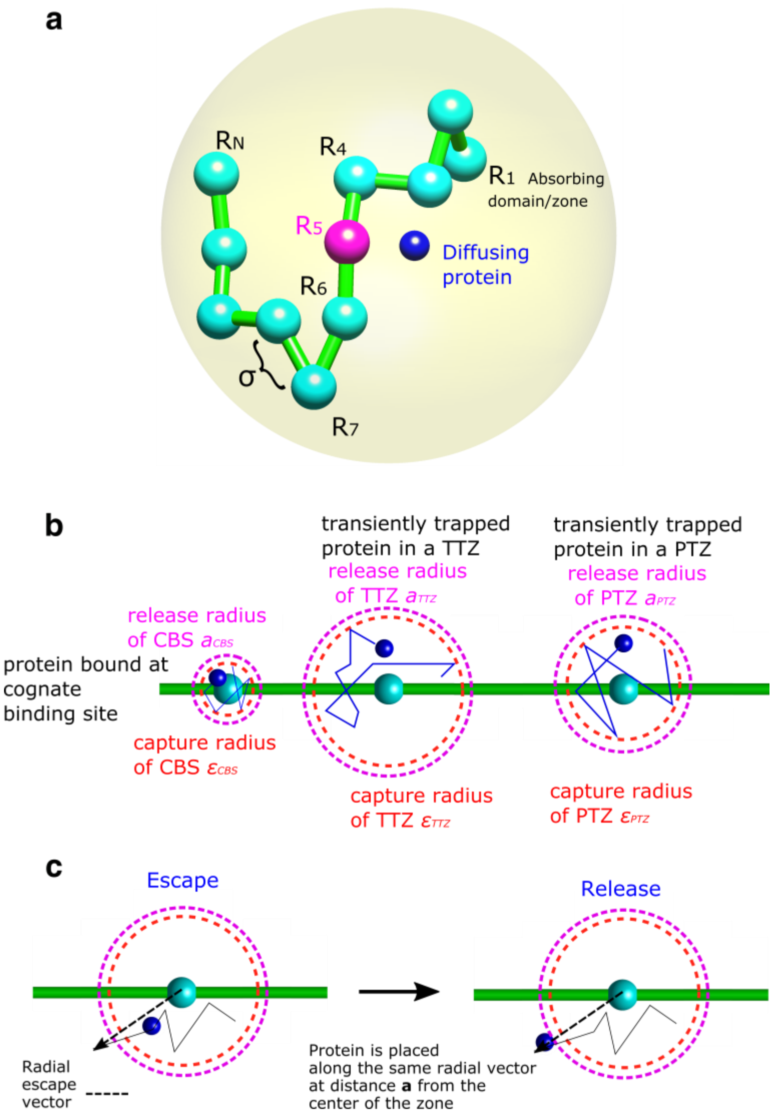
(**d**) Anisotropy heatmap. Plot of $f_{180/0}$ as a function of the length of the first and second displacement for U2OS C32 Halo-CTCF.

Anisotropy plots for C45 Rad21-Halo (knock-in) in either G1-phase or S/G2 phase of the cell cycle, and a mut-Rad21-Halo, which cannot form cohesin complexes[3] (transiently over-expressed) in mESCs showing fold-anisotropy, $f_{180/0}$, at the bulk level (**e**), as a function of the lag time (**f**) and as a function of the mean displacement length (**g**).

Error bars in plots above show standard deviation from 50 subsamplings with replacement using 50% of the data and centre values show value using 100% of the data.

(**h-j**) Anisotropy heatmaps. Plot of $f_{180/0}$ as a function of the length of the first and second displacement for the indicated cell lines.

Here KI refers to endogenous knock-in cell lines and OE refers to exogenous over-expression using an optimized protocol as outlined in **Supplementary Fig. 10**.

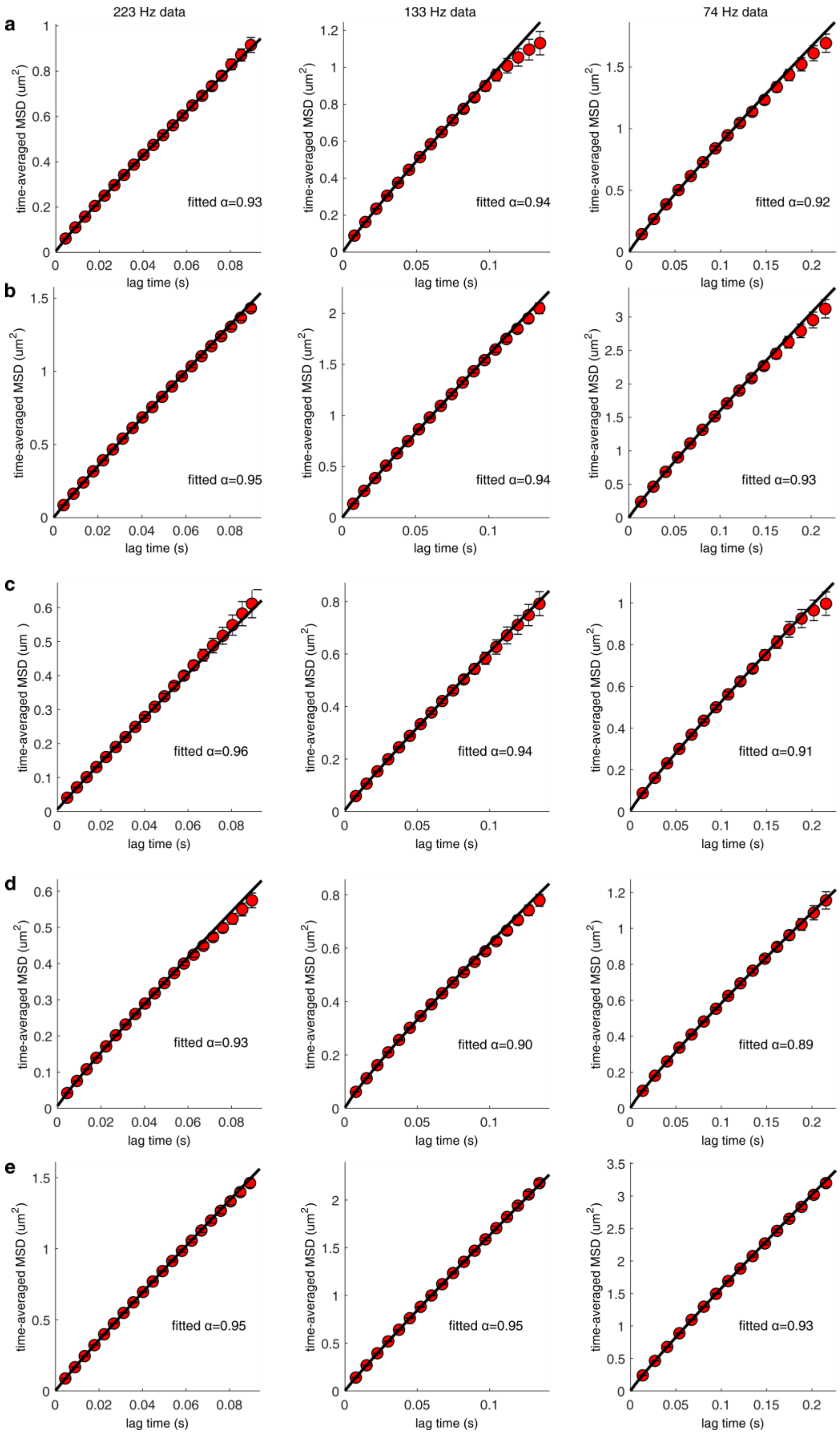

a
RN
R4
R1 Absorbing domain/zone
R5
Diffusing protein
R6
σ
R7
b
transiently trapped protein in a TTZ
release radius of TTZ a_{TTZ}
transiently trapped protein in a PTZ
release radius of PTZ a_{PTZ}
release radius of CBS a_{CBS}
protein bound at cognate binding site
capture radius of CBS ε_{CBS}
capture radius of TTZ ε_{TTZ}
capture radius of PTZ ε_{PTZ}
c
Escape
Release
Radial escape vector -----
Protein is placed along the same radial vector at distance **a** from the center of the zone

Supplementary Figure 6. Overview of chromatin polymer simulations.

Overview of the model. (**a**) Chromatin is described as a long chain of N monomers (beads, cyan) connected by harmonic springs. The monomers additionally interact via LJ interactions with a LJ size σ. Each monomer corresponds to one of three zones/domains (TTZs,CBSs,PTZs). The monomer assignment is random. The polymer is equilibrated within a domain (nucleus) which has a radius of 5μm. The CTCF protein is a Brownian particle (dark blue). The protein was released from monomer (zone) 5 after being trapped there (magenta).

(**b**) (**Left)** The protein is absorbed/trapped with probability $P_{trap,CBS}$ if it arrives to a distance ε_{CBS} (red dashed-line circle) from the center of a cognate binding site (CBS). It is then trapped there for a characteristic time τ_{CBS}. When released, the protein is positioned with equal probability on a sphere centered around the CBS with a radius a_{CBS} (magenta dashed-line circle). (**Middle**) The protein is absorbed/trapped with probability $P_{trap,TTZ}$ if it arrives to a distance ε_{TTZ} (red dashed-line circle) from the center of a Transiently Trapping Zone (TTZ). It diffuses within the TTZ which has radius ε_{TTZ}. When released, the protein is positioned on a sphere center around the CBS with a radius a_{TTZ}, at the same angular direction from which it escaped (magenta dashed-line circle). (**Right**) The protein is absorbed/trapped with probability $P_{trap,PTZ}$ if it arrives to a distance ε_{PTZ} (red dashed-line circle) from the center of a Power-law Trapping Zone (PTZ). The distance ε_{PTZ} is randomized from a distribution (see Methods). It diffuses within the PTZ which has radius ε_{PTZ}. When released, the protein is positioned on a sphere center around the CBS with a radius a_{PTZ}, at the same angular direction from which it escaped (magenta dashed-line circle)

(**c**) When the protein escapes from the zone it is released from the spherical border (radius a_{TTZ} or a_{PTZ} depending on the zone from which it escaped) along the same radial vector from the center of the zone it had when it crossed the border.



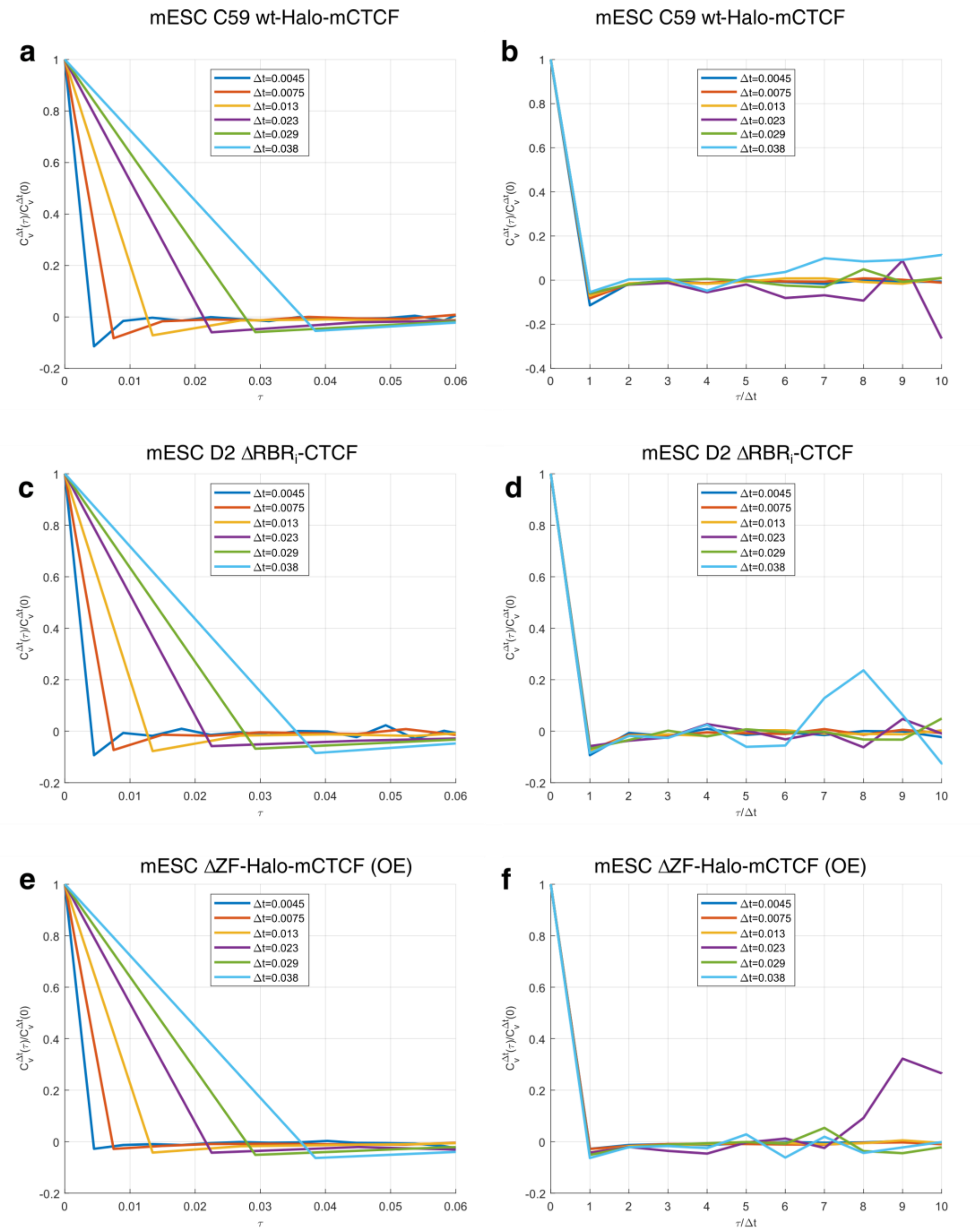
Supplementary Figure 7. MSD from model simulations.

The MSD was estimated form the Brownian simulation, using the same algorithm described at https://gitlab.com/anders.sejr.hansen/anisotropy and in **Supplementary Fig. 1**.

(**a-b**) The MSD and anomalous exponent correspond to the data plotted in **Fig. 2e-f**. (**a**) Model where the trapping probability at the TTZ is high $P_{trap} = 0.99$. (**b**) The trapping probability at the TTZ is low $P_{trap} = 0.1$.

(**c-e**) The MSD and anomalous exponent correspond to the data shown for the data plotted in **Fig. 2d-h**. (**c**) Representing wt with binding probability $P_{trap,TTZ} = 0.99$, $P_{trap,PTZ} = 1$. (**d**) representing ∆RBRi-CTCF $P_{trap,TTZ} = 0.2$, $P_{trap,PTZ} = 1$ (**e**) representing ∆ZF-CTCF $P_{trap,TTZ} = 0.01$, $P_{trap,PTZ} = 0.01$.

Error bars show standard deviation (from subsampling 50% of the simulated data) and the number of timepoints was limited to 20, 18, and 16 for 223 Hz, 134 Hz, and 74 Hz, respectively. The centre values, show value using 100% of the simulated data from n=1 simulation. The data shown in the figure in each condition is the result of n=30 independent simulations. In each simulation, the trajectory of CTCF was recoded until its 10000's capture event.

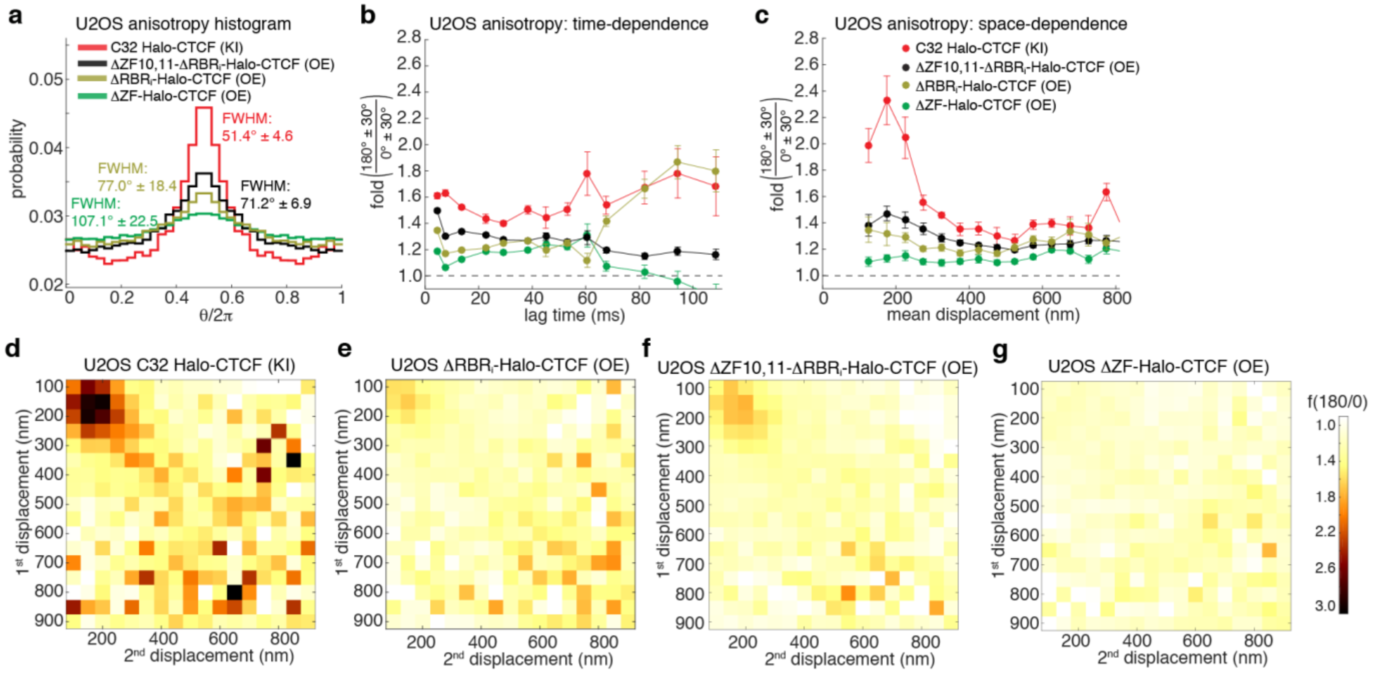


Supplementary Figure 8. The normalized velocity autocorrelation function for the CTCF experiments.

(a) The normalized velocity autocorrelation function $C \equiv C_v^{\Delta t}(\tau)/C_v^{\Delta t}(0)$ from Ref [9,10] computed at different time steps as a function of time for mESC C59 wt-Halo-mCTCF.

(**b**) The time-scaled normalized velocity autocorrelation function for the same cell line.

(**c**,**d**) mESC D2 ΔRBRi-CTCF (**e,f**) mESC ΔZF-Halo-mCTCF (OE).



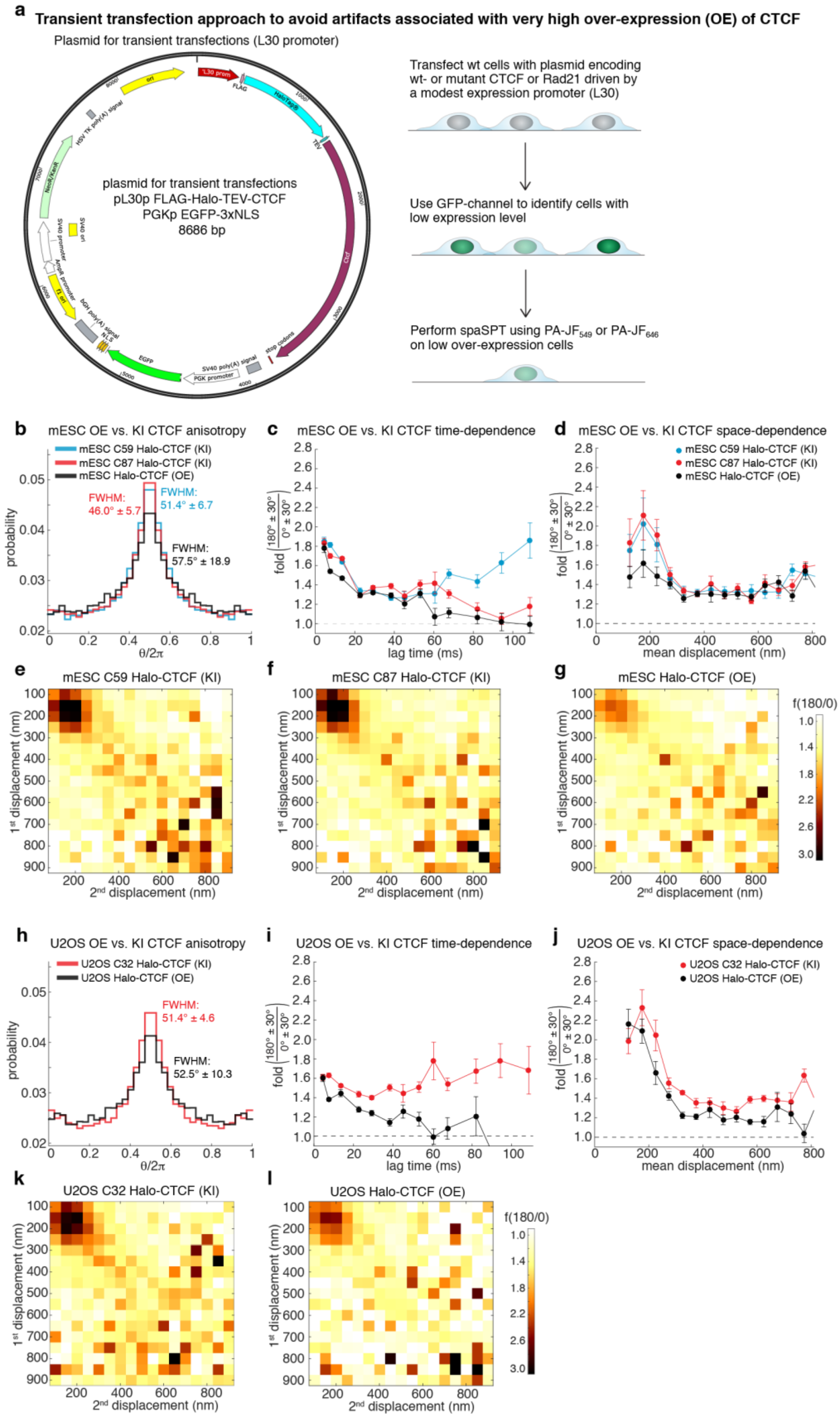
Supplementary Figure 9. Anisotropy of wild-type and mutant CTCF in human U2OS cells.

Anisotropy plots for C32 Halo-CTCF (knock-in), Halo-CTCF without Zinc Finger 10 and 11 and the RBR_i (transiently over-expressed; ΔZF10,11-ΔRBR_i-Halo-CTCF), ΔRBR_i-Halo-CTCF (transiently over-expressed) and Halo-CTCF without all 11 Zinc Fingers (transiently over-expressed; ΔZF-Halo-CTCF) in human U2OS cells showing fold-anisotropy, $f_{180/0}$, at the bulk level (**a**), as a function of the lag time (**b**) and as a function of the mean displacement length (**c**).

Error bars in plots above show standard deviation from 50 subsamplings with replacement using 50% of the data and centre values show value using 100% of the data.

(**d-g**) Anisotropy heatmaps. Plot of $f_{180/0}$ as a function of the length of the first and second displacement for the indicated cell lines.

Here KI refers to endogenous knock-in cell lines and OE refers to exogenous over-expression using an optimized protocol as outlined in **Supplementary Fig. 10**.



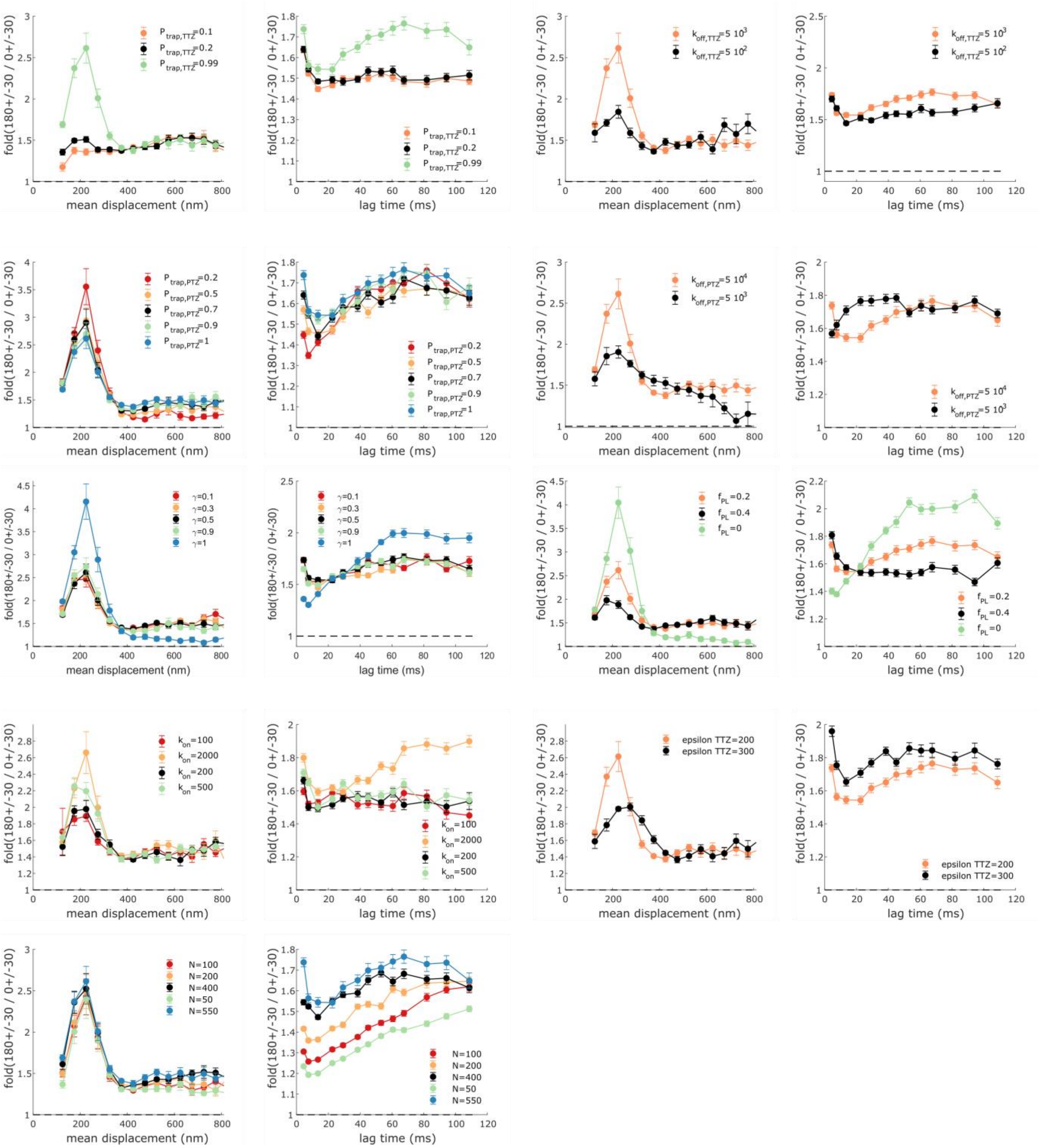
Supplementary Figure 10. An optimized protocol for minimizing artifacts associated with protein over-expression and comparison of knock-in (KI) and over-expression (OE) results.

(**a**) Overview of transient transfection approach to minimize artifacts associated with very high over-expression of CTCF. As previously described, strong exogenous over-expression of CTCF causes cell-cycle arrest and altering of CTCF dynamics[3,11]. Notably, the bound fraction of CTCF is greatly reduced when strongly over-expressed[3]. We therefore optimized a protocol to minimize the artifacts associated with CTCF over-expression, since certain likely lethal mutants of CTCF can only be studied under exogenous over-expression (OE) conditions. We use a robust, but modest expression promoter (L30) to express CTCF and related proteins and mutants. But the plasmid also encodes a GFP-3xNLS protein. We can then use the GFP channel to identify cells with the lowest level of expression and then only study those using spaSPT in the PA-JF$_{549}$ or PA-JF$_{646}$ channels. We found that unlike when using strong promoters like CMV or EF1a, this approach only led to a modest reduction in the wt-CTCF bound fraction (of ~10-15%) even when over-expressed exogenously in the presence of endogenous CTCF.

(**b-g**) Anisotropy analyses for endogenous Halo-wt-CTCF (C59 and C87) in mESCs and for exogenously over-expressed Halo-wt-CTCF using the protocol outlined in (**a**).

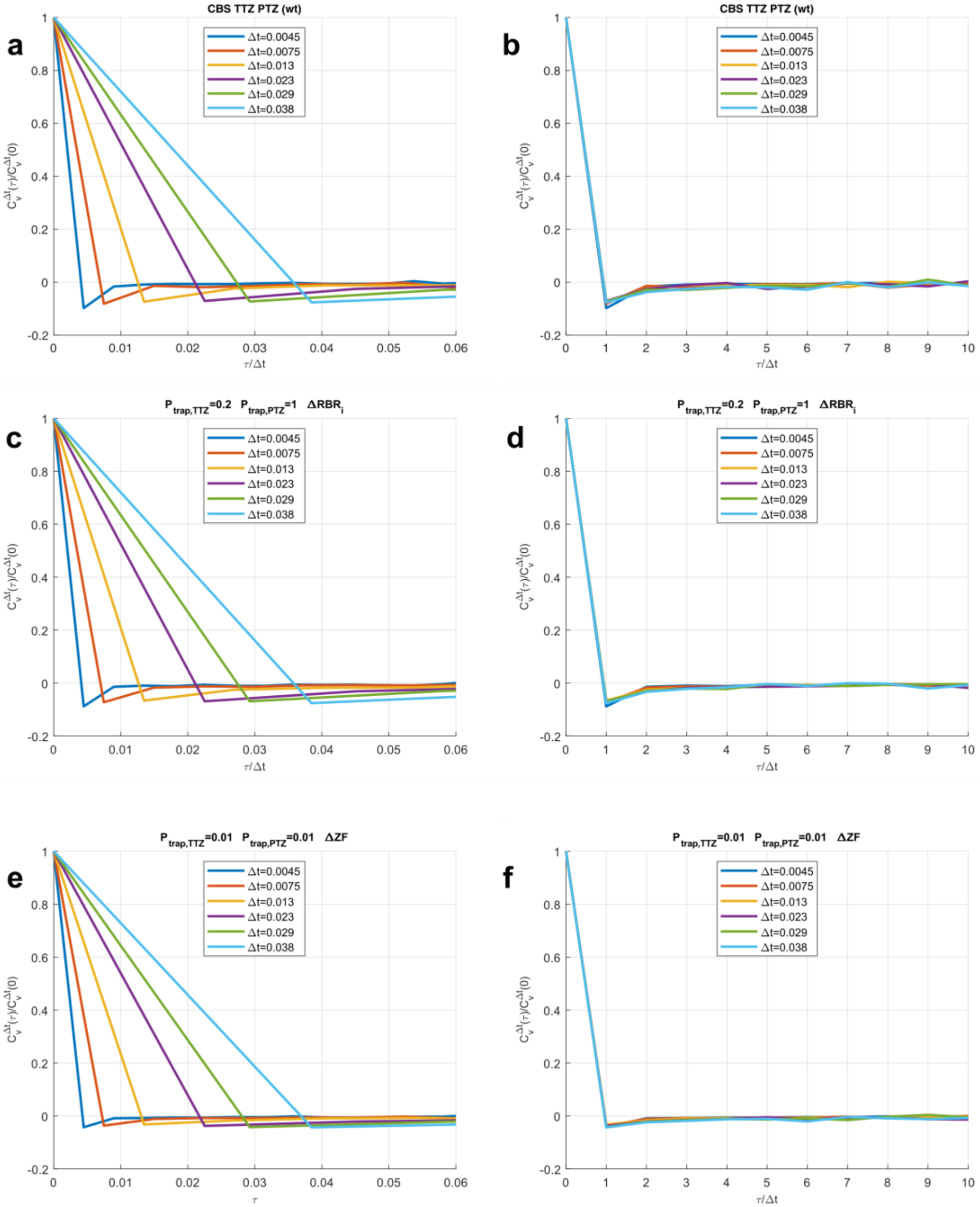
(**h-l**) Anisotropy analyses for endogenous Halo-wt-CTCF (C32) in U2OS cells and for exogenously over-expressed Halo-wt-CTCF using the protocol outlined in (**a**).

Error bars in plots above show standard deviation from 50 subsamplings with replacement using 50% of the data and centre values show value using 100% of the data and experiments were performed in at least n=3 biological replicates.



Supplementary Figure 11. Parameter scan. Plot of $f_{180/0}$ vs. mean displacement length averaging over all lag times or vs. lag time averaging over all displacement lengths for different model parameters. In each curve only one parameter is modified. The value of the other parameter is the same as the model simulation wt-CTCF. See **Fig. 2e-h**, red curve and **Supplementary Table 2**. The data shown in the figure in each condition is the result of n=30 independent simulations. In each simulation, the trajectory of CTCF was recoded until its 10000's capture event.

Error bars in plots above show standard deviation from 50 subsamplings with replacement using 50% of the data and centre values show value using 100% of the simulated data based on n=30 simulations.



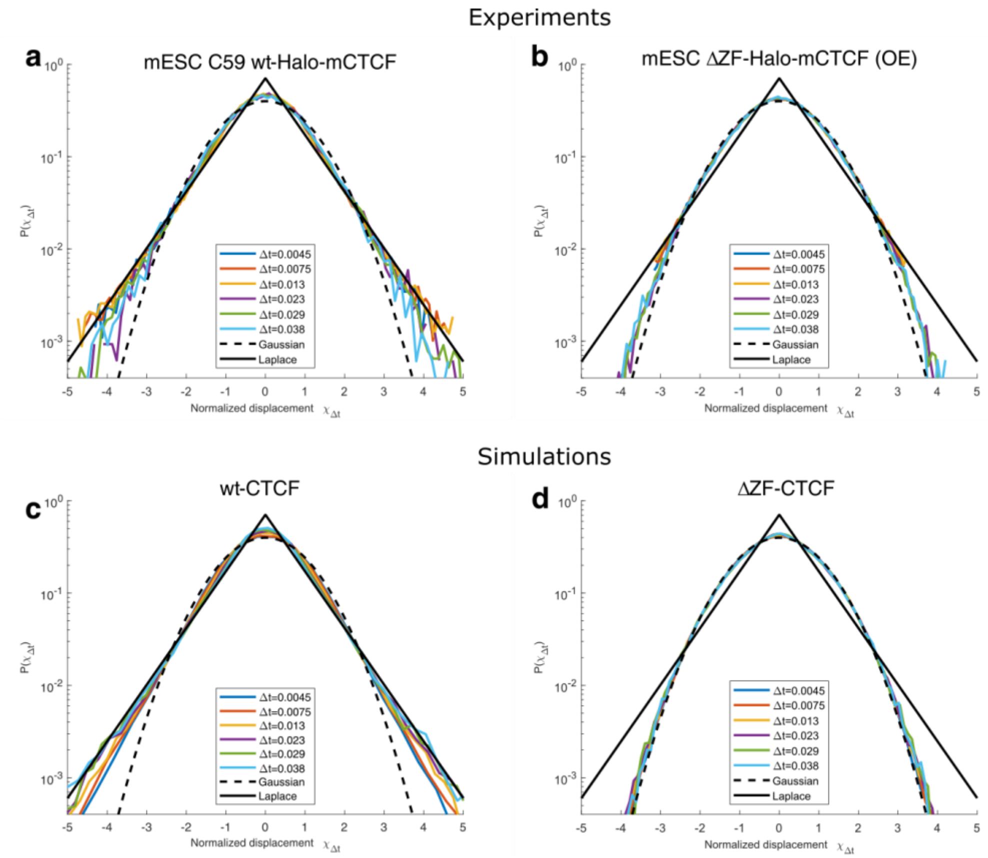
Supplementary Figure 12. The normalized velocity autocorrelation function for the CTCF simulations.

(a) The normalized velocity autocorrelation function $C \equiv C_v^{\Delta t}(\tau)/C_v^{\Delta t}(0)$ from Ref [9,10] computed at different time steps as a function of time for the model corresponding to wt-CTCF.

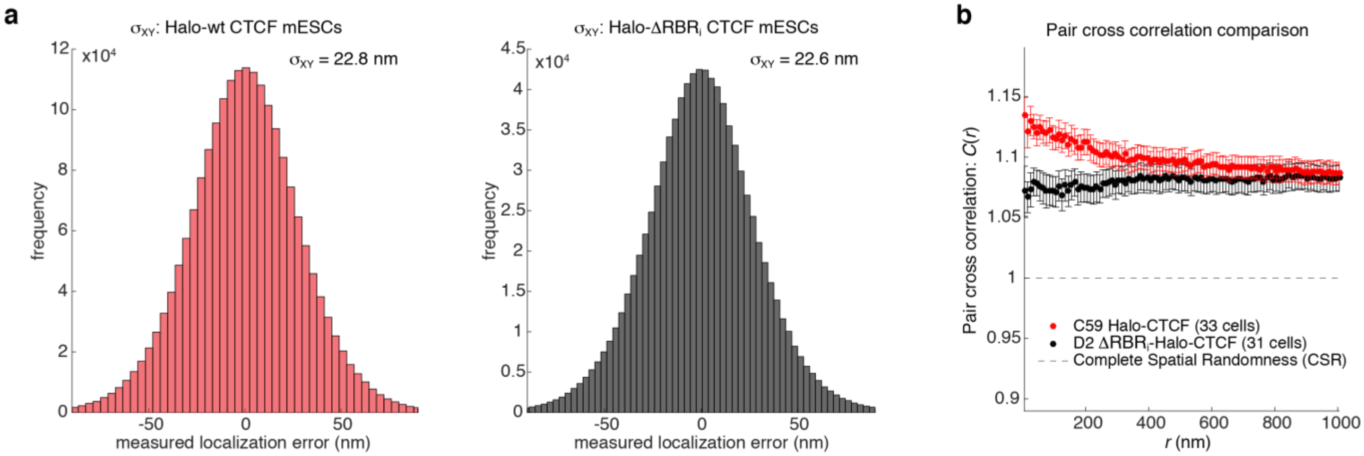
(**b**) The time-scaled normalized velocity autocorrelation function for the same cell line.

(**c**,**d**) model corresponding to ΔRBR$_i$-CTCF

(**e,f**) model corresponding to ΔZF-Halo-CTCF. See Fig. 3g-h.



Supplementary Figure 13. The displacement probability distribution function.
The displacement probability distribution function (pdf) for both the experimental data and model data, normalized by the standard deviation of each curve for different time steps. Also plotted is the Gaussian distribution (with mean zero and variance of one) (dashed line), and a Laplace distribution (full line) with the same moments for comparison. (**a**) mESC C59 wt-Halo-mCTCF (KI). (**b**) mESC ΔZF-Halo-mCTCF (OE). (**c**) The model corresponding to wt-CTCF. (**d**) The model corresponding to ΔZF-CTCF. See **Fig. 3g-h**.

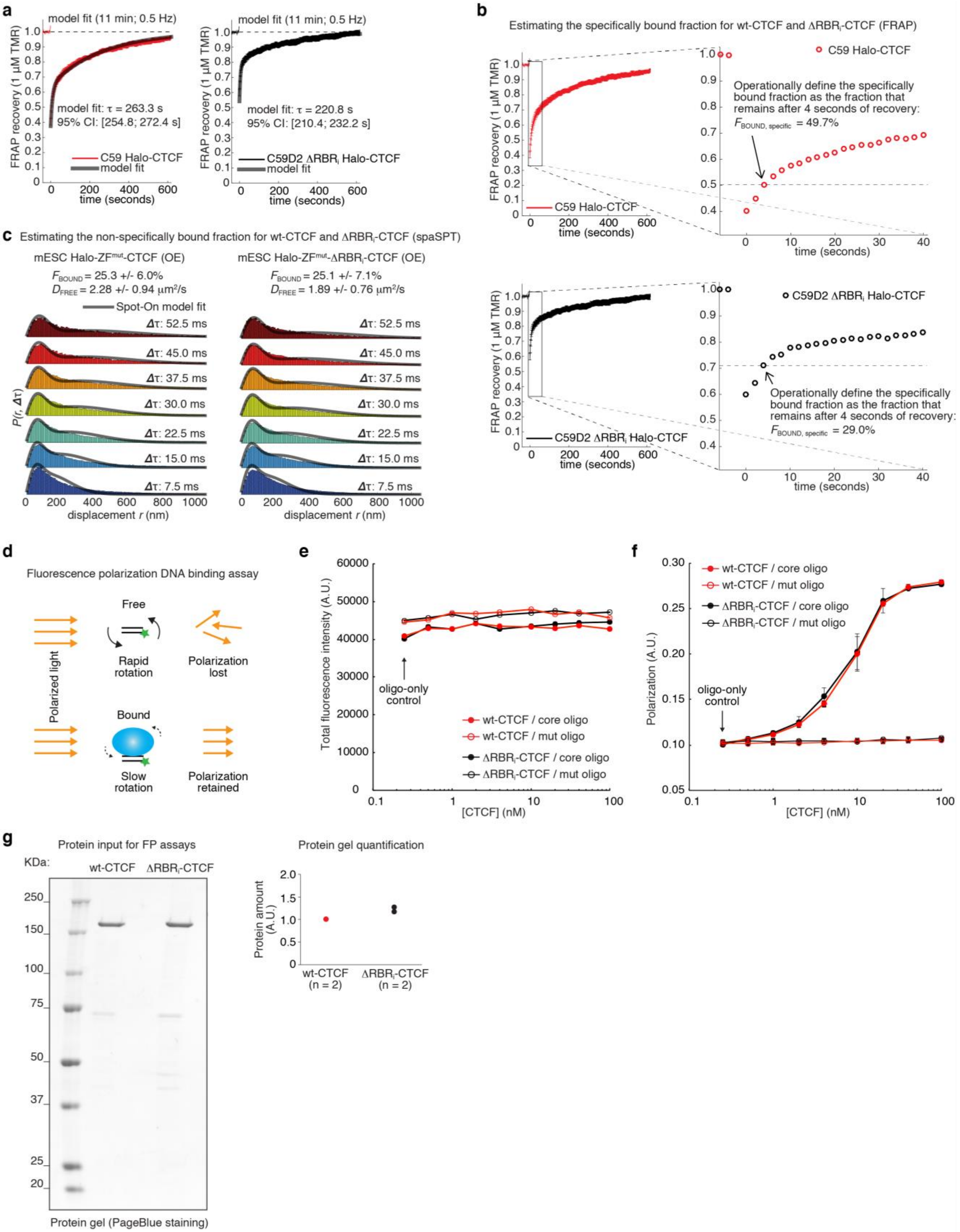


Supplementary Figure 14. Plots related to 2-color spaSPT and PALM.

(**a**) Measured localization errors (defined at the standard deviation) for mESC C59 wt-Halo-CTCF and mESC C59D2 ΔRBR$_i$-Halo-CTCF. For bound trajectories lasting at least 20 frames, we calculated the mean and then took the difference in x and y from this mean as the measured localization error. The histograms show all of these measured localization errors across n=3 replicates and at least 30 single cells.

(**b**) Pair cross correlation analysis. The analysis is the same as shown in **Fig. 4e**, except we are comparing all of the PALM localizations to the anisotropic spaSPT localizations.

Errors bars shown standard error of the mean based on a total of 33 cells for C59 and 31 cells for D2 from n = 3 biologically independent replicates. Centre values show the mean.



Supplementary Figure 15. Model-fit to FRAP data and supplementary spaSPT and *in vitro* binding experiments.

(**a**) Raw FRAP data (330 frames at 1 frame every 2 seconds and bleach applied at frame 20) for mESC C59 Halo-CTCF (left) and mESC C59D2 ΔRBR$_i$-Halo-CTCF (right). A 2-state reaction dominant model was fit and the slowcomponent interpreted as the residence time for binding to cognate DNA binding sites (shown as well as 95% confidence interval).

(**b**) Estimating the specifically bound fraction from FRAP. As a simplified form of analysis, we operationally defined the bleached fraction that remains after 4 seconds as the specifically bound fraction (assuming that the diffusing fraction and non-specifically bound fraction has largely recovered at this point). We used this to estimate the speed-up of the search process by the RBR$_i$. For full details, please see the Methods.

(**c**) Estimating the non-specifically bound fraction from spaSPT. We operationally define the bound fraction that remains after mutating the 11 Zinc Fingers (point mutation in each; His→Arg) as the non-specifically bound fraction and subtract this from the total bound fraction, in order to estimate the specifically bound fraction and the search times. For full details, please see the Methods.

(**d**) Schematic representation of fluorescence polarization-based *in vitro* DNA binding assays. A labeled DNA duplex freely rotates in solution in the absence of protein. When excited by polarized light, the emitted light is mainly depolarized due to the small size and rapid rotation of the oligo. Upon binding to a specific protein, the mass of the complex slows the rotation so that the emitted light is still polarized. The residual polarization is thus an indirect measurement of the amount of protein bound to the oligo.

(**e**) Total fluorescence intensity of oligo-only control samples (CTCF core binding site or mutant oligo) and of samples containing an increasing amount of either wt or mutant CTCF recombinant protein (0.5,1, 2, 4, 10, 20, 40 or 100 nM). Total fluorescence remains within the error range (mean of 2 replicates), with no evident change after protein addition.

(**f**) Polarization values used to calculate the fraction bound of **Fig. 5e**. No significant binding is measured in the presence of the mutant oligo, while wild type and mutant CTCF proteins bind indistinguishably to the specific core DNA oligo (mean and standard deviation of 3 independent experiments performed with the same batch of oligos and recombinant proteins).

(**g**) Protein inputs for the binding assays. Left: a representative PageBlue-stained SDS-PAGE gel of recombinant wt and mutant CTCF; Right: quantification of band intensities for 2 replicates (relative to wt-CTCF).

Supplementary Video Legends

Video 1. Single Halo-CTCF protein exhibiting anomalous diffusion inside mESC nucleus.
Frame-rate: 134 Hz; dye: PA-JF$_{549}$; laser excitation pulse: 1 ms of 561 nm; Cell line: C59. Raw microscopy data with tracking overlaid in red. Representative single trajectory from 311,144 total trajectories collected from 28 single cells across n=4 independent biological replicates.

Video 2. Single Halo-CTCF protein exhibiting anomalous diffusion inside mESC nucleus.
Frame-rate: 134 Hz; dye: PA-JF$_{549}$; laser excitation pulse: 1 ms of 561 nm; Cell line: C59. Raw microscopy data with tracking overlaid in red. Representative single trajectory from 311,144 total trajectories collected from 28 single cells across n=4 independent biological replicates.

Video 3. Single Halo-CTCF protein exhibiting anomalous diffusion inside mESC nucleus.
Frame-rate: 134 Hz; dye: PA-JF$_{549}$; laser excitation pulse: 1 ms of 561 nm; Cell line: C59. Raw microscopy data with tracking overlaid in red. Representative single trajectory from 311,144 total trajectories collected from 28 single cells across n=4 independent biological replicates.

Video 4. Single Halo-CTCF protein exhibiting anomalous diffusion inside mESC nucleus.
Frame-rate: 134 Hz; dye: PA-JF$_{549}$; laser excitation pulse: 1 ms of 561 nm; Cell line: C59. Raw microscopy data with tracking overlaid in red. Representative single trajectory from 311,144 total trajectories collected from 28 single cells across n=4 independent biological replicates.

Supplementary Notes

Supplementary Note 1 - Data processing and anisotropy and MSD calculations

Full details of the data processing and how the calculations were performed including the raw code are available on GitLab: https://gitlab.com/anders.sejr.hansen/anisotropy. Here we will describe each step briefly. We wrote all the code in Matlab and ran it in Matlab 2014b.

First, we converted raw images into SPT trajectories using the MTT-algorithm[12] as described above.

Second, we use "MergeQC_SPT_data.m" to merge data. When performing tracking and localization, each single cell results in a single file. Thus, to keep things manageable, we use "MergeQC_SPT_data.m" to merge data from all these single cells (~20-30 cells) into a single file for a given frame rate, by concatenating the frames (e.g. if two movies with 20,000 frames are merged, frame 1 in the second movie becomes frame 20,001). To minimize tracking errors, we also use "ClosestDist = 2" (in units of μm) to abort trajectories where two particles came closer than 2 μm to each other. This is achieved by calling the function "RemoveAmbigiousTracks.m".

Third, we use "Batch_vbSPT_classify.m" to classify trajectory segments into "bound" and "free" segments. "Batch_vbSPT_classify.m" first removes gaps from all the trajectories and then calls vbSPT[13]. We implemented a 2-state Hidden Markov Model (HMM) through vbSPT, which uses a Bayesian approach to infer the most likely state for each displacement in a trajectory ("bound" or "free") based on displacement lengths and trajectory history. This is important because the apparent movement of bound molecules is dominated by localization errors. Thus, we want to exclude bound/immobile molecules from the analysis since we are interested in understanding the nuclear search mechanism, which only applies to diffusing/free molecules and also because bound/immobile molecules will artefactual appear anisotropic due to localization errors around a relatively fixed position. Thus, by using a 2-state HMM we can filter out the bound population and restrict our subsequent analysis to the free population (the number of states is controlled by the variable "maxHidden". "Batch_vbSPT_classify.m" will automatically run on all SPT datasets in the directory "input_path", reformat data to remove gaps and save the reformatted data to "path_reformatted" and finally save the classified data to "path_classified". The final classified SPT data contains 4 variables:

- "CellTracks", a cell array where each element is a trajectory and is a Nx2 matrix with the XY coordinates for each of the N frames.
- "CellTrackViterbiClass", a cell array where each element is a N-1 column vector corresponding to the relevant trajectory in "CellTracks". In other words, "CellTrackViterbiClass" classifies each displacement and thus is one length shorter than the number of localizations in "CellTracks". "1" corresponds to bound and "2" corresponds to free. E.g. if a trajectory had 5 localizations, "CellTrackViterbiClass" will have length 4.
- "vbSPT_metadata", a structure array object containing the most relevant vbSPT metadata such as the inferred diffusion coefficients, subpopulation sizes and transition matrix.
- "LagTime", the time between frames in units of seconds.

"Batch_vbSPT_classify.m" calls two dependent functions: "InferFrameRateFromName.m", which infers the frame rate from the filename and "EditRunInputFile_for_batch.m" which edits the file "vbSPT_RunInputFileBatch.m" to automatically feed the relevant information to vbSPT. In summary, at the end of this step the trajectories have been classified to allow subsequent analysis to focus exclusively on the free/diffusing population.

Fourth, we use "CompileTemporalSubSamplesOfHMM.m" to temporally subsample the existing SPT data and generate trajectories with longer lag times. E.g. by subsampling every 10th frame (frames 1, 11, 21, …) of the 223 Hz data and every 6th frame (frames 1, 7, 13, …) of the 133 Hz data, we can generate new SPT trajectories at 22.2 Hz. Full details on how the 223 Hz, 133 Hz and 74 Hz SPT data was temporally subsampled is given in the structure array, "TempSubSampleStruc" in lines 40-117. We use this approach to generate SPT data at the following frame rates: 44.4 Hz, 34.2 Hz, 26 Hz, 22.2 Hz, 18.8 Hz, 16.5 Hz, 14.8 Hz, 12.2 Hz, 10.6 Hz, 9.2 Hz. The subsampling is performed in the dependent function "TemporallyReSampleCellTracks.m" and we note one potential ambiguity here. To illustrate, suppose we want to take every 3rd frame of a trajectory (i.e. frames 1, 4, 7, …). Since the SPT data has already been HMM-classified, a trajectory of length N will have N-1 displacements classified as either bound or free. We would like to carry over this classification to the temporally subsampled trajectory. While most trajectories are either entirely free (“2”) or bound (“1”), some trajectories show transitions. In this example, say the HMM-classification is [1,2,2,2,2,1] for frames 1-7. In this case, the subsampled displacement from frame 1-to-4 will have HMM-classification [1,2,2], but we have to label it as either “1” or “2’ in the sub-sampled data. In these cases, we took the most conservative approach. Since our primary goal is to filter out the bound population, we labelled any temporally subsampled displacement as bound as long as any one of the intermediate displacements were classified as bound, even if the majority were free. This is implemented in the function “TemporallyReSampleCellTracks.m”. Finally, at the end of this procedure, all the temporally subsampled and HMM-classified SPT datasets are saved to the directory “HMM_first_QC_data”.

Fifth, we wrote “Process_SpatioTemporal_AngleAnalysis_v2.m” to perform all the analysis of the HMM-classified trajectory data at multiple spatial and temporal scales. The bulk of the analysis is performed in the dependent function “angleFWHM_Amp_HMM_analyzer_v5.m” and the code also calls “AngleMatrix_analyzer.m” and “ComputeAmpFWHM.m”. The analysis is somewhat complicated and for full details, we refer the reader to the underlying function “angleFWHM_Amp_HMM_analyzer_v5.m”. But briefly, we describe the analysis, input parameters and output results here. For a given trajectory, as long as it consists of at least 3 localizations and thus 2 displacements, it is possible to calculate an angle between adjacent displacements. If we define the 3 localizations making up the angle as p1, p2 and p3, we can define 2 column vectors and calculate the angle between them as follows (using Matlab syntax):

```
>> v1 = (p2-p1)';
>> v2 = (p3-p2)';
>> angle(1,1) = abs(atan2(det([v1,v2]),dot(v1,v2)));
>> angle(1,2) =  2*pi-angle(1,1);
```

Here, the second element in the angle is due to the factor that whether a 150 degree angle is classified as 150 or 210 degrees is arbitrary. Note though that the above calculation is in units of radians. We then loop over all the trajectories. However, we want to be careful to filter out the bound population and we therefore apply two criteria. First, the displacements must be HMM-classified as “free” and, second, both displacements must be at least of length “MinMinJumpThres”. Only if both criteria are satisfied, do we count the angle. These criteria are used for the bulk analysis of the angles (subplots 1-6 in the plotting step). Afterwards, the analysis quantifies 4 different anisotropy metrics:

- AC: anisotropy coefficient which is defined as: AC = $\log_2$(P(a∈[150°-210°])/ P(a∈[330°-30°])), and this metric was introduced previously by Izeddin *et al.*[14] Thus, the AC quantifies how much more likely a molecule is to go in the backwards after having going forwards.
- Amp: how the amplitude is calculated is described in the function “ComputeAmpFWHM.m”. Briefly, we build a histogram of the probability as a function of the binned angle. We then use fine-scale interpolation to overcome the binning. From this we fit the background density and then calculate the “excess” anisotropy. Since the histogram sums to 1, the amplitude takes values between 0 and 1. This provides a related but somewhat orthogonal metric or anisotropy.
- f(180°±30°/0°±30°) or f(180/0) for short: is identical to the AC, but does not use a logarithm. Essentially, f(180/0) quantifies how many times more likely a particle is to go backwards relative to continuing forwards. For example, if f(180/0)=1.6, the particle is 1.6-fold more likely to return in the backwards direction than to continue forwards.
- FWHM (full width at half-maximum): quantifies the width of the anisotropy histogram and is defined as the full width of the histogram peak centered around 180° at the half-maximal value (i.e. when the probability is half-way in between the peak of the of the angle histogram and the background probability).

This analysis quantifies the “bulk” anisotropy.

To analyze how the anisotropy changes with space and time, we do binning. For spatial analysis, i.e. how the anisotropy metrics depend on the length of the two displacements making up the angle, we control the range of translocations in the variable “MovingThreshold”, which we allow to run from 100 nm to 950 nm in bins of 50 nm. Thus, in this case we additionally consider displacements that are HMM-classified as free and also which are at least 100 nm long (set by the variable "GlotbalMinJumpThres"). Since there are two displacements making up the angle, one can either quantify their mean displacement, their minimal displacement (univariate analysis) or the length of both displacements (bivariate analysis), and we took all three approaches here. For spatial analysis, we averaged over all 13 time-scales (from 223 Hz to 9.2 Hz) and calculated the anisotropy metrics. For example, for f(180/0) as a function of mean displacement, we populated the [100 nm ; 150 nm] bin by taking all angles from all 13 frame rates where the mean of the two displacements making up the angle is between 100 and 150 nm and then calculated f(180/0) using all of these angles. And likewise for the other metrics. For analysis of anisotropy as a function of time, we use all the angles making up a single frame rate satisfying the both of the two criteria listed above: HMM-classified as "free" both displacements at length "MinMinJumpThres" (200 nm).

Errorbars were estimated using re-sampling and is controlled by the variables "JackKnife_fraction" and "JackKnife_iterations". Specifically, we resampled 50% of the angles with replacement 50 times and calculated the anisotropy metrics for each iteration. The error bars reported here show the standard deviation between these 50 iterations.

We noticed that at very long displacements, we would occasionally see very strange trajectories such as a particle shifting back and forth between 2 points separated by a large distance (~800 nm). These are almost certainly a tracking artifact, perhaps from 2 bound molecules blinking out-of-frequency. To remove this and avoid these biasing the analysis especially at long displacements, we removed trajectories where more than half of the angles were highly anisotropic using the variables "MaxNumAngles" and "MaxAsymAnglesFrac".

Finally, all of the anisotropy metrics are saved to the structured array "FinalResults" and saved. With the example data provided on GitLab, the workspace saved is "U2OS_C32_SpatioTemporalAnalysis.mat".

Sixth and finally, we used "PLOT_SpatioTemporalAnalysisResults_v4.m" to plot and visualize the results for each individual data set and similar code to overlay results from multiple different samples. For a more comprehensive description of the analysis and the underlying source code, please see GitLab: https://gitlab.com/anders.sejr.hansen/anisotropy

MSD

Full details and all the raw code used for calculating the time- and ensemble-averaged MSD is available on GitLab https://gitlab.com/anders.sejr.hansen/anisotropy. Here we provide a brief overview. We wrote all the code in Matlab and tested it in Matlab 2014b. The mean-squared displacement (MSD) is a classic analysis approach for SPT. Briefly, it involves analyzing displacements at increasing time-lags and then plotting the time- and ensemble-averaged mean squared displacement with time. If the particle exhibits normal Brownian motion, the MSD should be linear with time according to:

$$MSD(\tau) = 4D\tau^{\alpha}$$

Accordingly, α=1 for Brownian motion. α<1 indicates subdiffusion (anomalous diffusion). Here, since our trajectories are generally too short to analyze individually[15], we therefore compute the time- and ensemble-averaged MSD. The MSD at a timelag τ is given by:

$$MSD_i(\tau) = \left(r_i(t+\tau) - r_i(t)\right)$$

Thus, if we average over all displacements with time-lag τ and over all trajectories, *i*, we obtain the time- and ensemble-averaged MSD for a given timelag τ. In the case of nuclear proteins that bind chromatin and which are tracked with a significant localization error (around 35 nm in our data), the calculations are substantially more complicated. For example, the bound molecules exhibit much longer trajectories than the freely diffusing population, since the free population rapidly moves out-of-focus[2]. Thus, at longer time-lags, unless corrected for, the bound population would dominate the MSD calculations. Moreover, MSDs should not be calculated and averaged over a mixture of two distinct populations. Thus, to filter out the bound molecules, we use the HMM-classification described above. We use the function "MSD_HMM_analyzer.m" to calculate the MSD using only the segments that are classified as free using the HMM. Since we have data at three different frame rates, we calculate the MSD for each frame rate individually (~223 Hz, 134 Hz, 74 Hz) and we also merge the data from all three frame rates.

When it comes to model-fitting, we fit to the HMM-classified MSD. But we must also account for localization errors and we therefore consider the expression:

$$MSD_{\mathrm{HMM}}(\tau) = 4D\tau^{\alpha} + 4\sigma^2$$

where σ is the localization error (standard deviation in one dimension; approximate 35 nm as determined using Spot-On[2]). In terms of model-fitting the time- and ensemble-averaged MSD, there are a number of considerations. First, how long time-lags to consider for calculating the MSD. Second, what fraction of the data to use for the fitting. To answer the first question, we used subsampling of the data: for each time-lag, we subsampled 50% of the data using 25 iterations and calculated the standard deviation (as shown by the error bars). We then limited the number of timepoints to use (using the vector "Conditions.MSD_timepoints") to 20, 18 and 16 for 223 Hz, 134 Hz and 74 Hz, respectively, such that we did not consider MSD values where the error bars got very large. In terms of least-squares fitting of the data, Saxton has argued that one should limit the fitting to only a fraction of the MSD curve[5,16]. Thus, here we only use the first 50% of the data in the fitting and we perform least-squares fitting as shown in "ProcessPlotFit_MSD.m".

We note that fitting the MSD was not very robust in this case. We see substantially variation in the inferred parameters between the 3 frame rates and it has been argued elsewhere that MSD-analysis is not an optimal method for analyzing SPT data[6]. We also note that MSD analysis is much less robust for inferring diffusion coefficients[2]. Nevertheless, since it is a standard method for determined the anomalous exponent, α, we include it here. We took the final inferred α as the one fitted when averaged over data from all three frame rates and we also calculated the 95% confidence interval. Full details and raw code to reproduce our analysis is given on Gitlab: https://gitlab.com/anders.sejr.hansen/anisotropy.

Supplementary Note 2 - Estimating the contribution of tracking errors (misconnections)

Single-particle tracking (SPT) is uniquely suited to analyzing the dynamics of proteins, how they explore the nucleus and whether they exhibit anomalous diffusion since it allows direct observation of molecular motion at high spatiotemporal resolution. However, experimental SPT is also subject to numerous biases, which complicates the interpretation of SPT experiments. First, while a frame is recorded, fast-diffusing molecules can move several pixels, which causes a "motion-blur" artifact. These molecules will not be recognized by most localization algorithm since they no longer resemble a PSF and thus fast-diffusing molecules will be undercounted. Second, most SPT experiments in live mammalian cells have used quite high particle densities (e.g. ~20-100 in-focus molecules per frame). This allows a user to rapidly and conveniently record large amounts of data, but comes at the cost of tracking errors. When the displacements of multiple particles overlap between frames, there is no way to unambiguously identify which molecule moved where. In general, the fraction of displacements that are misconnected increases with particle density. In the high-density limit, when the trajectories contain many tracking errors, it is no longer possible to make inferences about how single molecules explore the nucleus.

To minimize these biases we introduced stroboscopic photo-activation single-particle tracking (spaSPT)[2,3], which builds on and integrates previous approaches[17,18]. First, spaSPT minimizes motion-blur bias by using stroboscopic excitation[18]. Here we use 1 ms stroboscopic excitation pulse, which is sufficient to achieve very good signal (signal-to-background ratio > 5), and which we have previously shown theoretically[3] and experimentally[2,3] causes negligible motion-blur bias even for fast-diffusing proteins such as Halo-3xNLS (D_{FREE} ~ 10-12 μm^2/s). Second, spaSPT largely avoids tracking errors, by keeping the particle density very low. We take advantage of the very bright photo-activatable Janelia Fluor dyes[8], which allows us to tune the photo-activation probability such that essentially any desired particle density can be achieved. Here we aimed for an average density of ~ 0.5-1 localizations per nucleus per frame, which provides a reasonable compromise between minimizing tracking errors and also recording large SPT datasets necessary for statistical analysis.

Even though tracking errors through misconnections between 2 particles are quite rare in spaSPT at low densities, they always will happen with some probability and we wanted to quantitatively assess their frequency. To do so, we labeled Halo-tagged proteins in live U2OS cells with the two spectrally distinct dyes, PA-JF$_{646}$ and PA-JF$_{549}$. The HaloTag protein binds only one dye covalently and irreversibly and under our imaging conditions there was no spectral overlap (i.e. any fluorescence bleedthrough between the color channels was far below the detection limit). Thus, we recorded simultaneous 2-color spaSPT data using U2OS C32 Halo-CTCF and U2OS Halo-3xNLS at 133 Hz and 74 Hz. We considered both Halo-CTCF and Halo-3xNLS since they represent opposite extremes: U2OS Halo-CTCF diffuses very slowly (D_{FREE} ~ 2.0-2.5 μm^2/s) and exhibits a high bound fraction ($F_{\mathrm{BOUND,total}}$ ~ 60%), whereas Halo-3xNLS diffuses extremely rapidly (D_{FREE} ~ 10-12 μm^2/s) and exhibits a minimal bound fraction (F_{BOUND} ~ 10%). Thus, we reasoned that considering the two extremes would allow us to place upper and lower bounds and the frequency of tracking errors.

We then analyzed the data as follows: We analyzed the JF$_{549}$ and JF$_{646}$ datasets individually and we then combined the localizations for both datasets and feed them into the tracking algorithm[12] such that the tracking algorithm was blind to a particle's "color" (and keeping all algorithm parameters the same). This allowed us to identify any cases where a single-trajectory contains localizations in more than one color, which must have involved a tracking misconnection. Moreover, to generate a "worst-case scenario" dataset, we used a particle density of ~0.5-1.5 for JF$_{549}$ and JF$_{646}$ individually (so ~1-3 in-focus particles per frame in total), such that the merged localizations would exhibit a 2-3 fold higher density than normal. Thus, this analysis would generate an upper bound on the number of tracking errors.

However, not all tracking errors will give a change in particle colors. Even in the 2-color datasets, JF$_{549}$-to-JF$_{549}$ and JF$_{646}$-to-JF$_{646}$ are also possible. More generally, if we refer to the colors as "1" and "2", the fractions of all localizations in each color will be:

$$f_1 = \frac{n_1}{n_1+n_2} \text{ and } f_2 = \frac{n_2}{n_1+n_2}$$

where n is the number of localizations per color per movie. If tracking is totally arbitrary (this is a slight approximation, since the probability of either color appearing in frame k+1 is not independent of frame k due to photo-activation being less likely than photo-bleaching), the probability of tracking being observed as correct is:

$$P_{\mathrm{correct}} \approx P(c_{\mathrm{Frame\ k+1}} = 1 | c_{\mathrm{Frame\ k}} = 1) + P(c_{\mathrm{Frame\ k+1}} = 2 | c_{\mathrm{Frame\ k}} = 2)$$

and similarly the probability of actually observing a tracking error is:

$$P_{\text{wrong}} \approx P(c_{\text{Frame k+1}} = 1 | c_{\text{Frame k}} = 2) + P(c_{\text{Frame k+1}} = 2 | c_{\text{Frame k}} = 1)$$

Again, if we assume tracking to be totally arbitrary and that there is no memory between frames (which is an approximation), these reduce to:

$$P_{\text{correct}} \approx P(c_{\text{Frame k+1}} = 1 | c_{\text{Frame k}} = 1) + P(c_{\text{Frame k+1}} = 2 | c_{\text{Frame k}} = 2)$$
$$= f_1 \cdot f_1 + f_2 \cdot f_2 = f_1{}^2 + (1 - f_1)^2 = 1 + 2\left(f_1{}^2 - f_1\right)$$

And

$$P_{\text{wrong}} \approx P(c_{\text{Frame k+1}} = 1 | c_{\text{Frame k}} = 2) + P(c_{\text{Frame k+1}} = 2 | c_{\text{Frame k}} = 1)$$
$$= f_1 \cdot f_2 + f_2 \cdot f_1 = 2f_1(1 - f_1) = 2\left(f_1 - f_1{}^2\right)$$

Thus, as expected in the limit where tracking is totally arbitrary and where $f_1 = f_2$, these reduce to $P_{\text{correct}} \approx 0.5$ and $P_{\text{wrong}} \approx 0.5$. In practice, due to slight differences in cell permeability and labeling kinetics between JF$_{549}$ and JF$_{646}$ [19], as well as experimental variations such as small pipetting errors and slight differences in photo-stability etc., it is difficult to get exactly 1:1 labeling and precisely the same number of localizations for both dyes. While the exact expressions above can be used for individual movies based on the number of localizations for each dye, in order to get good statistics it is necessary to averages over many movies with slightly different fractions. Moreover, in almost all of the movies the fraction of localizations in one color was in the range 40-60%. P_{wrong} is symmetric and inverse U-shaped, and even $P_{\text{wrong}}(f_1 = 0.4) = P_{\text{wrong}}(f_1 = 0.6) = 0.48$, which is very close to 0.5, as expected if we obtained perfectly 1:1 labeling. Thus, averaging over all movies despite these having slightly different relative numbers of localizations is a very small approximation.

Having verified that we can reasonably average all of the movies, we next analyzed how the number of observed tracking errors depends on density and displacement length. First, we note that the probability of a tracking error resulting in a color change is approximately 50% assuming $f_1 \approx f_2$. That is, if we are tracking a particle in color 1, a misconnection to another particle in color 1 will not appear as a tracking error since it does not result in a color change. Only if the misconnection occurred to a particle of the opposite color, will it appear as a color change. Thus, when we count tracking errors as color changes, the actual number of tracking errors is ~2-fold higher than the observed number assuming $f_1 \approx f_2$. Thus, we multiplied the observed number of tracking errors by 2 to get the total number of tracking errors, both observed and unobserved. Somewhat encouragingly, after multiplying by 2 the fraction of incorrect connections, it was still quite low despite imaging at a 2-3 fold higher particle density than normal. Specifically, we got the following number of incorrect connections (after multiplying by 2): U2OS C32 Halo-hCTCF; 74 Hz gave 3.36%; U2OS C32 Halo-hCTCF; 133 Hz gave 1.85%; U2OS Halo-3xNLS; 74 Hz gave 6.65%; U2OS Halo-3xNLS; 133 Hz gave 4.12%.

As expected, the number of tracking errors increase when the frame rate decreases (74 Hz vs. 133 Hz) since the mean displacement increases with the lag time and also as expected, the number of tracking errors is higher for Halo-3xNLS, which diffuses very rapidly (mean displacement at 74 Hz: 637 nm) and thus is more likely to cross paths with another molecule than it is for Halo-hCTCF (mean displacement at 74 Hz: 178 nm), which diffuses much more slowly. Thus, even at 2-3 fold higher particle density, the number of tracking errors for Halo-hCTCF at 133 Hz is still <2%.

We further analyzed the 2-color spaSPT data to understand under which conditions tracking errors start appearing (**Supplementary Fig. 4a**). This analysis reveals several important points: First, the mean displacement for misconnections is much longer than for correct tracking connections (e.g. 134 nm vs. 766 nm for Halo-hCTCF at 133 Hz). Second, although the mean number of localizations per frame is generally higher when there is a misconnection compared to when the tracking was correct, it is only slightly higher (e.g. 2.05 localizations/frame for correct connections vs. 2.35 localizations/frame for misconnections for Halo-hCTCF at 133 Hz; please note that this is not the average number of localizations per frame, which is significantly lower; but since we only count displacements, e.g. frames without localizations are not counted). Third, the probability of a displacement being a misconnection (tracking error) increases *exponentially* with the displacement length (until saturation). This is perhaps surprising at first glance and to understand why this would be and to facilitate filtering out misconnections, we will distinguish two types of tracking errors: Misconnection due to Overlapping Trajectories (MOT) and Misconnection due to new Appearance of Particle (MAP) (**Supplementary Fig. 4b**).

Misconnection due to Overlapping Trajectories (MOT; **Supplementary Fig. 4b**): MOT errors will tend to occur at high densities. The higher the number of particles, the more ambiguous the tracking becomes. While many tracking algorithms[6], including the MTT-algorithm[12] used here, are more sophisticated than simply connecting nearest neighbors (for example by taking trajectory history into account), as trajectories begin to overlap, there is simply no way of unambiguously connecting molecules between frames. Accordingly, we expect the number of MOT errors to increase with: 1) increasing density of particles; 2) longer lag times (i.e. time between frames; slower frame rates); 3) larger molecular diffusion coefficient (e.g. since CTCF diffuses more slowly than Halo-3xNLS (D_{FREE} ~ 2.5 μm^2/s vs. 10-12 μm^2/s) trajectories are less likely to overlap for a given density and frame rate since the average displacement is much lower; accordingly, the fraction of misconnections is ~2-fold smaller for CTCF compared to Halo-3xNLS).

Misconnection due to new Appearance of Particle (MAP; **Supplementary Fig. 4b**): MAP errors tend to occur when a new particle appears close to an existing particle. This can either be due to a particle outside of the focal plane moving inside the focal plane (this may occur regardless of whether photo-activation is used or not) or due to spontaneous photo-activation of a particle close an existing particle. If the existing particle either bleached or moved further away, a misconnection to the newly appeared particle is then likely. MAP errors due to particles out-of-focus moving into focus are density dependent, but depend on the density of out-of-focus particles and thus will not necessarily be captured in a simple statistic like localizations/frame, which only count in-focus and detected localizations. Conversely, MAP errors due to photo-activation might be expected to be inversely dependent on density: the probability of such a MAP error occurring increases at low densities where any newly photo-activated molecule will be connected to a molecule that photobleached in the previous frame assuming that the two were sufficiently close. It is also important to note that MAP errors due to photo-activation should be expected to scale with the displacement squared. If we assume that photo-activation is equally likely to occur anywhere in the nucleus, the probability of such a MAP error to occur will depend on the area of the maximally allowed displacement ($A(r) = \pi r^2$). Thus, for this reason, we might initially expect the misconnection probability to increase with r^2 instead of exponentially as observed above. However, theory actually predicts the misconnection probability to increase exponentially with distance. To understand why, consider a case of a protein (very similar to CTCF) where roughly half of the molecules are bound to chromatin and with D_{FREE} ~ 2.5 μm^2/s. The distribution of displacements (leaving out the normalization constant) is then given by:

$$P(r) = F_{\mathrm{BOUND}} \frac{r}{2(D_{\mathrm{BOUND}}\Delta\tau + \sigma^2)} e^{-\frac{r^2}{4(D_{\mathrm{BOUND}}\Delta\tau+\sigma^2)}} + (1 - F_{\mathrm{BOUND}}) \frac{r}{2(D_{\mathrm{FREE}}\Delta\tau + \sigma^2)} e^{-\frac{r^2}{4(D_{\mathrm{FREE}}\Delta\tau+\sigma^2)}}$$

Here, F_{BOUND} is the fraction of molecules that are bound to chromatin, D_{BOUND} is diffusion coefficient of chromatin bound molecules, D_{FREE} is diffusion coefficient of freely diffusing molecules, *r* is the displacement length, $\Delta\tau$ is lag time between frames and σ is localization error. If we assume that MAP errors contribute to a 5% misconnection probability at 500 nm as seen in the raw 2-color Halo-hCTCF data at 134 Hz such that $A(r = 500\ \mathrm{nm}) = 0.05;\ P(r = 500\ \mathrm{nm})$, then we get the results shown in **Supplementary Fig. 4c**. On the top plot, we see that at long displacements, MAP errors (red) become more likely than true displacements (green). If we now plot the fraction of displacements that are incorrect as a fraction of all displacements (right plot), we see precisely the same trend as in the experimental data: We see a steep exponential increase at lower displacements (up until ~100-150 nm), followed by a kink and then a slower exponential increase from around ~150 nm to ~700 nm, where saturation begins to happen. The misconnection probability increases exponentially because the correct connection probability decreases exponentially and thus the exponential trend is due to $P(r)$ and not $A(r)$. Of course, where the transitions occur depend on the details of the parameters and this model involves several approximations, but it is notable that such a simple theoretical model can faithfully capture observed experimental trends. Most importantly, it highlights that tracking errors increase exponentially with displacement length.

The above analysis revealed MOT errors to be highly density dependent and MAP errors somewhat density dependent. Consistently, misconnections occur at, on average, slightly higher particle densities (**Supplementary Fig. 4a**). We note that although the above 2-color experiments were performed at a 2-3-fold higher particle density than the 1-color experiments and thus represent a worst-case scenario, occasionally frames with high densities are unavoidable. Since photo-activation and photo-bleaching are both Poisson processes, we would expect the distribution of localizations per frame to roughly follow a Poisson distribution. For example, if the mean density is 1 localization/frame, 26.4% of frames will nevertheless have 2 or more localizations. Likewise, even at a very low mean density of 0.5 localizations/frame, 9.0% of frames will have 2 or more localizations. Thus, a simple approach to minimizing MOT errors is to abort and remove all parts in frames n+1 and beyond, if two trajectories get closer than a certain threshold distance, $r_{\text{threshold}}$. To determine how to choose $r_{\text{threshold}}$ we plotted the probability of misconnections as a function of $r_{\text{threshold}}$ (**Supplementary Fig. 4d**). A couple of points stand out: First, as expected, restricting the closest allowable distance between particles in the same frame reduces the number of tracking errors, i.e. the probability of a misconnection. Second, the strategy is effective both for Halo-CTCF (slow diffusion, overwhelmingly bound) and Halo-3xNLS (very fast diffusion, overwhelmingly free). E.g. setting $r_{\text{threshold}} = 2\ \mu\text{m}$, reduces the misconnection probability from 6.77% to 4.55% for Halo-3xNLS at 74 Hz but also from 1.85% to 0.94% for Halo-CTCF at 133 Hz. Third, in all four cases diminishing returns are observed and eventually the curve plateaus. This happens 1.5 μm for Halo-CTCF at 133 Hz and around 2 μm in the other three cases. Most likely, the plateau occurs because this approach can filter out MOT, but not most MAP, errors. Thus, it is likely that the plateau is largely composed of MAP errors.

Importantly, filtering out data in this way also filters out some correctly tracked particles and there is always the risk of introducing unintended biases. Therefore, using $r_{\text{threshold}} = 2\ \mu\text{m}$ seemed like the optimal choice and we therefore applied this to all the datasets ("ClosestDist" variable in MergeQC_SPT_data.m).

Analyzing anisotropy at multiple spatiotemporal scales has proven very informative in distinguishing different mechanistic models of anomalous diffusion[14]. Thus, we would like to determine how far out we can extend the analysis in space (i.e. displacement length) for a given time (i.e. frame rate), while maintaining a relatively low probability of misconnections. We therefore applied the $r_{\text{threshold}} = 2\ \mu\text{m}$ to the raw 2-color dataset and then sampled the trajectories at different time scales. For example, considering localizations 1, 3, 5, 7… of trajectories recorded at 74 Hz, yields a dataset at 37 Hz. Since photobleaching is quite high under our spaSPT conditions, re-sampling the data in this way leads to smaller datasets. At 12.2 Hz, we only had 7,805 Halo-3xNLS trajectories and 32,563 Halo-CTCF trajectories. At this point, analyzing the as above by binning based on displacement length etc. causes the analysis to start to get noisy due to the limited amount of data, so we did not further extend the analysis to longer lag times. First, we plotted the probability of misconnections (after applying $r_{\text{threshold}} = 2\ \mu\text{m}$) as a function of the frame rate or lag time (**Supplementary Fig. 4e**). As observed before, Halo-CTCF had fewer tracking errors than Halo-3xNLS and the 133 Hz data had fewer tracking errors than the 74 Hz data (this is why the trend is slightly irregular, since the second point is composed exclusively of 74 Hz data, whereas the third point is composed exclusively of 133 Hz data). Nevertheless, the probability of misconnections did not increase with increasing lag time – if anything, it slightly decreased. Thus, this suggests that we can extend our analysis to substantially longer lag times than under which the data was originally recorded – for example, for Halo-CTCF at ~12.2Hz the probability of a tracking error is just 0.58%, that is, just 1 out of every 172 tracking connections was an error.

To further understand how tracking errors occurred in the temporally subsampled datasets, we repeated the analysis above for both Halo-CTCF and Halo-3xNLS at two representative frame rates: 35 Hz and 12.2 Hz (**Supplementary Fig. 4f**). As can be seen, most of the conclusions derived from the analysis of the 74 Hz and 133 Hz datasets, also apply to the temporally subsampled datasets (shown: 35 Hz and 12.2 Hz). In particular, the probability of a displacement being a misconnection reaches our somewhat arbitrary 5%-threshold once we reach displacements of ~1000 nm. Thus, this analysis suggests that we can reliably only analyze displacements up to about ~1000 nm, even at longer timescales if we want to make sure that we are not affected by tracking errors. Also, note that since the trajectories are quite short, when we temporally subsample very long lag times, the amount of data available decreases a lot, which is why the plots in **Supplementary Fig. 4f** are more noisy.

In summary, the above analysis and **Supplementary Fig. 4** show that tracking errors contribute minimally to our results as long as we do not consider displacements that are much longer than 800 nm.

Supplementary Note 3 - Comparision of ADTZ to other anomalous diffusion models

Many nuclear proteins exhibit subdiffusive behaviors[4,20], where their MSD grows as a power law with time $MSD_i(\tau) = \left(r_i(t+\tau) - r_i(t)\right) \sim \tau^{\alpha}$, with $\alpha < 1$. By analyzing the trajectories, we found that CTCF exhibits anomalous diffusion with an exponent in the range $0.83 < \alpha < 0.92$, depending on the time step (**Supplementary Fig. 1**; but note that the inferred α-value is quite sensitive to MSD-fit conditions). Observed anomalous diffusion has been attributed to motion in a crowded environment[21] or diffusion in a hypothetical fractal domain of the nucleus[22]. This motion is often phenomenologically modeled as a continuous time random walk (CTRW), where a particle is trapped for a time drawn from a long-tailed power-law distribution[23]. To examine the consequences of the ADTZ model on the dynamics of a protein, we computed the MSD of our simulation. We find that the protein appears to perform subdiffusive motion with an anomalous exponent of about $\alpha \sim 0.92$ (**Supplementary Fig. 7**; value is sensitive to conditions) due to the rapid reattachment at the release site. The value of α is not universal and depends on model parameters. Interestingly, higher reattachment probability leads to a slightly lower anomalous exponent (**Supplementary Fig. 7a,b)**. Thus, the ADTZ model can mechanistically explain the phenomenology of subdiffusive protein motion in the nucleus without invoking power-law dissociation times as is assumed in CTRW[23]. Importantly, CTRW would not, on its own, result in anisotropic motion[24], unless confined[20].

Motion anisotropy is also expected for a particle performing Fractional Langevin motion (fLm)[25], which results from the viscoelastic properties of the medium in which it moves. In fLm, anisotropy or antipersistence is identified as a negative value of the velocity autocorrelation function, suggesting that consecutive steps are anticorrelated. To examine whether fLm could explain the observed anisotropy of CTCF, we calculated the normalized velocity autocorrelation function $C \equiv C_v^{\Delta t}(\tau) / C_v^{\Delta t}(0)$ for different time steps against both time and scaled time (**Supplementary Fig. 8**). C has a negative peak and relaxes fast to zero, corresponding to the antipersistent/anisotropic behavior of CTCF. Interestingly, C is not invariant with the time-step as is the case for fLm[9] as the time-scaled autocorrelation function do not collapse – the value of the negative peak depends on the time step (**Supplementary Fig. 8b)**. This behavior is different from what would be expected for fLm[10]. For a particle performing CTRW in a confined domain, the velocity autocorrelation function is expected to show a single negative peak [10]. Hence, analysis of C suggests that neither fLm nor CTRW can account for CTCF dynamics.

To understand if the behavior of the autocorrelation function can be recapitulated by the ADTZ model, we computed the velocity autocorrelation function for the simulated trajectories. Interestingly, our model shows the same asymptotic relaxation to zero of C as the experimental data (**Supplementary Fig. 12**). Moreover, as in the experimental data, C is not invariant with different time-steps. The reason our model does not show a single negative value in C as would be expected for confined CTRW or time-step invariance of C as expected for fLm[10] is twofold: 1) The PTZs have a wide distribution of sizes. Since the time to explore each zone is different, we would expect to see negative values corresponding to all length scales of the zones. 2) The reattachment process appears as an antipersistent motion. The larger the time step is, the farther the protein can diffuse in a single time step, and the smaller is the probability of reattachment[26]. The distance traveled before reattachment has a power-law scaling and depend on the properties of the polymer [26].

Finally, we computed the probability distribution function (pdf)[1] of displacements for both the experimental data and simulated trajectories (**Supplementary Fig. 13**). Interestingly, we find the pdf to be non-Gaussian both for the experimental data and in our model. However, the experimental distribution is closer to a Gaussian than the model (see **Supplementary Table 3**). We find that the displacement pdf of mESC ΔZF-Halo-mCTCF (OE) is more Gaussian than that of mESC C59 wt-Halo-mCTCF (**Supplementary Table 3**). This suggests that the origin of the non-Gaussian behavior is the interaction with DNA and RNA. Our model's displacement pdf is more similar to the Laplace distribution. The non-Gaussian nature of the model pdf is likely the result of having different diffusivities in the model (the diffusion coefficient of CTCF is smaller in the zones). Our simulation of ΔZF-CTCF yield a more Gaussian displacement pdf consistent with the experimental data.

CTRW is not expected to result in anisotropic motion, while in fLm, anisotropy is the result of the long memory effect of the diffusing particle interacting with the viscoelastic environment in which it moves. Because we show here that the anisotropy results from the interaction (direct or indirect) with RNA and DNA, fLm can be ruled out as the main mechanism leading to anisotropy. Additionally, since chromatin moves much slower than a typical nuclear protein, a chromatin-bound CTCF molecule will appear immobile on a time scale of several milliseconds and will not contribute to the anisotropic motion on the scale of several hundred nanometers. We suggest that the primary contribution to wt-CTCF anisotropic dynamics is its interactions within RBR$_i$-mediated TTZs. Within a zone, CTCF is allowed to move and participate in transient interactions, which effectively reduce its in-zone diffusion coefficient. The ADTZ model can mechanistically explain sub-diffusion as coming from the retention and reattachment mechanisms (**Fig. 2**). While inside the zone, CTCF is reflected from its boundary with a certain probability (*retention*), resulting in a characteristic residence time within the zone. Once CTCF escapes one zone, it is very likely to rebind to the same zone because of its proximity (*reattachment*), which results in anisotropic behavior. Thus, the ADTZ model suggests that CTCF anisotropy results from the non-uniform distribution of zones inside the nucleus. Mechanistically, our results suggest that the transient trapping zones (TTZs) likely correspond to clusters of CTCF and that the trapping is mediated by CTCF's RBR$_i$-domain – this model is corroborated by our direct observation of anisotropic diffusion near CTCF clusters for wt-CTCF, but not for ΔRBR$_i$-CTCF (**Fig. 4**).

References

1. Lampo, T. J., Stylianidou, S., Backlund, M. P., Wiggins, P. A. & Spakowitz, A. J. Cytoplasmic RNA-Protein Particles Exhibit Non-Gaussian Subdiffusive Behavior. *Biophys. J.* **112,** 532–542 (2017).
2. Hansen, A. S. *et al.* Robust model-based analysis of single-particle tracking experiments with Spot-On. *Elife* **7,** e33125 (2018).
3. Hansen, A. S., Pustova, I., Cattoglio, C., Tjian, R. & Darzacq, X. CTCF and cohesin regulate chromatin loop stability with distinct dynamics. *Elife* (2017). doi:10.7554/eLife.25776
4. Saxton, M. J. A Biological Interpretation of Transient Anomalous Subdiffusion. I. Qualitative Model. *Biophysj* **92,** 1178–1191 (2007).
5. Saxton, M. J. Single-particle tracking: The distribution of diffusion coefficients. *Biophys. J.* **72,** 1744–1753 (1997).
6. Lee, A., Tsekouras, K., Calderon, C., Bustamante, C. & Pressé, S. Unraveling the Thousand Word Picture: An Introduction to Super-Resolution Data Analysis. *Chem. Rev.* **117,** 7276–7330 (2017).
7. Tiana, G. *et al.* Structural Fluctuations of the Chromatin Fiber within Topologically Associating Domains. *Biophys. J.* **110,** 1234–1245 (2016).
8. Grimm, J. B. *et al.* Bright photoactivatable fluorophores for single-molecule imaging. *Nat. Methods* 66779 (2016). doi:10.1101/066779
9. Jeon, J. H., Leijnse, N., Oddershede, L. B. & Metzler, R. Anomalous diffusion and power-law relaxation of the time averaged mean squared displacement in worm-like micellar solutions. *New J. Phys.* **15,** (2013).
10. Weber, S. C., Thompson, M. A., Moerner, W. E., Spakowitz, A. J. & Theriot, J. A. Analytical tools to distinguish the effects of localization error, confinement, and medium elasticity on the velocity autocorrelation function. *Biophys. J.* (2012). doi:10.1016/j.bpj.2012.03.062
11. Rasko, J. E. J. *et al.* Cell growth inhibition by the multifunctional multivalent zinc-finger factor CTCF. *Cancer Res.* **61,** 6002–6007 (2001).
12. Sergé, A., Bertaux, N., Rigneault, H. & Marguet, D. Dynamic multiple-target tracing to probe spatiotemporal cartography of cell membranes. *Nat. Methods* **5,** 687–694 (2008).
13. Persson, F., Lindén, M., Unoson, C. & Elf, J. Extracting intracellular diffusive states and transition rates from single-molecule tracking data. *Nat. Methods* **10,** 265–9 (2013).
14. Izeddin, I. *et al.* Single-molecule tracking in live cells reveals distinct target-search strategies of transcription factors in the nucleus. *Elife* **2014,** 1–27 (2014).
15. Michalet, X. & Berglund, A. J. Optimal diffusion coefficient estimation in single-particle tracking. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **85,** (2012).
16. Saxton, M. J. Modeling 2D and 3D diffusion. *Methods Mol. Biol.* **400,** 295–321 (2007).
17. Manley, S. *et al.* High-density mapping of single-molecule trajectories with photoactivated localization microscopy. *Nat. Methods* **5,** 155–157 (2008).
18. Elf, J., Li, G.-W. & Xie, X. S. Probing transcription factor dynamics at the single-molecule level in a living cell. *Science* **316,** 1191–4 (2007).
19. Yoon, Y. J. *et al.* Glutamate-induced RNA localization and translation in neurons. *Proc. Natl. Acad. Sci.* **113,** E6877–E6886 (2016).
20. Metzler, R., Jeon, J.-H., Cherstvy, A. G. & Barkai, E. Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking. *Phys. Chem. Chem. Phys.* **16,** 24128–24164 (2014).
21. Höfling, F. & Franosch, T. Anomalous transport in the crowded world of biological cells. *Reports Prog. Phys.* **76,** (2013).
22. Bancaud, A. *et al.* Molecular crowding affects diffusion and binding of nuclear proteins in heterochromatin and reveals the fractal organization of chromatin. *EMBO J.* **28,** 3785–98 (2009).
23. Metzler, R. & Klafter, J. The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Phys. Rep.* **339,** 1–77 (2000).
24. Burov, S., Jeon, J.-H., Metzler, R. & Barkai, E. Single particle tracking in systems showing anomalous diffusion: the role of weak ergodicity breaking. *Phys. Chem. Chem. Phys.* **13,** 1800 (2011).
25. Weber, S. C., Spakowitz, A. J. & Theriot, J. A. Bacterial chromosomal loci move subdiffusively through a viscoelastic cytoplasm. *Phys. Rev. Lett.* (2010). doi:10.1103/PhysRevLett.104.238102
26. Amitai, A. Chromatin Configuration Affects the Dynamics and Distribution of a Transiently Interacting Protein. *Biophys. J.* **114,** 766–771 (2018).