Advanced data analysis of single-molecule fluorescence signals: from viral particles to DNA sequencing

Qing Zhao

St Anne's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2025

Acknowledgements

I would like to express my heartfelt gratitude to all members of the Kapanidis group for their assistance and support throughout this project. I am especially grateful to those who made significant contributions:

Prof. Achillefs Kapanidis provided invaluable guidance and immense support throughout my DPhil journey. His insightful feedback and intellectual encouragement were instrumental in the completion of this project.

Dr Christof Hepp developed the protocol and carried out the majority of the experiments for influenza A virus DNA-PAINT. He also contributed the following plots: 2.8, 2.11, 2.12, 2.16, 3.3, 3.4, 3.5, and 3.9.

Dr Mirjam Kummerlin designed all the fluorogenic probes and performed fluorogenic PAINT experiments. She also made following plots: 3.11, and 3.13.

Dr Jagadish Hazara established the SPIN-seq method and conducted all related experiments, and big thanks to his work on Figure 4.6, 5.7, 5.9, 5.10, 5.13, and 5.11.

I would also like to thank Dr Stelios Chatzimichali for setting up the microfluidic system.

I would love to express my sincere thanks to all my colleagues for fostering such a supportive and friendly working environment. Dr Hafez El Sayyed for the delicious soup, Dr Heesoo Uhm and Dr Abhishek Mazumder for their help at the early stage of my DPhil, Dr Rasched Haidari, Dr Anna wang and Dr Emma Lalande for the nice chatting during the coffee break, Dr Alison Farrar and Dr Suchintak Dash for joining our own little trips, Sammi Ta for organising an amazing lab retreat.

Abstract

Single-molecule fluorescence microscopy is a powerful tool in the life sciences, enabled by advances in detectors, fluorophore and sequence design, and analysis methods. It reveals dynamic processes, intermediates, and molecular subpopulations that are invisible to ensemble measurements. By analysing the diffraction-limited image of an isolated fluorophore, emitter positions can be determined with high precision, forming the basis of SMLM (single-molecule localisation microscopy). SMLM yields quantitative information on the spatial distribution and density of molecular targets. Leveraging the specific and reversible binding of complementary nucleic acid strands, techniques such as DNA-PAINT (Point Accumulation for Nanoscale Topography) use transient, repetitive binding of short labelled oligos to their complements to generate stochastic "blinking" signals. The predictability and programmability of DNA interactions arise from hybridisation. The thermodynamic differences between complementary and mismatched strands allow us to design sequences with desired binding properties, identify nucleotides in a strand, and measure free energy from their binding behaviour.

We analysed DNA interaction data using two approaches: fluorescence time profiles and detectable localisations of fluorophores. One approach focuses on temporal aspects, while the other focuses on spatial aspects. We employed various algorithms to accommodate these differences and enhance analysis by combining their insights.

There are two primary components in this project: first, we developed a single-virion DNA-PAINT method to visualise the structure of viral RNPs (ribonucleoprotein complex), their relative spatial arrangements in IAV (influenza A virus) particles for the first time. The spatial organizations of vRNPs and their interactions were linked together to provide insights into the long-standing enigma of the IAV's genome assembly process. The ability to study the overall interactions from mature virus particles directly provides another puzzle piece compared to sequencing-based approaches like SPLASH, SHAPE, etc.

Second, we developed a new sequencing technique (SPIN-Seq) that can link the kinetic behaviours of molecule-DNA interactions with the DNA sequence on the

single-molecule level. We achieved a high accuracy (over 97% across all assays) and applied it to study two sequence-dependent biological processes: catabolite activator protein-DNA interaction and bacterial transcription initiation. SPIN-Seq is a new, parallel method to explore the sequence space for protein-DNA interactions in vitro without amplification or commercial sequencers.

Our experimental methodology can be applied using any TIRF (total internal reflection fluorescence) microscopy with single-molecule sensitivity, and the analytic pipeline can be easily applied to other data from SMLM. Thus, we believe that the methods developed in this project will be increasingly important and useful for future studies in fluorescence single-molecule biophysics.

Contents

List of Figures

List of Tables

List of Abbreviations

Influenza A virus and transcription

DNA . . . . . deoxyribonucleic acid

ss . . . . . single strand

ds . . . . . double strand

IAV . . . . . influenza A virus

RNP . . . . . ribonucleoprotein

HA . . . . . hemagglutinin

NA . . . . . neuraminidase

M . . . . . matrix protein

PA . . . . . polymerase acidic

PB1 . . . . . polymerase basic 1

PB2 . . . . . polymerase basic 2

NS . . . . . non-structural

NP . . . . . nucleoprotein

RNAP . . . . . RNA polymerase

cRNA . . . . . complementary RNA

CR . . . . . coding region

NCR . . . . . noncoding region

CAP . . . . . catabolite activator protein

cAMP . . . . . cyclic Adenosine monophosphate

RP_o . . . . . promoter open complex

ITC . . . . . initially transcribing complex

NTP . . . . . nucleoside triphosphate

Microscopy and fluorescence techniques

AFM . . . . . atomic force microscopy

FISH . . . . . fluorescence in situ hybridization

EM . . . . . electron microscopy

SMLM . . . . single molecule localisation microscopy

PSF . . . . . point spread function

SNR . . . . . signal to noise ratio

NA . . . . . numerical aperture

FOV . . . . . field of view

PAINT . . . . point accumulation for imaging in nanoscale topography

sCMOS . . . . scientific complementary metal-oxide-semiconductor

ROI . . . . . region of interest

TIRF . . . . . total internal reflection fluorescence

FRET . . . . . Förster resonance energy-transfer

IF . . . . . immunofluorescence

SPIN-Seq . . . Single-molecule Phenotyping and In-Situ Sequencing

FF . . . . . fluorogenic factor

Algorithm and calculation

MLE . . . . . maximum likelihood estimation

AIM . . . . . adaptive intersection maximization

ICP . . . . . iterative closest point

RCC . . . . . redundant cross-correlation

DBSCAN . . . density-based spatial clustering of applications with noise

HMM . . . . . hidden Markov models

FRC . . . . . Fourier ring correlation

PELT . . . . . pruned exact linear time

FP . . . . . frequent pattern

LC . . . . . localization counting

NN . . . . . nearest-neighbour

FFT . . . . . fast Fourier transform

IID . . . . . independent and identically distributed

DNN . . . . . deep neural network

Materials

BHQ . . . . . black hole quencher

PEG . . . . . polyethylene glycol

HBS . . . . . HPBES-buffed saline

HB . . . . . hybridization buffer

SSC . . . . . sodium citrate buffer

EDTA . . . . . ethylenediaminetetraacetic acid

HB . . . . . hybridization buffer

CB . . . . . clearing buffer

DB . . . . . DNA-PAINT buffer

IB . . . . . imaging buffer

AB . . . . . assay buffer

MQ . . . . . Milli-Q

1 Introduction

1.1 SMLM and DNA-PAINT

This section is a brief introduction to DNA-Point Accumulation In Nanoscale Topography (PAINT). It is a super-resolution technique that plays a critical part in this research and belongs to the family of single-molecule localisation microscopy (SMLM). SMLM significantly improves spatial resolution over traditional diffraction-limited microscopy and enables imaging of biological structures at the molecular scale.

1.1.1 Single-molecule biophysics and fluorescence microscopy

Traditional investigations in the life sciences have typically been conducted at an ensemble average level. It has advantages when observing homogenous samples that contain a large number of the same molecules or cells. However, many biological samples, such as cells, are inherently heterogeneous, displaying a wide range of biological, chemical, and physical properties. Even a supposedly homogenous population can never be truly homogenous. Relying on a population signature for these properties can obscure significant anomalies within the samples. This inherent variability may hold valuable insights that become overlooked when averaging. By concentrating on individual molecules, single-molecule biophysics research

explores the structure, dynamics, and interactions of single biomolecules, aiming to understand how they function both in vitro and in vivo.

Since the early success of recording single-ion channel activity in the 1970s [1], single-molecule techniques have evolved into powerful tools for the biophysical community. These techniques allow for the direct observation of non-uniform kinetics, rare molecular events, sub-populations of molecules, and transient states. They also enable the manipulation of individual molecules and the direct measurement of molecular forces. These capabilities of single-molecule techniques come with the advent of revolutionary techniques, such as atomic force microscopy (AFM), magnetic and optical tweezers, super-resolution fluorescence microscopy, etc.

Fluorescence microscopy is notable for its versatility and accessibility among various single-molecule techniques. Advances in biochemistry and genetics now allow for specific and precise fluorescent labelling of biomolecules, with minimal disruption to biological samples. The methods for optical excitation and detection enable visualisation of individual biomolecules with high spatial and temporal resolutions. Detecting emission from single fluorophores opens up exciting opportunities for studying biomolecules, as many molecular properties become accessible using fluorescence as a readout. For example, Förster resonance energy transfer (FRET) monitors the physical movement of biomolecules by analysing the correlated changes in the fluorescence intensities of two nearby fluorophores. Fluctuations in fluorescence signals enable the extraction of concentration and diffusion kinetics in fluorescence correlation spectroscopy. Visualisation of fluorescence in spatial coordinates (imaging) has been employed to study molecular geometry, relative positioning, and to track the movements of molecules.

One important consideration in single-molecule imaging is selecting the appropriate fluorophore. There is a vast selection of fluorophores, such as fluorescent proteins, organic dyes, and quantum dots. Ideally, the chosen fluorophore should

be bright, with high quantum efficiency and a high extinction coefficient. It should also be photo-stable (a high photon budget that allows for long-term imaging) and not interfere with biological activities. Additionally, it is preferable that the fluorophore emits light in a spectrum that corresponds to the camera's highest quantum efficiency and possesses tunable chemical and photophysical properties to meet the specific demands of different single-molecule experiments.

When observing the behaviour of single molecules under applied force, techniques like AFM or optical/magnetic tweezers are often required. In contrast, observing them in systems without external force is typically conducted under standard widefield microscopy. Widefield microscopy enables the simultaneous imaging of thousands of molecules, allowing for the collection of large amounts of data for analysis and increasing the likelihood of detecting rare events. However, stray signals outside the focal plane can interfere with the molecules of interest by creating background fluorescence, which lowers the signal-to-noise ratio (SNR). Furthermore, it is often necessary to image at high frequency (low exposure time), which can further push the sample into a low SNR environment. This issue can be mitigated by employing confocal microscopy, total internal reflection fluorescence (TIRF) microscopy, and utilising sensitive, high-speed cameras.

1.1.2 Single-molecule localisation microscopy

The diffraction pattern of an arbitrarily small light source imaged using a lens-based microscope is called the point spread function (PSF), which is typically an Airy disk with a central peak radius of approximately 200-300nm. It prevents traditional optical microscopy from resolving subcellular structures. New techniques, such as stimulated emission depletion microscopy, structured illumination microscopy, and single-molecule localisation microscopy (SMLM), have been developed to overcome this diffraction limit.

The idea behind SMLM is that the spatial coordinates of fluorophores can be determined with high precision when their PSFs do not overlap. This technique allows for localisation precision that is limited by the SNR, rather than the wavelength of light. By temporally separating the fluorescence signals from densely labelled objects, SMLM produces super-resolved images with nanometer resolution. Fluorescent molecules contribute to this separation by switching between "on" (bright) and "off" (dark) states. They typically blink over several thousand frames or more, and the accumulated localisations are then combined to create a single high-resolution image.

SMLM only requires a wide-field microscope equipped with standard continuous-wave lasers and a camera capable of detecting single molecules. The sample's fluorescence emission is typically imaged through an oil-immersion objective lens with a numerical aperture (NA) of 1.4 or higher to facilitate effective photon collection. The pixel size of the camera should be approximately equal to the half-width of the PSF, typically between 100 and 150 nm. To achieve 3D localisation, one can encode and decode the axial position into and from the PSF. Modifications in optical setups, i.e., exploiting PSF engineering by using cylindrical lenses, phase masks, multifocal planes, etc., have enabled the localisation of molecules in three dimensions. The most common approach is astigmatic 3D. Astigmatism is a type of off-axis aberration where the PSF appear to be elongated in different directions above and below the focal plane. However, PSF shaping or multi-plane detection diverts photons, resulting in the loss of lateral localisation accuracy.

1.1.3 DNA Point Accumulation In Nanoscale Topography

Currently, there are three main techniques in SMLM: photoactivated localisation microscopy, direct stochastic optical reconstruction microscopy, and DNA point accumulation in nanoscale topography (DNA-PAINT). In DNA-PAINT, a pair of short, complementary single-strand DNAs enables the blinking effect. One DNA strand with a docking extension is immobilised on the target molecule, and

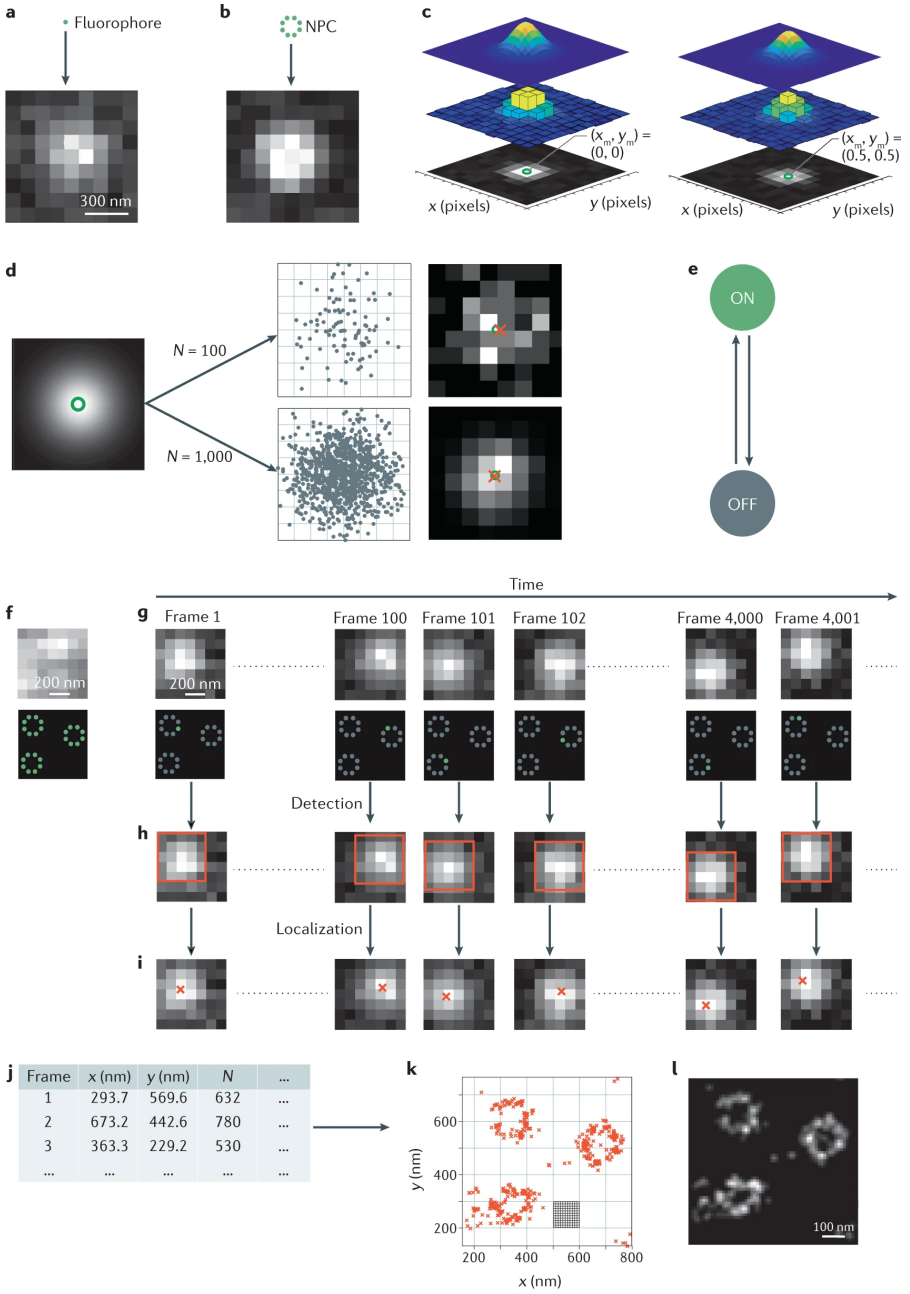

Figure 1.1: working principle of SMLM. a and b show the diffraction patterns from an isolated emitter and from dense sources, respectively. In b, the PSFs from simultaneously active emitters overlap, causing a blurring of the structure. c demonstrates that even a sub-pixel shift can lead to detectable changes in the PSF. Figure d highlights the impact of photon number (indicated above the arrows) on the pixel values recorded by the camera. A higher photon count results in better SNR and increased localisation precision. f presents an experimental image of a nuclear pore where fluorophores are activated simultaneously. g, h, and i illustrate how information accumulates over time in SMLM. A stochastic subset of emitters is recorded at different frames. This temporal separation is achieved by leveraging the emitters' ability to switch between on and off states, as illustrated in e. The coordinates of the emitters are calculated and summarised in a table, depicted in j. The accumulated localisations can be visualised as a point cloud (k) or rendered into a super-resolved image, as shown in l. (image adapted from [2])

its complementary strand, known as the imager, is labelled with a fluorophore. Generally, freely diffusing imagers cannot be detected by the camera because they move across multiple pixels within a single frame, resulting in a blurred background. However, when these imagers transiently bind to the docking strands, they are fixed at a specific position for a short period, allowing the camera to capture the accumulated photons and making the imagers detectable. DNA-PAINT can be employed for any target molecules that can be linked to a docking strand, including antibodies, nanobodies, aptamers, affimers, or genetically encoded tags.

Image quality of SMLM is dependent on the imaging process. Characteristics of the fluorophore, such as brightness, photon-induced damage to the docking site, the amount of nonspecific binding of the imager strand, and binding kinetics, all play a part in image quality. The brightness and SNR are the main factors which determine the localisation precision. The depletion of binding sites caused by dye-induced generation of reactive oxygen species [3] reduces the sampling and therefore has a negative influence towards the overall image quality. This damage is dependent not only on the choice of fluorophore but also on laser power and wavelength. The choice of fluorophore also influences the proportion of off-target localisations (sticking), which affects both image quality and subsequent quantitative analysis. The binding duration τ_b depends on the stability of the DNA duplex and is inversely proportional to dissociation rate (k_{OFF}) and can be engineered by strand length, GC content, salinity of the buffer, etc. τ_b increases by roughly an order of magnitude when the length of the imagers increases by 1bp. It can be tuned to extract the highest number of photons per binding event, thereby achieving high localisation precision [4]. Meanwhile, the frequency of binding events can be tuned by the influx rate of the imager, such as the concentration of imager strands, and the hybridisation rate (k_{ON}).

For a multicolour image, one can employ spectrally separated fluorophores and a dichroic filter with multiple pass bands or use exchange-PAINT, which targets

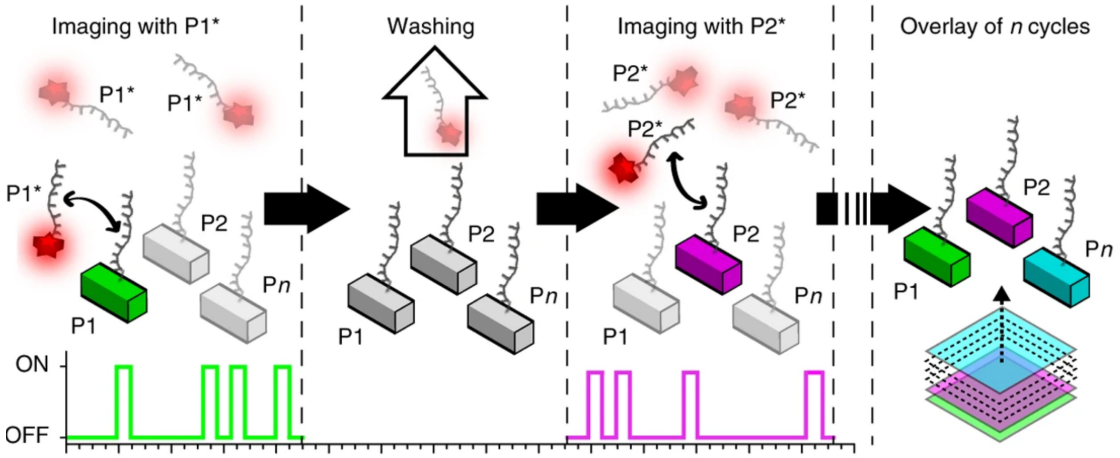

Figure 1.2: working principle of sequential exchange-PAINT. Exchange DNA-PAINT images multiple targets using orthogonal sequences with the same fluorophore. The imager $P1^*$ interacts with its complementary sequence, which is fixed on target $P1$, to enable imaging of $P1$. After the first acquisition round, $P1^*$ is washed away, and $P2^*$ is introduced to image $P2$. This process is repeated to image multiple targets. Finally, all images from different rounds are aligned and overlaid to create a multiplexed image, with a pseudo-colour assigned to each target. (image adapted from [5])

the molecules of interest with orthogonal docking strands. In exchange-PAINT, different biological targets are imaged sequentially by respective complementary imagers. After one DNA-PAINT image is acquired, the imaging buffer is changed to remove the old imager strands and introduce new imagers. By repeating this removing and reintroduction process, one can acquire super-resolution images with a large number of biological targets (1.2).

There are numerous software developed to analyse SMLM data. We chose Picasso [5] (a collection of tools for PAINT) as a backend for localisation. It does localisation detection by extracting local maxima from the neighbourhood. This local area is a square box with a length set by the user. Then the net gradient (G_{net}) is calculated for each square around a local maximum by

$$G_{net} = \sum_{box} g_i \cdot \mathbf{u_i} \tag{1.1}$$

where g_i is the central difference gradient at pixel i and $\mathbf{u_i}$ is a unit vector originating at pixel i and pointing towards the central pixel of the square box. The pixel regions identified during detection are analysed further to calculate sub-pixel coordinates if

they have gradients exceeding a user-defined threshold. Among many published spot-fitting algorithms, one standard of localisation algorithm is maximum likelihood estimation (MLE) because it approaches the Cramer-Rao lower bound at high SNR [6–8]. The MLE algorithm implemented in Picasso assumes a CCD camera which features constant gain and offset across all pixels, and the per-pixel model required for sCMOS cameras is not supported. Therefore, the advantages of the MLE algorithm can not be utilized without substantial modification of Picasso's localisation method. Picasso also offers two alternative localisation algorithms: least-square Gaussian and average of the region of interest (ROI). The Gaussian fit demonstrates robustness to minor pixel-to-pixel variations. Additionally, the Gaussian fit directly yields the PSF widths, which are necessary for the astigmatic 3D method employed in this study. For these reasons, the Gaussian fit was selected for localisation. Localisation precision is calculated by [7]

$$\delta \geq \sqrt{(\frac{\delta_0^2 + a^2/12}{N})(\frac{16}{9} + \frac{8\pi\delta_0^2 b^2}{a^2 N^2})} \tag{1.2}$$

where a is the pixel size, b is the background intensity, N is the photon number, and δ_0 is the Gaussian width of the PSF. After acquiring localisations, some post-processing steps, such as filtering out suboptimal localisations, can be applied based on parameters like the width of the Gaussian, the estimated photon number, and the background.

1.2 Quantitative analysis in SMLM

In traditional microscopy, data is represented as an intensity pixel or voxel in arrays. In contrast, SMLM data is essentially a point cloud. This means that SMLM data can be analysed by either generating an image from these coordinates or directly dealing with the coordinates. As a result, some image processing tasks, such as registration and segmentation, need to be approached differently than in conventional microscopy. A wide range of data analysis methods has been applied to extract quantitative information from SMLM data regarding the distribution, topology, and

spatial organisation of biological samples. Quantitative studies—including counting proteins, analysing spatial arrangements and co-localisation between organelles, as well as single-particle tracking—are routinely conducted on SMLM data.

To achieve accurate quantitative measurements, several issues must be addressed. High-frequency vibrations can blur individual, diffraction-limited frames, affecting localisation precision and cannot be corrected computationally. Therefore, isolating the microscope from sources of vibration is crucial. Sample drift is a common phenomenon in SMLM. It occurs when the sample shifts relative to the objective during movie acquisition, leading to distortion and degradation of the data. Typically, axial drift can be corrected in real-time using hardware, while lateral drift correction is performed after data acquisition. Drift can be measured by tracking fiducial markers or by using marker-free methods. Various types of fiducial markers are available, including fluorescent beads, gold particles, and quantum dots. These markers are effective when they remain stationary throughout the entire acquisition process and do not interfere with the sample being imaged. They are particularly suitable for TIRF imaging, as the targets and samples are in the same focal plane. However, when these conditions are not met, it is essential to employ marker-free algorithms. Numerous marker-free algorithms have been proposed, such as redundant cross-correlation (RCC) [9], the mean shift algorithm [10], entropy minimisation [11], and adaptive intersection maximisation (AIM) [12]. These approaches assume a stable and static dataset, and may not perform reliably if the structure is dynamic in the conditions under investigation or deforms over time due to external experimental factors.

Among these methods, we chose the AIM algorithm for its precision, efficiency and robustness to noise. For an ideal drift-free system, the localisations detected from the same emitter across the span of all frames should be intersected with a radius of localisation precision. AIM estimates the drift by a two-stage strategy to maximise the interaction of localisations across temporally distinct datasets.

The first stage generates a preliminary estimation of drift from segmented sub-datasets by maximising the intersections between the reference segment and other temporal segments. Then, the preliminary corrected dataset is used as the new reference for the second stage of refined drift correction. In practice, AIM takes three parameters: segmentation (the number of frames per segment), intersection distance (the maximum distance between two localisations in two temporally consecutive segments to be considered as the same molecule), and maximum drift in segment (the maximum expected drift between two consecutive temporal segments) to correct localisation coordinates. Other preprocessing methods, which are not routinely used in this project, will be introduced at the corresponding parts.

When working with point clouds, clustering techniques are often used to group localisations into biological structures, aiding in the visualisation and interpretation of the data. Clustering is a task that classifies a given set of data into subsets, where data points within a subset are more similar to each other based on the specified properties. Clustering approaches commonly used in SMLM estimate local density and construct clusters from the detections with density above a threshold. Ripley's K-function [13] and its variants, as well as Density-based spatial clustering of applications with noise (DBSCAN) [14], estimate density based on the number of localisations within a specific distance. The Voronoi diagram [15] uses the area of tiles in the associated tessellation. When the data contains structures with various densities, clustering methods can be repeated with different parameters to segment different components. More recent advances offer an alternative approach to handling varying densities through persistence or topographic prominence. Topological data analysis provides a framework for detecting topological properties and examining the shape of the point cloud [16]. Additionally, Bayesian clustering algorithms have also been utilised in SMLM data. For instance, Griffié et al. developed a method to classify localisations as either background or signal using a binomial distribution [17].

While clustering methods help us identify biological molecules, co-localisation analysis helps us find molecules that may have a functional relationship. Co-localisation analysis has been applied to traditional optical fluorescence microscopes to understand their role and biological processes, but suffers from a spatial resolution of about half a wavelength. With the advent of super-resolution microscopy, co-localisation analysis is approaching molecular resolution, bridging the gap between FRET, which typically operates in the range of about 2-10 nm, and traditional fluorescence microscopy. Co-localisation analysis methods can be categorised into two types: pixel-based methods, which measure the global correlation between different channels, and object-based methods, which first segment molecules and then analyse their spatial distribution using statistical methods. One can construct super-resolution images and apply standard co-localisation analysis methods on these constructed intensity-based images. Therefore, the results will depend on the process of image construction, which is often done by histogram or blurring localisations. Because two molecules are almost impossible to be at the exact same position, the co-localisation in SMLM is usually defined as intermolecular distance, or spatial association. In this context, distance to the nearest neighbour, Ripley's K function, and pair correlation function can all serve as references. Additionally, clustering algorithms can also be extended for co-localisation analysis by taking data from other channels into consideration.

We also applied time series analysis on the fluorescence time profile extracted from the potential position of molecules of interest. The time series is one of the most fundamental representations of sequential data, characterised by its numerical and continuous nature. Concepts such as clustering, classification, and segmentation in general machine learning have been widely applied to the time-series domain. There are three origins of fluorescence traces in this project: the binding of the imager strand and its docking strand, the binding of the labelled protein and its consensus sequences, and FRET efficiency. Although the biological interactions and processes in question differ, the overall goal is to reconstruct the kinetic scheme

from traces that are complicated by measurement noise, interference from nearby bound fluorophores, and photophysical effects.

Various algorithms have been employed to analyse single-molecule time series, including non-parametric Bayesian methods, change point detection, total variation methods, and step detection algorithms, such as the Kalafut–Visscher algorithm [18]. The hidden Markov model (HMM) is particularly popular in this context; it is a probabilistic model used to represent systems that transition between unobservable states over time while producing observable outputs. However, contemporary methods often face challenges related to model selection bias and parameter identifiability. We must assume the existence of a certain number of biophysically relevant states and possibly their interrelationships. Applying a specific HMM to datasets with significant per-molecule variance in intensity, behaviour, and noise level proves to be difficult. Numerous variants of the HMM have been proposed to address these issues. In addition to HMM, we have employed two change point detection methods, which will be detailed in the corresponding section.

1.3 DNA hybridization and stacking

Our understanding of how DNA sequences encode information has advanced significantly since the early successes of solving the double helix structure and establishing the central dogma. The very utility of DNA as a carrier of hereditary information lies in its ability to undergo hybridisation. The predictable and specific Watson-Crick hybridisation makes nucleic acids so useful in biology, biotechnology, nanoscale engineering, and other fields. Some early studies measured thermodynamic parameters for the formation of double strands between complementary and mismatched short strands [19].

A single mismatched base pair can cause a significant decrease in the binding affinity of a DNA duplex compared to a perfectly matched duplex [20]. This characteristic enables high specificity in target-probe hybridisation, which is crucial

for distinguishing between sequences that are often highly similar. The impact of a mismatch on the stability of a DNA duplex depends on its location, the nearest neighbours, and the orientation of the mismatch [21]. A mismatch located in the middle of the strand is generally less stable than one positioned at the ends. To accurately represent internal mismatches, it is more effective to use trimer sequences with mismatches in the central position rather than the usual dimer sequences employed in nearest-neighbour (NN) parameters. The thermodynamic advantage gained from correctly paired bases can outweigh the destabilising effect of a mismatch. However, cross-hybridisation—where probes bind to targets that are not perfectly matched—can become kinetically trapped and hinder the desired hybridisation. To effectively discriminate between mismatches, one must enhance the yield of the intended hybridisation. Various methods have been employed to achieve this, including molecular beacons, sequence engineering, the use of artificial nucleases, temperature adjustments, and the addition of denaturing agents. These strategies shift the Gibbs free energy of both the intended (ΔG_{intend}) and unintended ($\Delta G_{unintendent}$) hybridisations in the same direction. Moreover, a natural conflict exists between specificity and sensitivity [22], making it essential to find a balance between these two requirements. Ideally, increasing the difference in Gibbs free energy between the two hybridisations would provide an effective solution.

In 2012, Zhang et al. [23] showed that a hybridisation reaction is specific when there is a large difference between the hybridisation yield of the intended target X and spurious target S. They defined specificity as discrimination factor $Q = \chi_X/\chi_S$ where χ_X and χ_S are hybridization yields of X and S respectively. The upper bound of Q is

$$Q < Q_{max} \equiv e^{\Delta\Delta G^o/RT}$$
$$\Delta\Delta G^o = (\Delta G^o(SC) - \Delta G^o(S) - \Delta G^o(C)) - (\Delta G^o(XC) - \Delta G^o(X) - \Delta G^o(C)) \tag{1.3}$$

$\Delta\Delta G^o$ is the difference in standard free energies of the hybridisation reaction for X and S, ΔG^o is the Gibbs free energy, R is the ideal gas constant, and T is the

temperature. The standard free energy difference caused by a single base determines the upper bound of the discrimination factor. This helps design hybridisation probes that would enable near-optimal single-base discrimination.

Except for the base pairing between opposite strands mediated by hydrogen bonds, the stacking between adjacent base pairs also plays an important part in DNA stability. Earlier attempts to quantify the contribution of base stacking interactions to the stability of double-stranded DNA measured the stacking energy on nicked/gapped DNA fragments or used thermal denaturation. More recent attempts include measuring force-dependent dissociation rates using optical tweezers [24], direct measurements between di-nucleotides using centrifuge force microscopy [25], and accessing an imager's binding kinetics with stacking and non-stacking configurations using DNA-PAINT [26].

The unified NN model has been developed to predict sequence-dependent DNA thermal stability without separating base-pairing and base-stacking interactions. In this model, the thermodynamic stability of duplex base-pairing can be determined by the sum of free energy changes due to all adjacent NN base pair formations in the sequence and helix initiation. It is used in melting temperature & secondary structure prediction, and modelling hybridisation kinetics. Recent studies started to determine the NN parameters under conditions of sequence mismatch [27, 28] and at low sodium concentrations [29]. The properties of mismatched base pairs depend strongly on their NN configuration [30]. Evaluating the dependence of the thermodynamic and structural properties of mismatches with all possible NNs is a challenging problem. The advantage of the NN model is numerical efficiency, but it does not provide any insights into intramolecular interactions. Other theoretical methods, including both microscopic and mesoscopic models, can also handle mismatches. For example, Oliveira et al. published 4096 melting temperature combinations covering single, double, and triple mismatches, using the Peyrard–Bishop model for analysis [31]. This statistical physics model employs

microscopic potentials to account for hydrogen bonding and stacking interactions [32].

1.4 Purpose of this research

We have two primary goals in this project. First one was to uncover the mechanism of influenza A virus (IAV) genome packaging by accessing the packaging stoichiometry, and spatial organisation of the IAV segments in large populations of single virions using DNA-PAINT. We derived a network of intermediate complexes from the statistical analysis of defective particles. To quantify segment packaging fidelity, we determined the fraction of virions with a complete genome. We also visualised the spatial organisation of viral ribonucleoprotein (vRNP) in single virions and its heterogeneity by constructing super-resolved images and volume rendering. By studying the diversity of relative vRNP orientation within an ensemble of virus particles, the heterogeneity can be quantified. We specifically localised all vRNPs and assessed their structural arrangement. These results helped us determine the relation between the genome organisation and vRNP abundance in a large number of virions.

Secondly, we developed a new, rapid, parallel single-molecule sequencing method that determines nucleotide sequences by measuring the kinetic differences between complementary DNA strands and strands with mismatches on gapped DNA substrates. As a proof of concept, we successfully sequenced strands of one, three, and five bases, achieving an average accuracy rate of over 97%. We then applied this method to investigate two sequence-dependent biological processes: catabolite activator protein binding its consensus DNA and transcription initiation. This method allowed us to link kinetic properties with DNA sequences at the single-molecule level. The entire experiment was conducted on the same slide, without the use of any special devices.

In both parts, we will address two types of data: point clouds representing labelled DNAs interacting with their complementary strand, and time series of fluorescence intensities, which indicate interactions or conformational changes in the molecules of interest. The spatial information in the point cloud allows us to localise and visualise biological molecules, as well as to quantify their distances and co-localisation. Meanwhile, the time series data provides valuable kinetic information, such as binding and unbound time, which reflect the underlying biological processes. A detailed description of the analysis methods will be provided in the corresponding sections. In the following chapters, we will present our methods and findings regarding IAV in Chapters 2 and 3. The details about our single-molecule phenotyping and sequencing method, SPIN-Seq, will be covered in Chapters 4 and 5.

All work that follows is my own except where material from other sources has been explicitly cited and referenced. All contributions from collaborators have been fairly acknowledged and clearly stated in the texts.

2

Structure analysis of IAV genome enabled by exchange PAINT

2.1 Structure, life cycle and genome of influenza A virus

Influenza A virus (IAV), a member of the Orthomyxoviridae family, is a significant viral pathogen in humans, known for causing seasonal epidemics and occasional pandemics. IAVs are categorised into subtypes based on two proteins: hemagglutinin (HA) and neuraminidase (NA). These proteins are situated on a host-derived lipid membrane, which is supported by matrix protein 1 (M1) and matrix protein 2 (M2). There are 18 different hemagglutinin subtypes (H1 to H18) and 11 neuraminidase subtypes (N1 to N11) found in nature, leading to various combinations [33]. While over 140 subtypes have been documented [34, 35], there may be even more combinations resulting from virus reassortment, with birds and pigs serving as their reservoirs. Additionally, IAV subtypes can be further classified into different genetic clades and sub-clades. Although clades and sub-clades differ genetically, they are not always antigenically distinct.

Antigenic shift is very prominent in seasonal flu, occurring when reassortment yields new IAV subtypes. The protection of vaccines against IAVs is limited by

the variation in HA and NA envelope glycoproteins [36]. Furthermore, we face the challenge of limited antiviral drugs and the increasing resistance to them. Thus, it is critical to understand the mechanism of IAV replication in order to cope with potential future pandemics and develop new antiviral medications.

In addition to the previously mentioned matrix proteins, IAVs also have two types of non-structural proteins: NS1 and NS2. NS1 is derived from unspliced mRNA, while NS2 comes from spliced mRNA. NS1 is one of the first proteins expressed during a viral infection and is well-characterised as a potent antagonist of type I interferon [37]. In contrast, NS2 is expressed at a later stage during the infection and plays a crucial role in nuclear export [38].

Infection initiates when IAVs attach to the potential host cell via HA receptor-binding sites, followed by endocytosis, during which the substances to be internalised are surrounded by an area of plasma membrane, and then bud off into the cell, forming an early endosome containing the virion. As the endosome matures, the low pH triggers conformational changes on the HA molecules, resulting in the fusion of the viral and endosomal membranes [39]. vRNPs that are released from endosomes are transported into the nucleus through the nuclear pore complex. In the nucleus, the heterotrimeric viral RNA-dependent RNA polymerase (RNAP) carries out the transcription and replication of viral RNA (vRNA). The replication process consists of two steps: the synthesis of complementary RNA (cRNA) and the synthesis of new vRNA using cRNA as a template. The primary transcription results in the production of viral mRNA, which is exported into the cytoplasm for translation by cellular ribosomes. After translation, the RNAP sub-units (polymerase acidic(PA), polymerase basic 1 (PB1), and polymerase basic 2 (PB2)) and nucleoproteins (NPs) are imported into the nucleus and form progeny vRNPs with vRNA. Following nuclear export, progeny vRNPs are transported across the cytoplasm to the cell membrane, where the assembly of progeny virions occurs before being released from

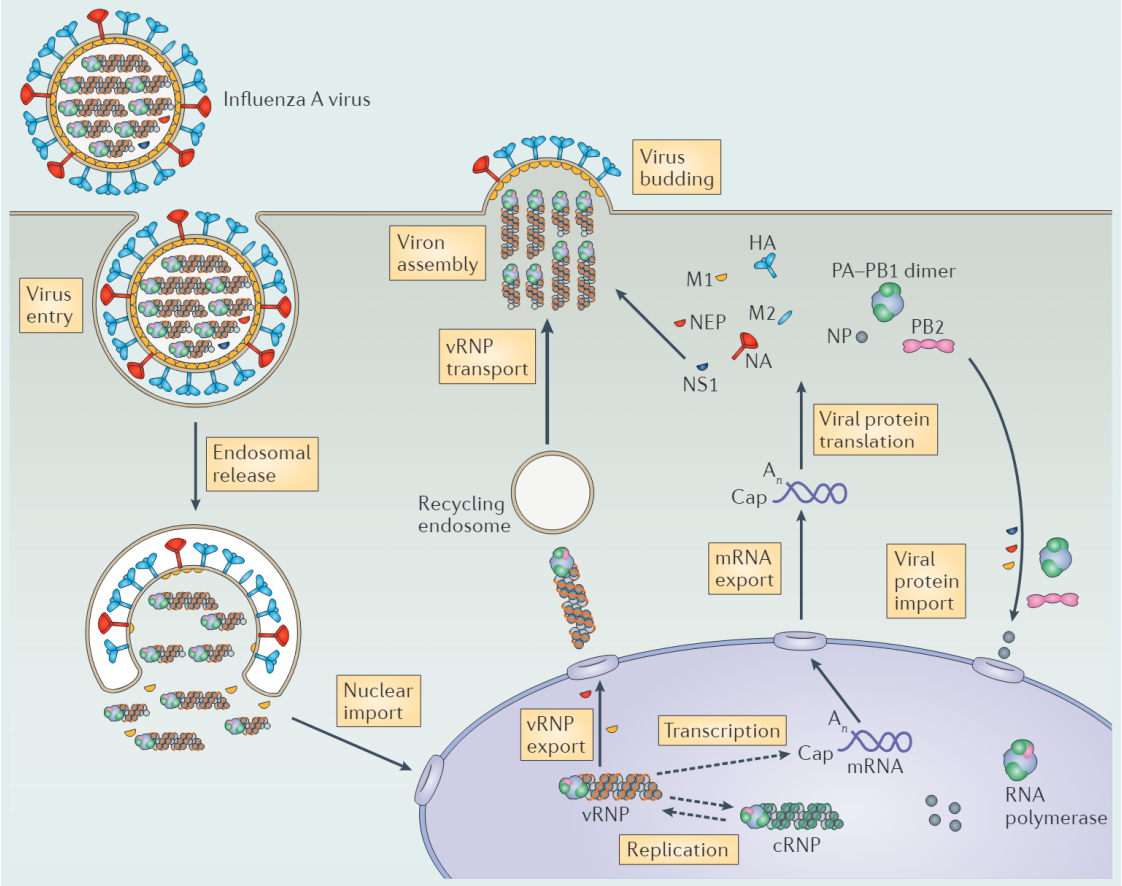

Figure 2.1: life cycle of IAVs. IAVs enter cells by endocytosis, after which the vRNPs are released into the cytoplasm and transported into the cellular nucleus, where transcription and replication occur. The newly synthesised vRNPs are exported to the cytoplasm, subsequently transported to the plasma membrane and incorporated into the progeny virus before budding. (image adapted from [40])

the cell membrane (2.1).

The genome of IAVs comprises eight single-stranded, negative-sense RNA segments. Despite the eight RNA (PB1, PB2, PA, HA, NA, NP, M, and NS) having various lengths from 2341 to 890 nt, they share the same structure. A central coding region is flanked by segment-specific non-coding regions (NCRs) and the terminal promoter regions U12 and U13 (12 and 13 nucleotides from the 3' and 5' ends, respectively) as shown in Figure 2.2A. These RNAs wrap around twisted rod-like vRNP complexes in the virions. Within a vRNP complex, NPs bind to vRNA while the 5' and 3' termini of the RNA segment are bound by an

RNAP. The diameter of the vRNP rods is about 13 nm, and their lengths vary between 30 and 110 nm [41]. Previously, the vRNPs were thought to lack secondary and tertiary structures (2.2B). However, recent studies show that the NPs prefer to bind to guanine-rich and uracil-poor regions on RNAs [42, 43], and they bind to about 12 nucleotides of vRNA with 25 nucleotides between adjacent binding sites on average [44]. The revised vRNP models have NP-free RNA regions where secondary and/or tertiary structures protrude from the twisted rod-like body (2.2C).

Several electron microscopy (EM) studies show that the vRNPs are arranged in a '7 + 1' pattern (seven vRNPs surrounding a central one) in virions. Noda et al. showed that a mutant virus that only contains seven vRNAs (lack of HA) also formed a '7 + 1' arrangement using host-derived ribosomal RNAs as the eighth RNP[46]. This is consistent with influenza C and D viruses, which naturally have seven vRNPs found in the same arrangement [47]. It suggests that the '7 + 1' configuration plays an important role in the selective packaging. In the electron tomography study by Noda et al. in 2012[48], some string-like structures are observed connecting neighbouring vRNPs through their entire lengths. They might be the secondary and/or tertiary structures mentioned previously, and the '7 + 1' arrangement is formed through multiple interactions via these structures. As for the specific sites on RNA where the vRNPs can interact with each other through base pairing, they are detected from both coding regions (CRs) and non-coding regions (NCRs) in all eight segments. However, mutations in these regions do not always cause a reduction in packaging efficiency. It is thought that each segment-specific packaging signal is a cluster made up of multiple, discontinuous short-nucleotide elements [49].

Evidence suggests that not all eight vRNAs are equally important in the packaging process, indicating a hierarchy in selective packaging [50, 51]. Single-molecule fluorescence in situ hybridisation (FISH) experiments also provide evidence for the selective packaging model, demonstrating that most virus has one copy of

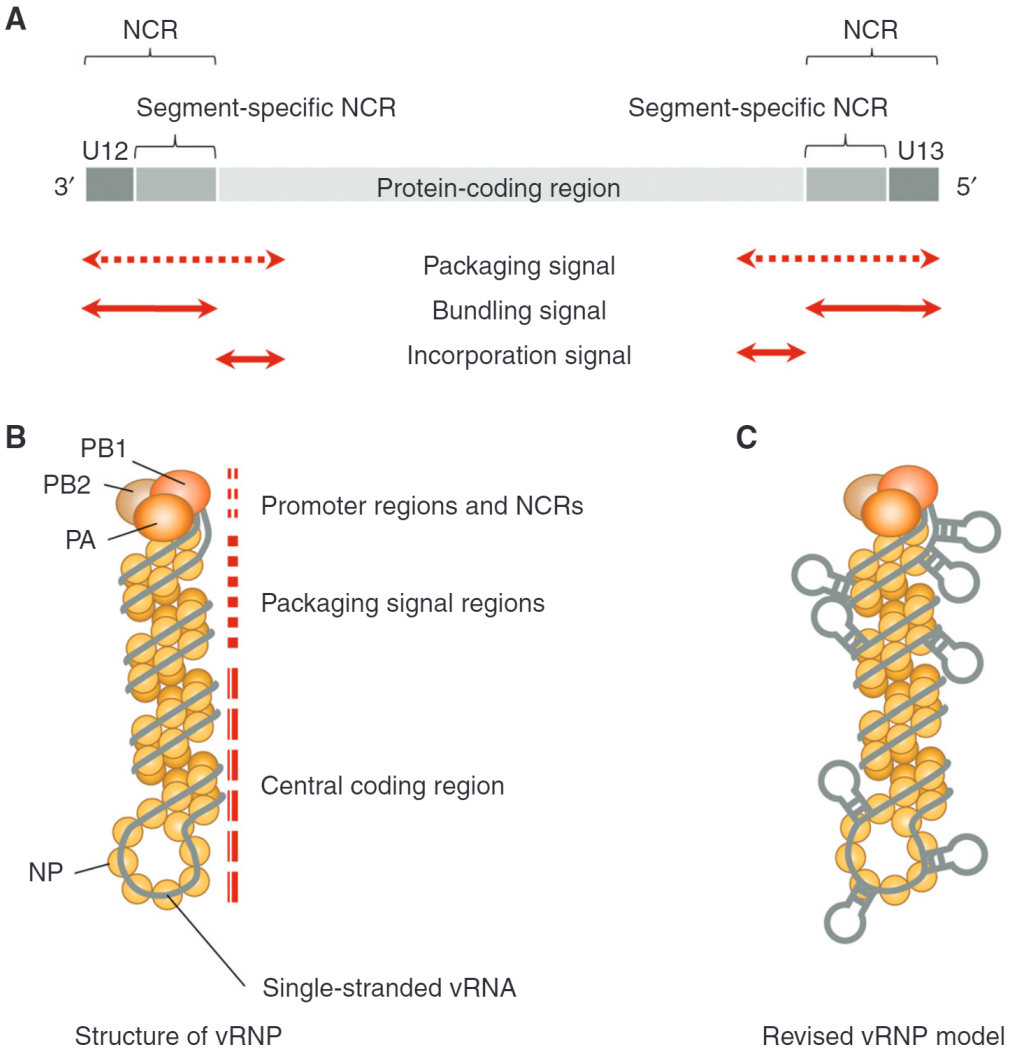

Figure 2.2: structures of RNA and RNP of IAV. A: A protein coding region is flanked by segment-specific non-coding regions and terminal regions. Different arrows indicate the packaging signal, bundling signal and incorporation signal. B: The conventional model of vRNP shows predicted promoter, CRs, NCRs and packaging signal regions in the rod-like vRNP. C: The revised vRNP model shows secondary and/or tertiary structures form at NP-free regions protruding from the rod-like body. (image adapted from [45])

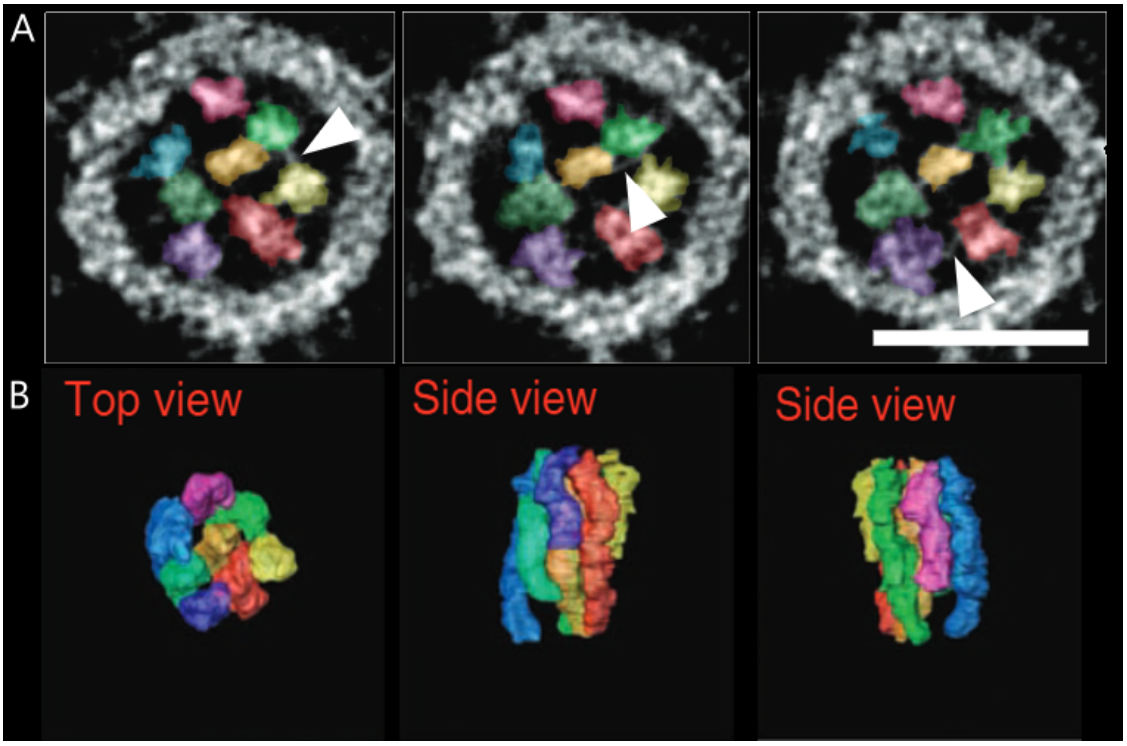

Figure 2.3: tomography and 3D model of the eight vRNPs within a virion. A displays the tomograms of 0.5-nm-thick sections, highlighting short string-like structures with arrows. B presents the top and side views of a 3D model depicting the eight vRNPs arranged in a '7 + 1' configuration, with each vRNP represented in a different colour. (images adapted from [48])

each of the eight segments [52]. Analysis of the combinations of different vRNAs in individual virions reveals that the number of virions with a complete genome is significantly higher than what would be expected from random selection [53]. In addition to the interactions among vRNAs through base pairing, NPs play a crucial role in selective packaging. Introducing mutations into specific NP residues can alter the packaging efficiency of vRNAs [54, 55]. Furthermore, the efficiency of defective packaging can be restored by mutating NP residues [56]. The interactions between vRNAs and NPs can influence the secondary and/or tertiary RNA structures, subsequently affecting RNA-RNA interactions. Sequencing of psoralen cross-linked, ligated, and selected hybrids (SPLASH) has been used to investigate direct vRNA-vRNA interactions within the context of viral ribonucleoproteins (vRNPs). Most vRNAs show multiple and redundant inter-segment interactions at both NCRs and CRs [57]. However, many of the interactions identified using SPLASH have yet to

be confirmed as having a significant impact on the packaging process.

Generally speaking, the introduction of a new influenza virus into humans from animals poses significant global health threats, and IAV has been the subject of intensive research for the last few decades. Numerous techniques have been employed to study the vRNP structures and their arrangements. Accumulating evidence suggests that the interactions between the eight vRNPs in IAV play a role in the selective genome packaging process. However, the overall picture of the assembly process of the vRNPs, the network of their interactions, heterogeneity in their structures and arrangement are still long-standing enigmas.

2.2 Overview of single-virion DNA-PAINT

To understand the role of individual vRNPs and their intermediates in selective packaging, we conducted a series of analyses on the configuration of vRNPs and packaging defects in H1N1 virus particles. A multiplexed DNA-PAINT method, which incorporates both super-resolution (discussed in Chapter 2) and stoichiometry assays (covered in Chapter 3), was developed. By integrating the findings from these two types of experiments, we gained valuable insights into the assembly process of the segmented genome (2.4).

In Chapter 2, we will employ 3D astigmatic DNA-PAINT to investigate the structures and relative spatial arrangements of vRNPs. Our analysis will focus on the distribution of distances between vRNP pairs, virus particle sizes, and the relationship between the distances among segments and their co-presence rates, among other aspects. In contrast to the 3D models generated through cryo-EM, DNA-PAINT allows for the identification of each vRNP within each virus particle. The substantial quantity of detected virus particles may reveal underlying structural heterogeneity. Furthermore, the correlation between the pairwise distance and their co-presence rates will enable the examination of the basic hypothesis that

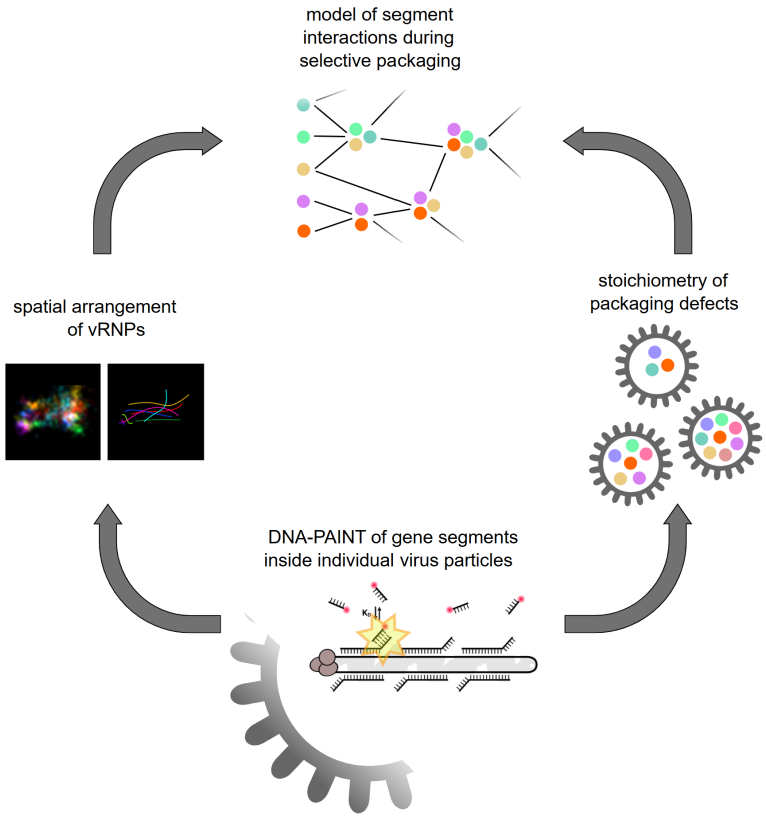

Figure 2.4: abstract of single-virion DNA-PAINT. This plot is an abstract of our methods and findings in studying IAV (Chapters 2 and 3). Spatial coordinates, super-resolution images, and vRNP combinations in incomplete virions help us further understand the genome packaging process of IAV.

stronger interactions are associated with reduced distances and heightened co-presence among segment pairs.

2.3 Astigmatic 3D calibration

The first step to apply astigmatism 3D SMLM is to measure the offset and gain of our scientific complementary metal-oxide-semiconductor (sCMOS) camera, which has read-out circuits for each pixel. We estimated them using the protocol described by Huang et al [58] and the ImageJ plug-in called GDSC SMLM [59]. The offset and read noise were calculated using the mean value and variance from a movie captured at zero exposure with 6000 frames. Under the assumption of constancy, gains were computed from 20000-frame movies of increasing exposure ranging from 20 ms to 240 ms, with increments of 20 ms. A linear regression of every

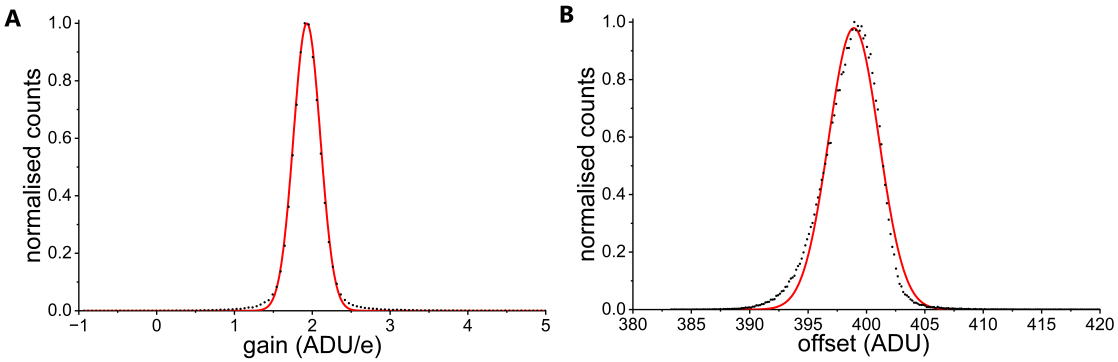

Figure 2.5: histograms of gains and offsets of sCMOS camera. The distributions of gain (A) and offset (B) are displayed. The red curves represent the results of the Gaussian fit. The peak, as well as the average values, are 2 and 400 for gain and offset, respectively.

pixel gave us the gain as the slope of the output against exposure. To be more specific, the gain was estimated using:

$$\begin{aligned} g_i &= (B_i B_i^T)^{-1} B_i A_i^T \\ A_i &= \{(v_i^1 - var_i), ..., (v_i^k - var_i), ..., (v_i^N - var_i)\} \\ B_i &= \{(\bar{D_i^1} - o_i), ...(\bar{D_i^k} - o_i), ...(\bar{D_i^N} - o_i)\} \end{aligned} \tag{2.1}$$

where i indicates different pixels, v_i^k is the variance at exposure time of k, $\bar{D_i^k}$ is the mean at exposure time of k, o_i is the offset, var_i is the variance at the zero exposure and N is the number of exposure levels. The distributions of gain and baseline are centred at approximately 2 and 400, respectively (2.5). The gain and offset parameters were tested within the ranges of 1.5 to 2.5 and 380 to 420, respectively. The coordinates of localisations were barely affected by these variations. Due to the limited impact of these parameters within the tested ranges, values of 2 and 400 were selected for the camera's gain and offset, and the parameters for Picasso were chosen accordingly.

Next, we recorded the PSFs of single Cy3B fluorophores from -300 nm to 300 nm in axial direction, with a 10-nm step size (2.6A). The biotinylated DNA oligos labelled with Cy3B were fixed on neutravidin-coated PEGylated slides. We simulated the calibration curve (plotting the widths of the PSF in the x and y directions against the axial positions) using a sixth-degree polynomial [60]. The

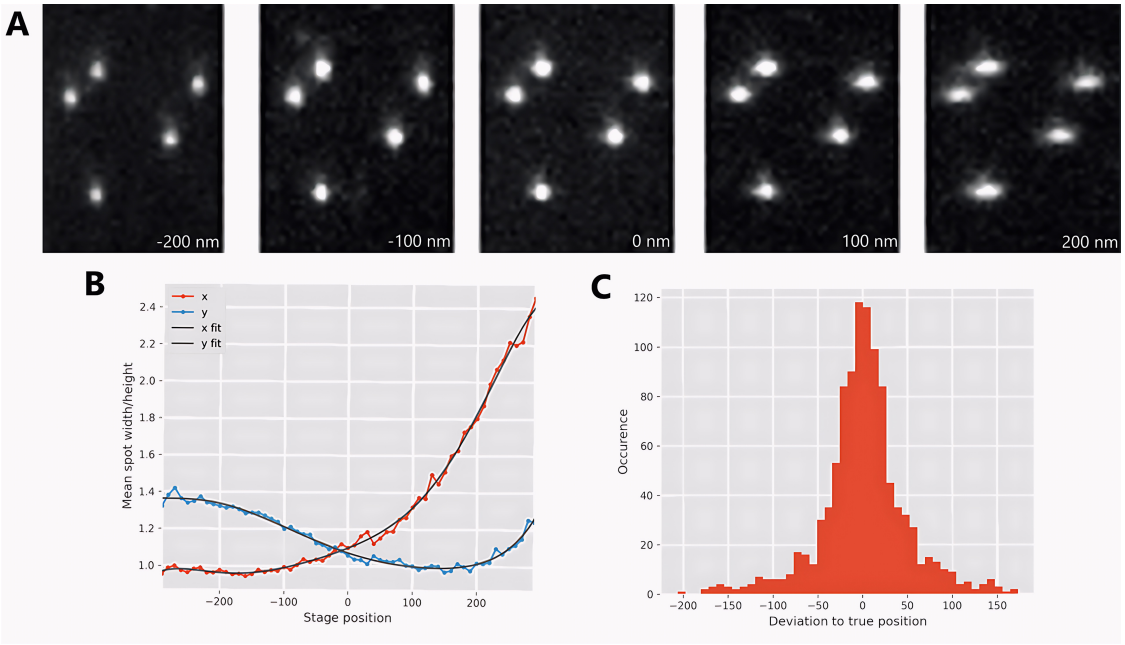

Figure 2.6: 3D astigmatic calibration. A: Examples of PSFs from -200 to +200 nm in the z-axis with a step size of 100nm. Scale bar is 1 μm B: calibration curve in the range of -300nm and +300nm with a step size of 10 nm. C: histogram of the deviation to z position whose Gaussian fit is centred at 0 with a sigma of 27 nm.

acquired calibration curve is asymmetrical and deviates from the theoretically expected shape. This deterioration occurs when the sample stage is positioned more than 150 nm from the focal plane, likely due to axial asymmetry and light loss caused by spherical aberration (2.6B) [61]. Adjusting the light path could potentially resolve this issue. However, imaging was performed using the Nanoimager, a sealed commercial system, which precludes user modification or repair. Fortunately, due to the sizes of the IAV virions, which range from 80 to 120 nm in diameter [62], we performed the 3D measurement outside of the deviated region, ensuring higher sensitivity and accuracy. The deviation between the calculated and actual z positions exhibits a broad distribution centred at zero (2.6C). The standard deviation of a Gaussian fit quantifies the localization precision in the z direction, yielding a value of approximately 27 nm.

When imaging with an oil immersion objective into an aqueous medium, the mismatch between the refractive indices of water and glass results in a focal shift. It

means the physical distance between the emitter and the coverslip surface is shorter than the distance that the objective needs to move to refocus. The net effect of refraction depends on the observation depth. However, when only small amounts of spherical aberration are present (imaging a few microns past the glass surface), this magnification can be treated as a constant [60]. We calculated the axial scaling factor using GATTA-PAINT 3D HiRes 80R Expert Line [63]. It is a nano-ruler with two fluorescent spots separated by roughly 80 nm at two ends (2.7A). We found these nano-rulers by selecting a few examples and utilising the 'Pick Similar' function in the Picasso render. Next, each nano-ruler was treated as a two-component Gaussian mixture model. The measured distance between the two centres(L_m), and the angle between the nano-ruler and the surface (θ_m), follow the equation [63]:

$$L_m = L\sqrt{\frac{1 + tan^2(\theta_m)}{1 + f^2 tan^2(\theta_m)}} \tag{2.2}$$

where f is the axial scaling factor and L is the actual length of the nano-ruler. We obtained a nano-ruler length of 77nm and a scaling factor of 0.65 (2.7E). Examples of nano-rulers are shown in the forms of a rendered super-resolution image and a scatter plot (2.7B, C). The relation between the corrected height and project length of nano-rulers resembles a quarter circle as it should be in theory (2.7D). So far, we have finished all the preparation work for 3D astigmatic SMLM.

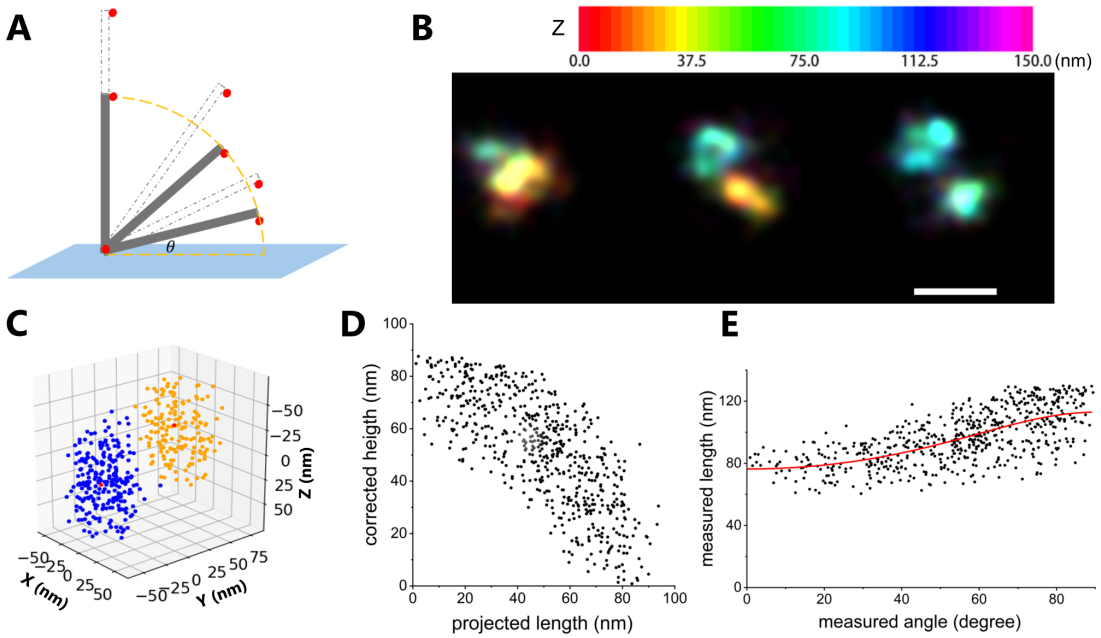

Figure 2.7: axial scaling factor determination using nano-ruler. A is a schematic representation of nano-rulers attached to a surface at various angles. Compared to the actual (grey) position, the distance between two fluorophores projected onto xy plane keeps the same, while the measured (dashed) z position needs to be corrected by a scaling factor f. B: Three examples of the nano-ruler, with the axial positions indicated in colour (scale bar measures 100 nm). C: An illustrative scatter plot displaying localisations that belong to the same nano-ruler; the top and base fluorophores are depicted as yellow and blue clusters, respectively, with their Gaussian centres highlighted by red dots. D: The projected lengths of the nano-rulers in the xy plane, along with their corrected heights. E: The relationship between the measured angles and lengths, alongside the result of the fitted function using equation 2.2.

2.4 Experiment protocol of single-virion DNA-PAINT

In terms of the spatial arrangement of vRNPs, Cryo-EM has provided high-resolution three-dimensional models; however, it does not offer the specificity needed for vRNPs of similar lengths or their structural heterogeneity. In this study, we investigate the spatial organisation of RNPs within individual IAV particles using exchange DNA-PAINT. For clarity, we have included the methods for the stoichiometry assay as well, which will be discussed in Chapter 3.

The virus sample was manufactured by Charles River Laboratories, aliquoted, snap-frozen in liquid nitrogen and stored at -80°C. Fluorescent imager strands

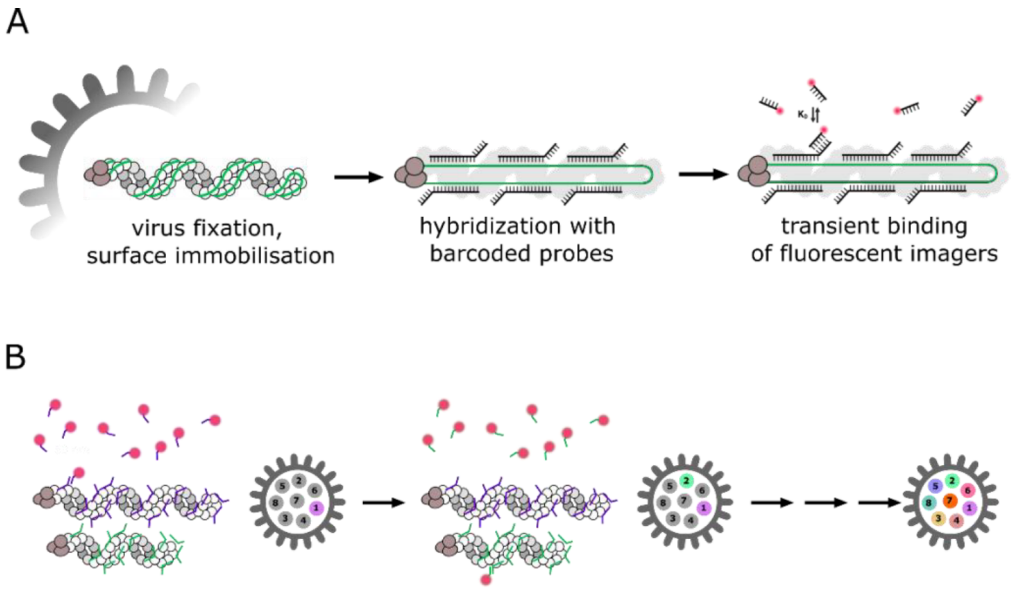

Figure 2.8: single-virion DNA-PAINT protocol. A outlines the key steps in the single-virion DNA-PAINT technique. Initially, virus particles are fixed and immobilised on a surface. They are then permeabilised and incubated with barcoded probes that contain 9nt segment-specific extensions. These extensions will transiently bind to imagers labelled with Cy3B. B illustrates two imaging rounds, each targeting a different viral vRNP using orthogonal imagers. The vRNPs are imaged sequentially until all of them have been recorded for the desired duration.

were purchased from Metabion International AG. Based on the gene sequences for PR8 (Pubmed EF467817.1 - EF467824.1, AF389122.1), the sequences for barcoded hybridisation probes on all eight RNA segments were designed using Stellaris Probe Designer 4.2 from LGC Biosearch Technologies with the following settings: Organism: Human, Masking Level: 5, Max. Number of probes: 48, oligo length: 20, Min.Spacing length: 2 (All sequences are listed in TableA.1). The docking-strand sequence features a 5'-TT-3' spacer at the 5' end, followed by a 9-nucleotide docking sequence (seen in Table 2.1, derived from Schnitzbauer et al. [5]). Cy3B is positioned at the 3' end of the imager strand to minimise the distance between fluorophores and hybridisation sites. If all target sequences are available, we have 19 to 48 sites on each segment.

Dr Christof Hepp developed the following protocol for sample preparation and

Imager	Target segment	Docking sequence (5' —>3')	Imager sequence (5' —>3')	Acquisition length (frames)		number of target sites
				Super-Res.	Segment-Stoch.	
P1	PB1	TTATACATCTA	CTAGATGTAT-Dye	4000	1000	48
P2	PB2	TTATCTACATA	TATGTAGATC-Dye	6700	1000	48
P3	NA	TTTCTTCATTA	GTAATGAAGA-Dye	7000	750	42
P4	PA	TTATGAATCTA	GTAGATTCAT-Dye	12300	900	48
P5	HA	TTTCAATGTAT	CATACATTGA-Dye	12300	1000	48
P6	M	TTTTAGGTAAA	CTTTACCTAAA-Dye	17400	2250	21
P7	NS	TTAATTGAGTA	GTACTCAATT-Dye	12300	1300	19
P8	NP	TTATGTTAATG	CCATTAACAT-Dye	8000	2250	33
P9 (control)	none	TTAATTAGGAT	CATCCTAATT-Dye	NA	NA	NA

Table 2.1: acquisition lengths and sequences for IAV segments. This table lists acquisition length, docking and imager sequences and the number of barcodes of each IAV target segment. These sequences are adapted from [5, 64] with modifications in the P6 imager.

imaging conditions for the virus DNA-PAINT. And I performed the previously described 3D astigmatic calibration and set up the data analysis pipeline which will be introduced later.

Coverslip cleaning Glass coverslips (Epredia, $24 \times 60mm$, #1.5) were cleaned using super sonication at 45kHz 100W for 15 minutes in 2% Hellmanex III (Hellma Analytics, Germany) solution. After this initial cleaning, the coverslips were rinsed five times with Milli-Q (MQ) water. They were then sonicated for an additional 5 minutes under the same ultrasonic conditions and washed five more times with MQ water to remove any residual Hellmanex. Next, the coverslips were dried using a nitrogen flow and subjected to plasma cleaning at 100 W (Henniker Plasma HPT-100) for 3 minutes. The cleaned coverslips should be used immediately for immobilisation.

Virus immobilisation, permeabilisation and hybridisation The virus aliquot was stored at -80°C and thawed on ice. The virus particles were fixed using 4% formaldehyde (purchased from Thermo Scientific, diluted in HEPES-buffered saline (HBS)) for high-throughput stoichiometry analysis or 0.5% for super-resolution experiments, incubating for 10 minutes at room temperature. The virus aliquot was diluted in 0.9% NaCl at a ratio of 1:1000 and centrifuged at 1300 rpm for 2 minutes at 4°C using SIGMA 1-14K centrifuge. 10 μL of supernatant was collected and added into a CultureWell gasket (6 mm diameter, Grace Biolabs, USA) and placed on a 38°C heat block (Grant QBA) to accelerate evaporation. Once the

sample was dry, the gasket was replaced with a flow chamber (VI0.4 ibidi, Germany). The sample was washed with HBS and permeabilised with 0.5% Triton X-100 for 15 minutes. For hybridisation, the sample was washed again with 2× sodium citrate buffer (SSC) and then incubated with hybridisation buffer (HB: 200 mg/mL tRNA, 20 mg/mL nuclease-free BSA, 2 × SSC, 10% formamide, and 1% ribonuclease inhibitors) for 10 minutes to block unspecific probe attachment. After blocking, the virus sample was incubated overnight at 37°C in HB buffer with an equal amount of 8-probe sets that target the respective gene segments at a concentration of 4 μM. Once hybridisation was complete, the sample was incubated with clearing buffer (CB:2 × SSC and 10% formamide) for at least 30 minutes to remove unbound probes.

The viral sample was prepared and mounted onto the microscope, followed by a wash with 2× SSC. Nanodiamonds (Cytodiagnostics, Canada), measuring 90 nm in diameter, were diluted at a 1:10 ratio in 2× SSC and then introduced into the chamber as fiducial markers. Incubation continued until approximately 20 fiducial markers were detected within the field of view (FOV). Next, the sample was rinsed with a DNA-PAINT buffer (DB: 5 mM Tris-HCl pH 8, 75 mM $MgCl_2$, 1 mM EDTA, and 0.05% Tween-20) [65]. The imagers were diluted to the desired concentration in DB. The imaging duration was adjusted to achieve a comparable average number of binding events for all segments. A comprehensive summary of the varying imaging conditions is presented in Table 2.2. After each imaging round, the chamber was rinsed with the DB and allowed to stand for 2 minutes to eliminate any residual imaging agents. This imaging and rinsing procedure was repeated until all sequences had been imaged the required number of times. All experiments were conducted using a Nanoimager (Oxford Nanoimaging) equipped with a 100× objective (NA 1.4). The FOV measured approximately $80 \times 49\,\mu m^2$. For movie acquisition, a 532nm laser with an intensity of 3.6 mW was used, employing TIRF mode at an illumination angle of 54.5°.

		super-resolution	segment stoichiometry
Imager concentration		5 nM	50 nM
Exposure time		200 ms	20 ms
Total imaging time	frames	4000 -17400	750-2250
	minutes	13-58	0.25-0.75
Sample size (FOVs)		1	49

Table 2.2: imaging conditions for single-virion assays. This table lists imaging conditions for super-resolution and high-throughput stoichiometry experiments, including imager concentration, exposure time, number of frames and FOVs.

PEG surface preparation To acquire the calibration curve for 3D astigmatic microscopy, single imagers were tethered to a PEG surface using a NeutrAvidin link and immersed in IB supplemented with a Gloxy-based oxygen scavenging system (1 mM Trolox, 1% glucose, 40 μg/mL catalase, and 0.1 mg/mL glucose oxidase). The PSFs from isolated fluorophores were recorded from -300 nm to +300 nm with a 10-nm step size.

Microscope slides were prepared following the methods described in previous studies [66, 67]. In summary, coverslips were cleaned using a plasma treatment and then sequentially treated with 2% Vectabond aminosilane (Vector Labs) in acetone. After drying under nitrogen, these coverslips were bonded to 6 mm silicone gaskets (GBL103280; Grace Bio-Labs), resulting in observation wells made of aminosilane-functionalized glass. Next, the wells were coated with 20 μL of a solution containing 30 mM mPEG-SVA (Laysan Bio) and 0.75 mM Biotin-PEG-SVA (Laysan Bio) in MOPS-NaOH (pH 7.5) for 90 minutes at room temperature. After this, the wells were washed with PBS and treated with 30 μL of a 10 μM NeutrAvidin (ThermoFisher) solution in 0.5x PBS for 10 minutes at room temperature. Following additional washes with PBS, the wells featured NeutrAvidin-biotin-PEG/mPEG-functionalized glass floors.

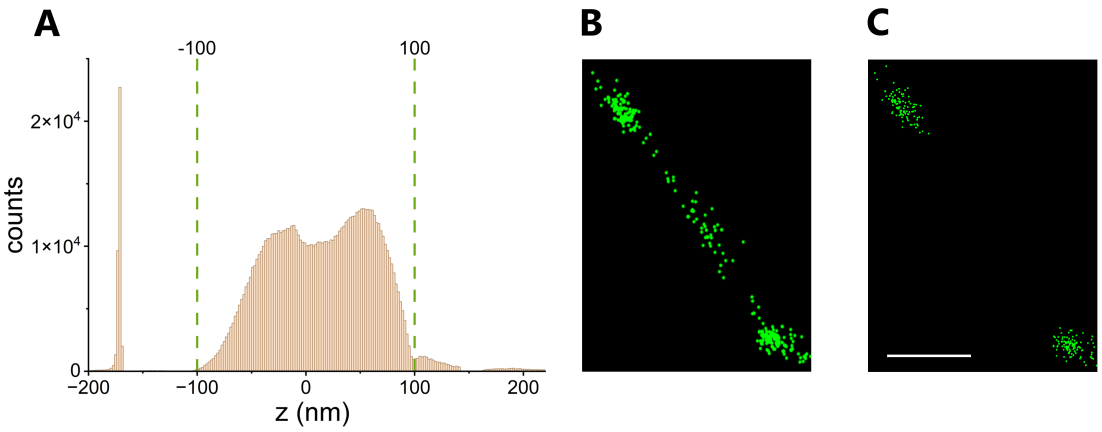

Figure 2.9: suboptimal localisation filtering based on z-axis position. This graph shows the histogram of axial position of NA segments (A) and an example of localisations from NA segment recording before (B) and after (C) filtering. The scale bar is 200nm. The green dashed lines indicate the thresholds.

2.5 Preprocessing for SMLM data

After localisations using Picasso (Box side length: 7, Min.net gradient: 3000, Baseline: 400, Sensitivity: 2.5, Quantum efficiency: 0.82, Pixel size: 117 nm, magnification factor: 0.65), we filtered the localisations based on their z positions, which is essentially based on the shape of the PSFs. When two imagers in proximity are detected simultaneously, their overlapped PSFs will be fitted as a single elongated PSF by the localisation algorithm. The elongation of PSFs is indicated by deviations in the z positions, and these localisations were filtered out accordingly (2.9).

We utilised the AIM method [12] for drift correction(segmentation: 100, interaction distance: 20 nm, maximum drift in the segment: 60 nm). Afterwards, we registered the localisations from different imaging rounds using fiducial markers. Since these fiducial markers do not form specific linkages with the surface, some of them are mobile during the recording or buffer exchange process. To address this, we implemented a two-step procedure to identify the stationary markers before using them to register the localisations from various recordings.

First, we filtered out fiducial markers that exhibited movement during the recording process by calculating the average Euclidean distance of each marker's localisations from its centre and applying a threshold of 12 nm. Next, we excluded fiducial markers whose positions changed significantly during buffer exchange. The positional shifts of different fiducial markers between two consecutively recorded movies should be similar. To identify the outliers, we calculated the squared Mahalanobis distances of these shifts. The squared Mahalanobis distance is defined as

$$d^2_{Mahalanobis} = (x - \mu)^T \cdot C^{-1} \cdot (x - \mu) \tag{2.3}$$

where x is the position vector of the object, μ is the arithmetic mean vector, and C^{-1} is the inverse covariance matrix of the independent variables. It measures the distance between a point and a probability distribution. Essentially, it quantifies the distance of a vector from the mean, standardised by the covariance matrix. As a result, the Mahalanobis distance is unitless, scale-invariant, and takes into account the correlations within the dataset. Fiducial markers with squared Mahalanobis distances greater than the 0.68 quantile of a chi-squared distribution were considered mobile during buffer exchange.

After identifying the static fiducial markers, the transformation matrix between two consecutive movies is calculated using the point-to-point iterative closest point (ICP) algorithm [68, 69]. In general, ICP iterates through two steps: first, finding a correspondence set $K = (p, q)$ from the target point cloud P, and source point cloud Q transformed with the current transformation matrix T; second, updating the transformation T by minimising an objective function E(T) defined over the correspondence set K. The objective function applied here is

$$E(T) = \sum_{(p,q)\in K} ||p - Tq|| \tag{2.4}$$

We selected point-to-point ICP to directly handle 3D coordinates while restricting the matrix T to a rigid-body transformation. To estimate the uncertainty of this registration method, we randomly divided the static fiducial markers into two

subsets for registration and compared the results. The differences observed were negligible (2.10A). After registration, the localisations were linked to prevent bias arising from various dwell times. Localisations that were separated by less than one dark frame in time and less than one pixel in space were considered from the same binding events. These two parameters are somewhat arbitrary due to the sparse nature of the data. We calculated the empirical localisation precision as the standard deviation of the coordinates among these linked localisations (2.10B).

The viral segments were detected using DBSCAN, which defines the clusters as dense regions separated by space with low-density noise. It works around three key concepts: core samples (samples have a minimum number (min_samples) of other samples within a specific distance eps), border samples (samples within the radius of eps of a core sample but do not have min_samples neighbours), and noise samples (samples are neither core nor border samples). A cluster is therefore a set of core samples and their border samples. In practice, DBSCAN takes two parameters: min_samples and eps to do the analysis. After adjusting the recording time for each segment, the average localisations from different movies are similar, which allows us to set the default values: eps = 58.5nm (half pixel), min_samples = 25 for the super-resolution assays. Next, we used the centroids of clusters to present the position of vRNPs. To assess the reliability of the centroid positions, we randomly divided the localisations in each cluster into two sub-clusters and measured the distances between their centroids (2.10C). The median value of this distribution is 4.6 nm, which is considerably smaller compared to the IAV particle size.

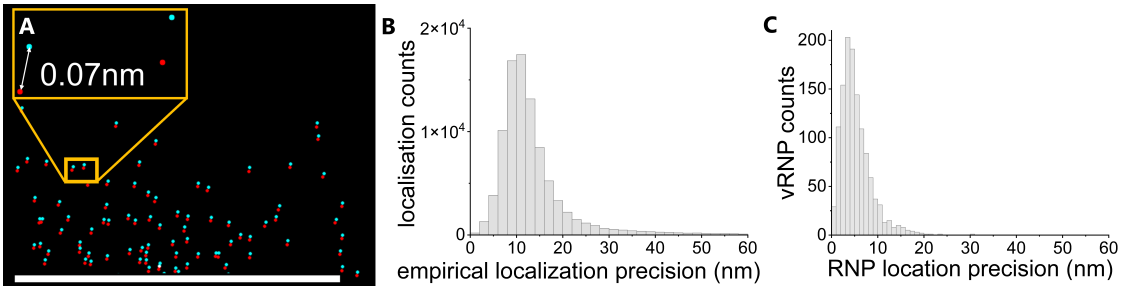

Figure 2.10: precisions of registration, localization, and vRNP position. A: Part of a PB2 segment aligned using two disjoint subsets of fiducial markers (Red and cyan). It shows the robustness of our channel registration method. The difference between the alignment results is less than 0.1nm. The scale bar is 5 nm. B: Histogram of empirical localisation precision, which is the standard deviation of localisations from the same binding events. C: This histogram shows the precision of the vRNP position. It is the distribution of distances between centroids of two randomly split subsets of localisations from the same segments.

2.6 Structure integrity of virions

Before delving into the details of the vRNPs, we conducted several assessments to ensure the structural integrity of the virus particles. First, we employed an anti-HA immunofluorescence (IF) co-staining method to visualise the virus envelope, followed by a DNA-PAINT assay targeting the NA segment(2.11A). The immobilised IAVs were incubated with rabbit anti-HA IgG as the primary antibodies and goat anti-rabbit IgG as the secondary antibodies. In the negative control, we substituted the primary antibodies with non-immunised rabbit IgG, which showed no fluorescence signals (2.11 C).

The sample showed more IF spots without RNA signal than RNA spots without IF signal, which could be explained by free HA or virus particles lacking the NA segment. This step is to ascertain that we are observing the whole virus particles (i.e., not groups of segments dissociated from the particles). Thus, we are more interested in the percentage of vRNPs that are associated with envelop than HA associated with vRNPs. Our results indicated that 73% of the particles exhibiting an NA signal also showed an IF signal (2.11B). Although this does not provide definitive proof of the intactness of the virions, it does show that the majority of the

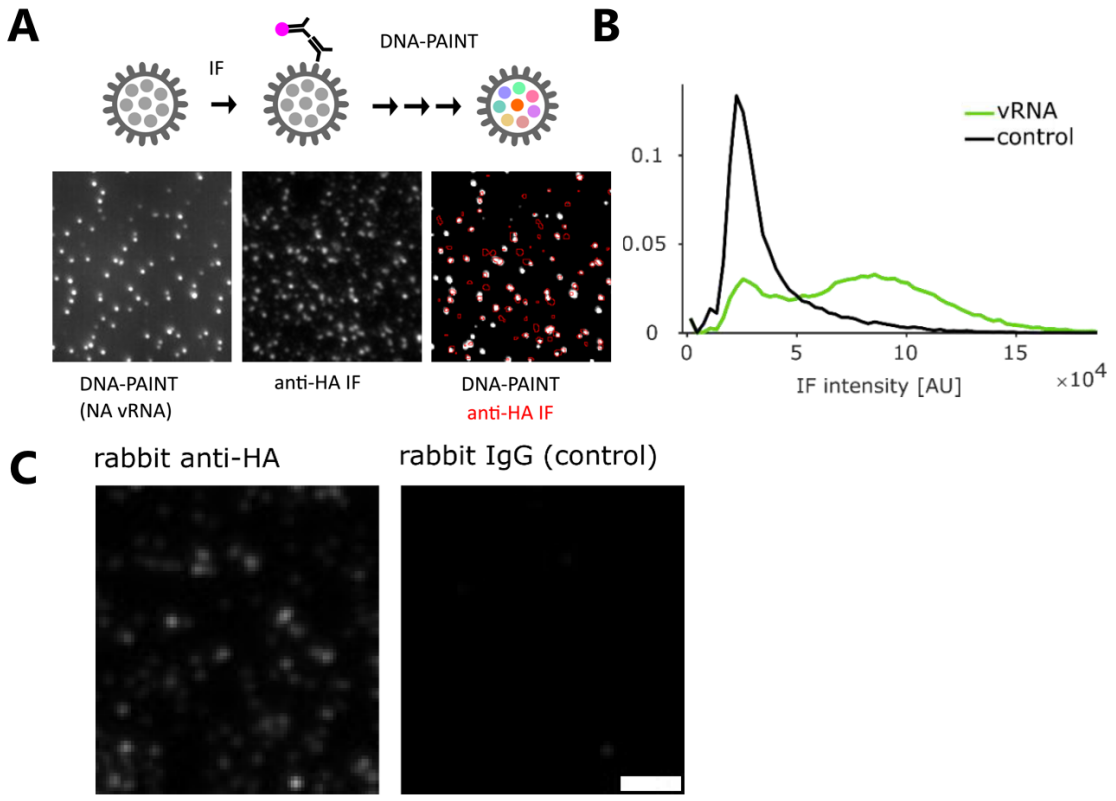

Figure 2.11: co-localisation of anti-HA IF signal and NA segment. A: Experiment to demonstrate co-localisation of vRNA and the viral envelope: Individual viral envelopes were stained via anti-HA IF, followed by image acquisition and DNA PAINT targeting NA segments. vRNA (left panel) and HA (middle panel) showed extensive co-localisation (right panel). The left panel in A is a single frame from the PAINT movie. Thus, it does not show all signals from NA vRNPs. The co-localisation is indicated by the red circles in the right panel. B: Distributions of HA IF intensity detected in positions with vRNA (green) and random positions in a control experiment (black). C: Comparison of the signals from the IF assay and the control experiment. The scale bar is 2μm.

virus samples contained the envelope component, HA. Additionally, the distribution of segment counts (which will be discussed later) supports the conclusion that the RNP assemblies we observed correspond to virus particles.

Next, we investigated the conformation and structural integrity of vRNPs using different barcodes of primary probes to image the 3' and 5' termini and the outline of the NA vRNP. To identify the terminus signal, we modified the DBSCAN parameters to eps = 10nm and min_samples = 4. Except for segments that are not elongated enough to assign the termini to one end, only about 17.4% (75 out

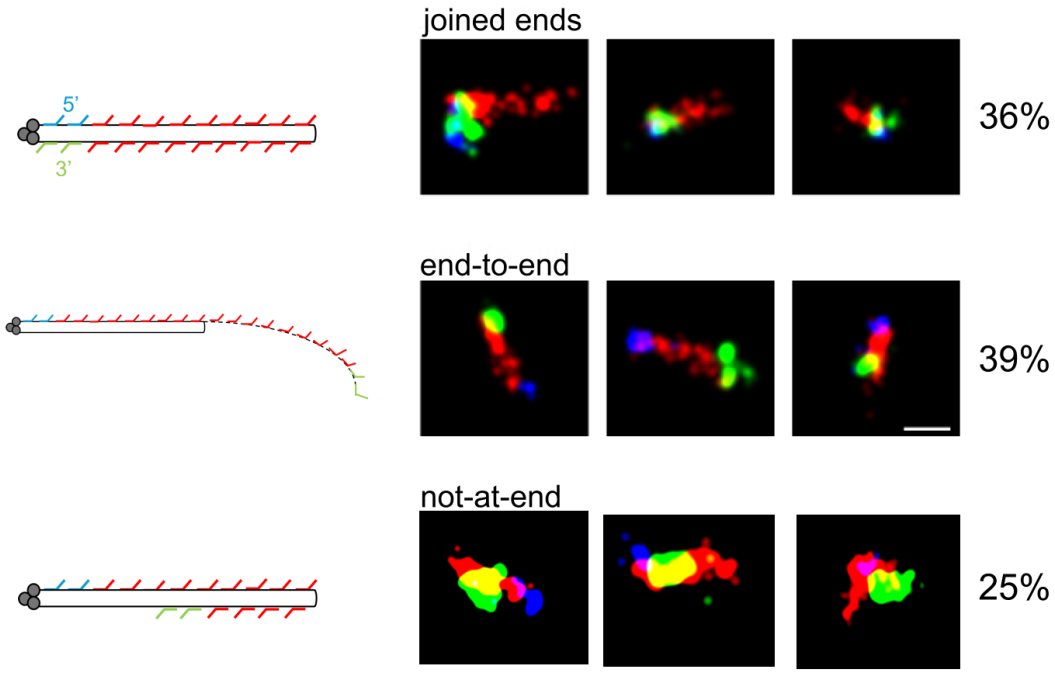

Figure 2.12: quaternary structure of NA segments. In the observed cases that both 5' (blue) and 3' (green) termini are identified, 75% of them had termini at one or both ends of the outline (red) of the NA segment. The scale bar is 50 nm.

of 432) of the viral particles have signals from both ends. This may be caused by the inaccessibility of the two ends when they bind to RNAPs. Among the NA segments that show strong signals from both ends, there is a similar probability of the termini being together (36% of the 75 segments) or at the opposite ends (39% of the 75 segments). The remaining 25% had at least one signal not clearly located at an end (2.12). We also found that the length of NA segments with termini on opposite sides is approximately twice as long as those with termini on the same side. This observation suggests that one end of the RNA termini may detach from the polymerase and unwind from the nuclear proteins of the vRNPs. This open structure may serve dynamic functions, such as replication and transcription, for example, allowing the 3' end to move between different binding sites on RNAP [70]. This type of open vRNPs accounts for approximately 6.7% of the total vRNPs observed and thus has a limited influence on the subsequent statistical analysis.

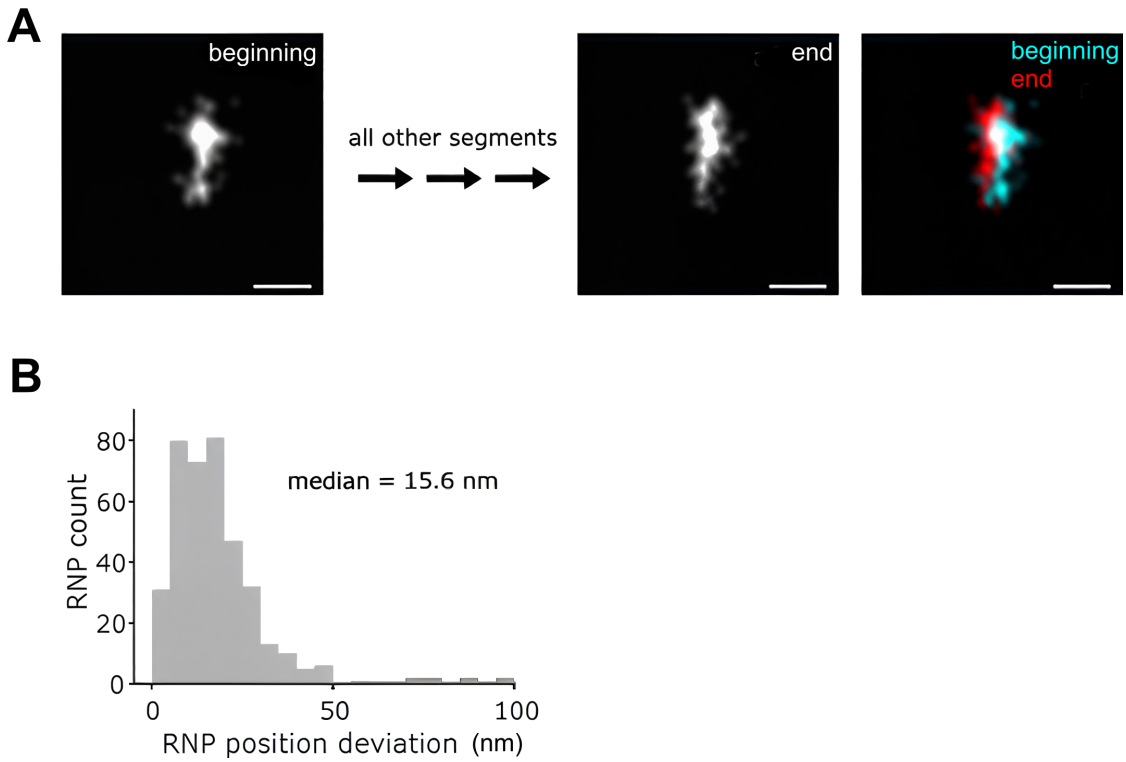

Figure 2.13: positional change of NA segments during imaging. A: Example of a super-resolved NA vRNP at the beginning and the end of the imaging cycle. The overlap on the right illustrates the positional shift. Scale bar is 50 nm. B: Histogram of the mean shifts in location of all NA vRNPs that could be detected both before and after imaging.

Finally, we confirmed the vRNPs remained in their positions through the experiment by recording the NA vRNPs at the beginning and end of the imaging sequence. The median value of the position difference between the beginning and end is 15.6nm, which is a small fraction compared to the inter-segment distances (2.13). So, it only has a limited influence on our later distance analysis. The relative positional changes among vRNPs differ, suggesting that rearrangement occurs during imaging and highlighting the need to optimize the fixation step in sample preparation.

2.7 Distance analysis of vRNPs

Following the detection of segments, a co-localisation analysis was performed using the centroids of each identified cluster. The coordinates of these centroids were subject to an agglomerative hierarchical clustering method, employing single linkage

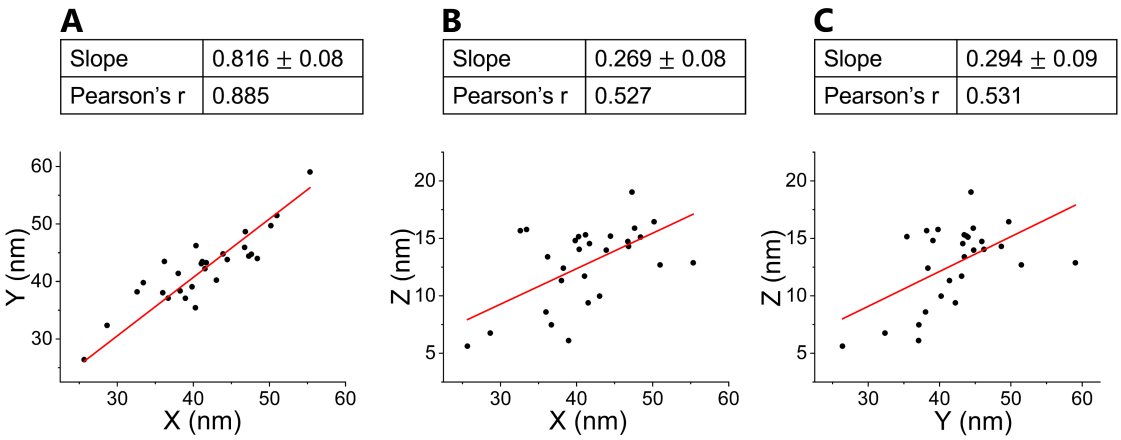

Figure 2.14: correlation of inter-segment distance between different directions. The plots show the correlations of inter-segment distances projected onto different directions with the results of a linear fit (red lines) dictated at the top. The moderate correlation proves the validity of our 3D astigmatic method.

with a distance threshold of 2 pixels (equivalent to 234 nm). Within this context, the co-localised vRNPs were classified together as belonging to the same virion. Agglomerative clustering is a bottom-up approach that begins with each sample as an individual cluster. Clusters are then successively merged if their distance is below the threshold. The hierarchical structure of these clusters is represented as a dendrogram, with the root node encompassing all samples. Standard linkage criteria utilised in hierarchical clustering include: Ward's method (minimum variance of the clusters being combined), the maximum method (maximum distances between all observations in the two clusters), the average method (the mean distance of all observations from the two sets), and single linkage (minimum distance between observations in the compared clusters).

Analysis of vRNP distances within the virus particles demonstrated a moderate correlation of the lateral and axial directions (2.14), which proves the validity of our 3D method. The average vRNP distances in z were only about one-third of the x/y distances, suggesting that either the virus particles may have been slightly flattened during the preparation process, or that the fraction of elongated particles preferentially aligned with the surface.

We used the maximum segment-centre distance to present the size of virus particles and observed that the size of virus particles reached a maximum when the segment number was six and decreased when the virus was even more complete (2.15C). One possibility is that the interactions within a complete or nearly complete virus particle can force the vRNPs into a more organised configuration. The median segment-centre distance of HA is significantly closer to the centre (2.15B). However, among virions with all segments, no one segment was constantly in the closest proximity to the centre. We also examined the frequency of segment-centre distances in the HA (the segment closest to the centre on average) and NS (the segment furthest from the centre on average) segments. Both segments share the same range, but the key difference lies in the frequency of small distances to the centre (2.15D).

Next, we examined the pairwise distances between segments. The median distances of segment pairs are illustrated in Figure 2.15A. One can see that the maximum distance is roughly the same size as a virion (100 nm), and the minimum distance (45 nm) is about twice the diameter of vRNP, leaving a 20 nm gap between two vRNPs. Additionally, the frequency distribution of the nearest segment pairs (NP-NS) and the furthest segment pairs (PB1-NS) is presented in Figure 2.15F. Similar to the segment-centre distances, the two inter-segment distances fall within the same range; however, the frequency decreases much more rapidly as the distance of NP-NS increases. Additionally, there is a moderate negative correlation with a Pearson's coefficient of -0.54 between the co-appearance rate of two segments and the distance between them in virus particles with 2-4 segments (2.15E). The negative correlation also exists if we expand the virions under investigation to other numbers of segments, such as 2-5 and 2-6, with slightly smaller absolute values of Pearson's coefficient. This supports our hypothesis that segment pairs with stronger interactions have closer distances and higher co-presence rates.

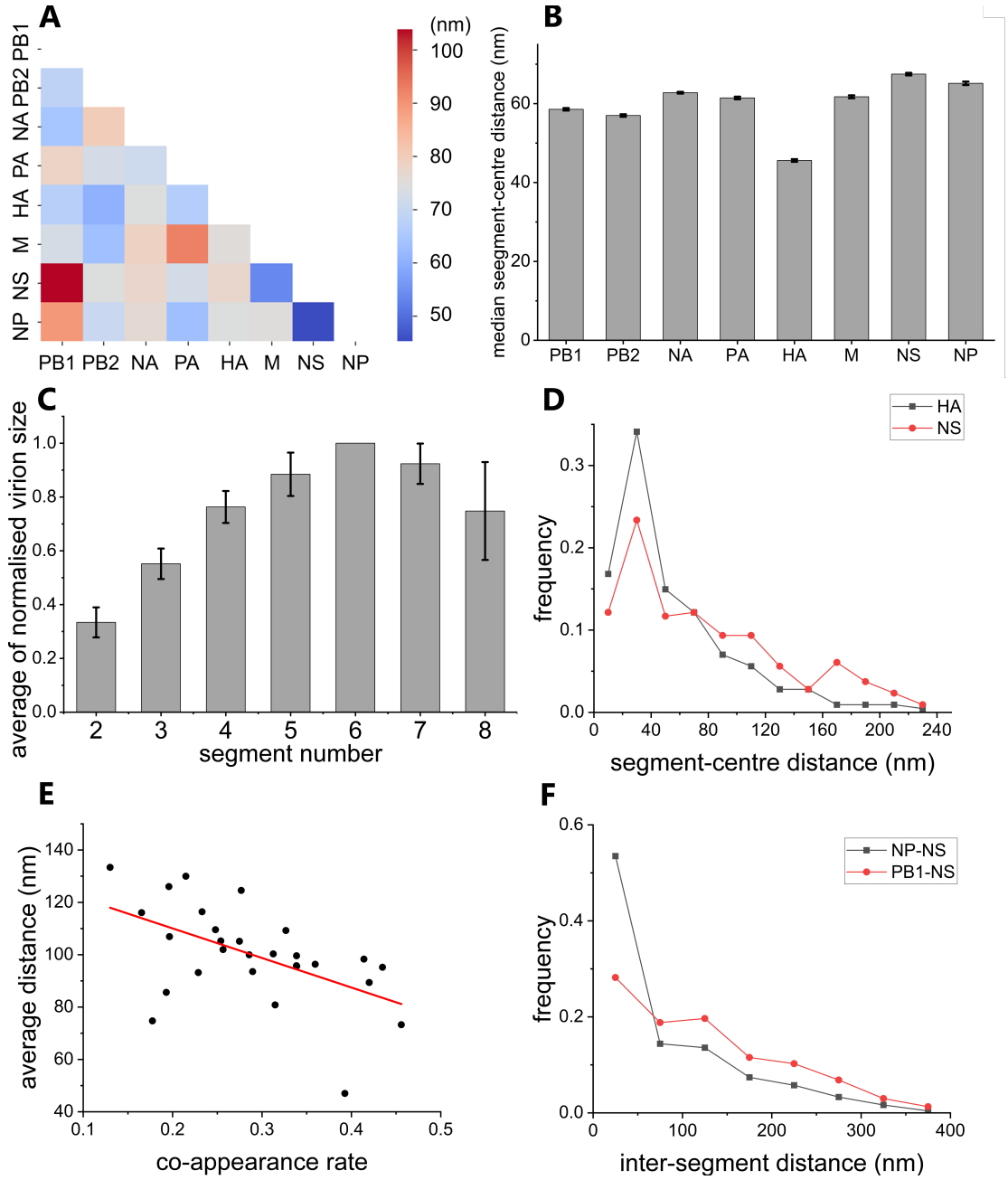

Figure 2.15: analysis of distances between vRNPs. A: A heatmap illustrating the median inter-segment distances measured in nanometres (nm). B: The median distances between the segments to the virion centre for eight vRNPs. C: The average size of the virion, indicated by the maximum segment-virion centre distances, changes with the number of segments. D: A graph displaying the frequency versus distance of the nearest (HA) and furthest (NS) segments to the virion centre. E: The correlation between inter-segment distances and their co-appearance rates in virions containing 2 to 4 segments shows a moderate negative correlation (as indicated by the red line representing a linear fit), with a Pearson's coefficient of -0.54. F: The frequency distribution of the closest pair of segments (NP-NS) and the farthest pair (PB1-NS) based on their distances.

2.8 vRNP Visualisation

2.8.1 2D super-resolution image rendering and skeletonisation

Once all gene segments had been assigned to a virion, the x and y coordinates of the localisations in each segment were used to fit the vRNP spine using the following procedure: a super-resolution image with a pixel size of 0.234 nm (500X oversampling) was created by rendering each localisation as a Gaussian function with a sigma of 3.9 nm. The resulting image was subjected to a low-pass order-one Butterworth filter with a cut-off frequency of 0.02 and a Gaussian blur with a sigma of 5.85 nm and converted to binary using the ISODATA method [71]. A Hilditch skeleton [72] was fitted to the binary image (2.16).

The squared low-pass Butterworth filter here is given by the following expression [73]:

$$H_{low}(f) = \frac{1}{1 + \left(\frac{f}{cf_s}\right)^{2n}} \tag{2.5}$$

where $f = \sqrt{\sum_{d=0}^{ndim} f_d^2}$ is the absolute value of spatial frequency (f_d is the d-th component of spatial-frequency vector–one per image dimension), f_s is the sampling frequency, and n is the filter order. c is the cut-off frequency ratio defined in terms of f_s, determining the position of the cut-off relative to the shape of the fast Fourier transform (FFT). The FFT spectrum covers the Nyquist range ($[-f_s/2, f_s/2]$).

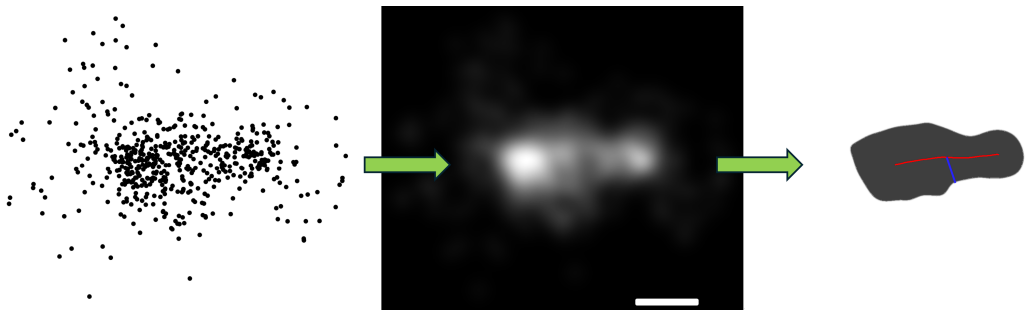

Figure 2.16: 2D image rendering process and skeletonisation. Scatter plots of localisations, the rendered super-resolution image, and the fitted spine (red) and radius (blue) of the binary image are displayed from left to right. These three plots are at the same scale, and the size of the scale bar is 50nm.

Thus c should have a value between 0 and 0.5.

ISODATA is a histogram-based method, also known as the Ridler-Calvard method. In ISODATA, an initial threshold (typically set at the average intensity of the image) is used to divide the histogram into two classes: A and B. The average intensities of these classes are denoted as m_A and m_B, respectively. A new threshold is determined as the average of m_A and m_B. The new threshold is updated iteratively based on the new average intensities until it converges.

Regarding skeletonisation, some general requirements must be met. The skeleton of binary pattern P should consist of thin lines that are one-pixel wide, lie along the centre of P, and have the same connectivity as the original pattern. Pixels are usually removed gradually. Once a skeleton or part of it is obtained after multiple passes, it should not be eroded by subsequent passes. Hilditch algorithm finds the skeleton by computing quasi-Euclidean distances and tests Hilditch conditions to preserve topology. It is a parallel thinning algorithm that uses 3×3 window (Hilditch also published a 4×4 version in a seminar in 1968) to determine whether a pixel P_1 should be removed or kept in the image. For deleting a pixel, the Hilditch algorithm checks whether the following four conditions are satisfied: The algorithm

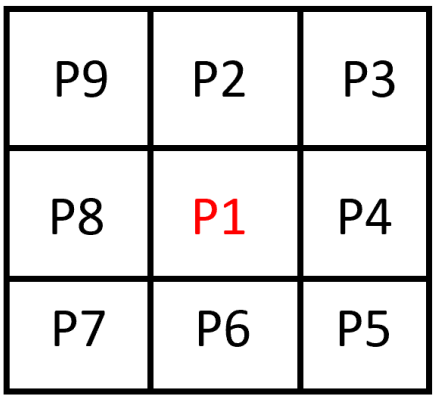

Figure 2.17: A 3×3 window that shows the arrangement of 8 neighbourhoods of pixel $P1$ that are used to decide whether $P1$ will be removed.

- $2 \leq B(P1) \leq 6$
- $A(P1) = 1$
- $P2 \cdot P4 \cdot P8 = 0$ or $A(P2) \neq 1$
- $P2 \cdot P4 \cdot P6 = 0$ or $A(P4) \neq 1$

where $B(P1)$ is number of non-zero neighbours of P1 and A(P1) is number of 0, 1 patterns in the sequence P2, P3, P4, P5, P6, P7, P8, P9, P2.

stops if no pixel is deleted at the end of a pass.

Blurring processes aim to create a continuous 2D structure for each segment, i.e. preventing isolated parts in the binary image and branches in the skeleton. However, these suboptimal situations do occur occasionally. In which cases, we used the convex hull of the detached components in binary images for skeletonisation. If branches exist, the longest one was used as the vRNP spine.

The length of the spine was considered to be the length of the vRNP, and the median value of distances from each pixel on the spine to the boundary of the binary image is considered as the radius of the vRNP (2.16). In addition, the orientation of the segment in 2D was determined by fitting the skeleton with a straight line and calculating its angle with respect to the x-axis. To measure the alignment of the 2D projections, we used the direction of maximum variance of all localisations in a virion to represent the overall orientation of the genome and calculated the angle deviation between each spine and the cardinal direction. Most projected length (701 out of 726) ranges from 0 to 100nm, which is consistent with the previously reported vRNP size. The longer segment (close to 150nm or more, 13 out of 726) may correspond to the detached vRNA we presented earlier. Moreover, most radii fall within a small range of 10-20nm, as one may expect from the projections of a rod-shaped object. These very short spine lengths correspond to a nearly circular 2D vRNP projection (2.18D) and could be from a segment perpendicular to the surface or suffer from incomplete labelling. In the most frequent cases, segments are aligned in the same direction to a certain extent, while the angle deviation can be anything from parallel to perpendicular (2.18E).

To test the accuracy of our spine-based visualisation method, we randomly split locations belonging to the same RNP and constructed the spine in the same way. The differences in spine length and orientation in the 2D plane were calculated (2.18A, B, C). We determined a median precision of 4nm, 10nm and 10^{o} for the

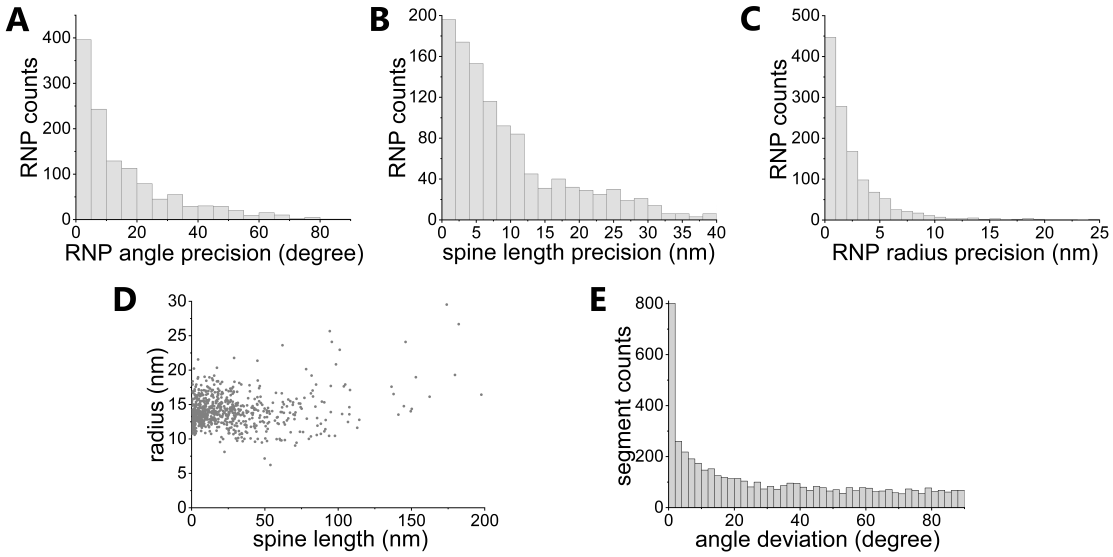

Figure 2.18: precisions and distributions of spine lengths, radii and relative orientation of vRNPs. The top plots illustrate the robustness of our pipeline based on constructed super-resolution 2D images, showing the precision of projected orientation (A), vRNP length (B), and radius (C). D is the scatter plot of spine length vs. radius of the NA segment, as one may expect that the 2D lengths of the segment distribute in an extensive range due to the various projection angles. E is the angle deviation of each segment from the cardinal angle.

vRNP position, length and orientation, respectively.

We observed three types of vRNP arrangements: 1. a global shape with short RNPs that barely overlap (160 out of 592); 2. well-aligned elongated vRNPs (192 out of 592); 3. showing a certain level of disorder (240 out of 592) (2.19). These arrangements are also observed in electron microscopy studies [74, 75]. Furthermore, the disordered arrangement is the most abundant in our dataset, which also aligns with the previous cryo-electron tomography study [74].

2.8.2 Volume rendering

In some cases, it is beneficial to create a conventional voxel-based image from a 3D coordinate dataset. There are various methods to achieve this, such as using a histogram of localisation positions or summing Gaussian functions for each localisation. These methods are essentially 3D versions of standard 2D image rendering techniques. Additionally, other volume rendering approaches utilise

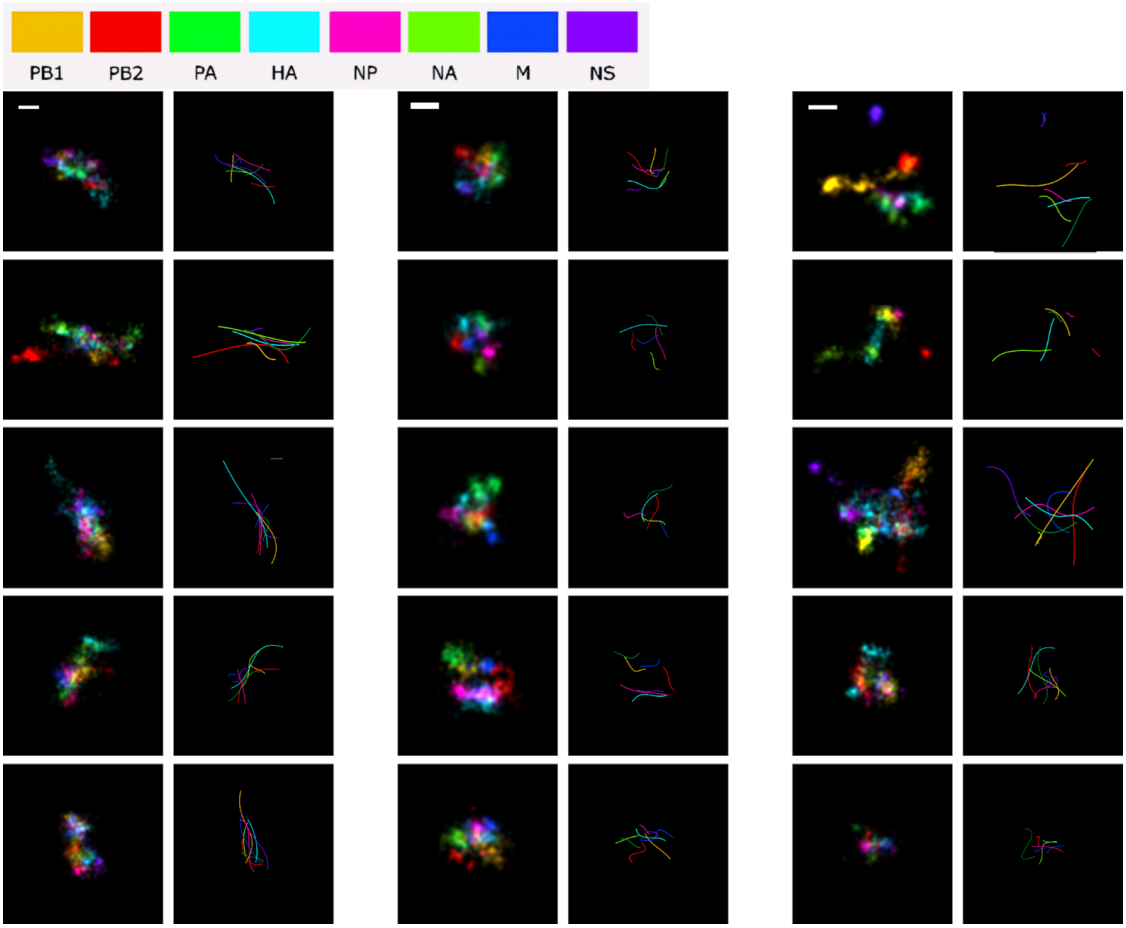

Figure 2.19: examples of constructed 2D images of vRNP complexes and their spines. The graph gives a few examples of the observed spatial arrangements and the segment spines. Different colours mark different segments within the virus particles. The three types of arrangements: elongated parallel vRNPs (left), Short separated vRNPs (middle), and disordered arrangement with no apparent consistent orientation (right). The scale bar is 50nm.

coordinates directly, such as Poisson surface reconstruction and tetrahedralization.

We performed volume rendering by extracting an iso-surface using the Python Microscopy Environment [76]. The process begins by placing points into an octree, which is truncated to a minimum of four localisations per cell. This method effectively divides the volume into cubic cells that adapt in size based on the local point density. As a result, cells are larger in areas with fewer localisations and smaller in regions with higher densities. The outcome is a volumetric data structure that retains the same information as a fully sampled reconstruction but contains

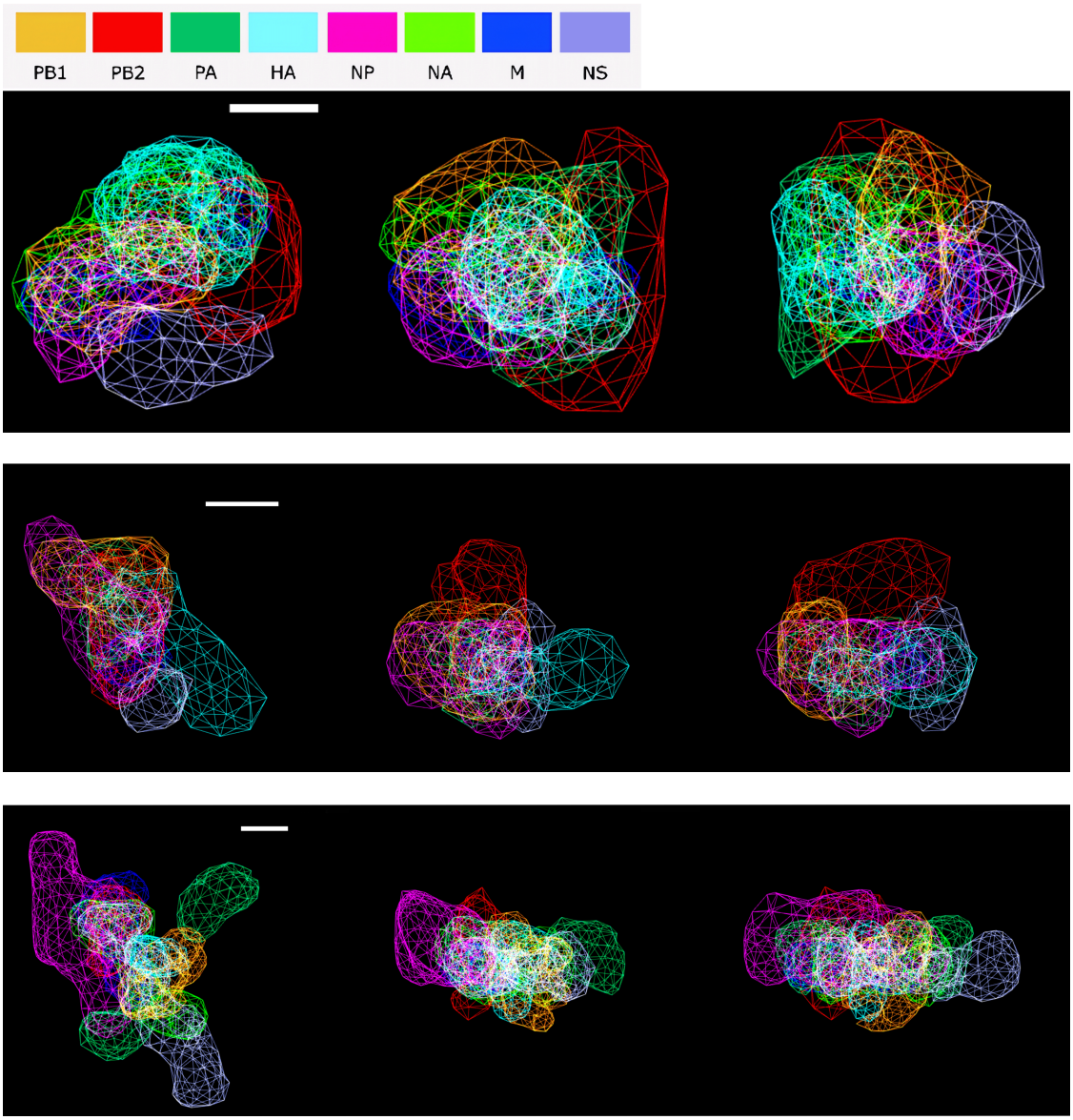

Figure 2.20: volume renderings of representative vRNP complexes. From top to bottom are three examples of iso-surfaces generated using the dual marching algorithm, belonging to classes 1, 2, and 3, respectively. The top, front, and side views of each complex are displayed in each row from left to right. All scale bars are 50 nm.

significantly fewer elements. After calculating the local density of localisations within each cell, the dual marching cubes algorithm [77] was employed with a density threshold of 2×10^{-5} localisations per cubic nanometer. This threshold was determined based on empirical localisation precision. The iso-surfaces are visualised as a wireframe in Figure 2.20.

In addition, we also realised that some techniques which are usually applied to cryo-EM data have been used in SMLM [78, 79], despite the fundamental differences in origins of signal between cryo-EM and SMLM. We followed the work of Sieben et al. [80] using single-particle reconstruction methods to create 3D models for individual segments. Briefly speaking, super-resolution images were aligned and classified by a template-free maximum-likelihood multi-reference refinement algorithm [81]. Then class averages were used to build an initial model using Eman2 [82]. The initial model was then refined using the projection matching (Xmipp3) protocol [83]. However, due to the flexibility and heterogeneity of our structures, lower symmetry order and limited data size, the 3D model we acquired lacks details and does not provide resolution enhancement.

3
High-throughput DNA-PAINT

In Chapter 3, we will use a high-throughput stoichiometry assay to analyse the vRNP combinations from over 60,000 IAV particles. This data will help us illustrate the relationships among different segment combinations, leading to an understanding of the assembly process of IAV's segmented genome. Additionally, we will introduce fluorogenic probes that reduce background interference from diffusing imagers, allowing a higher concentration of imagers and a faster imaging process.

Due to the limitations of the FOV size and the requirement to prevent overlap of nearby virus particles, we typically observed only a few hundred virus particles per FOV using the previous super-resolution method. In addition to the long imaging time, it is impractical to collect data from more than 10,000 virus particles using the exchange DNA-PAINT method we established in Chapter 2. To increase the data size, we increased the imager concentration from 5nM to 50nM, reduced the exposure time to 20 ms and the number of frames to a few hundred (2.2). The high imager concentration led to simultaneous binding events, subsequent PSF overlap, and a reduction in localisation precision. As a consequence, we lost the geometric information of the vRNPs. However, this approach allowed us to collect numerous approximate positions rapidly (3.5 min per FOV), with adequate precision to conduct co-localisation analysis, which enables us to extract segment combinations from a

large number of virions in a short period. We collected 49 FOVs per assay using the multiple FOV collection in the Nanoimager (Oxford Nanoimaging, UK) under TIRF mode at an illumination angle of 54.5° with a laser intensity of 3.6mW at 532nm.

3.1 IAV genome assembly pathway

Localisations are identified using Picasso with the following parameters: box side length of 7, minimum net gradient of 3000, baseline of 400, sensitivity of 2.5, quantum efficiency of 0.82, and pixel size of 117 nm. The magnification factor is not specified here, which differs from our previous super-resolution method, as we are now working with 2D SMLM. Channel registration was performed using the RCC method in Picasso Render. Since we are acquiring only 2D data, and considering the minimal rotation among different imaging rounds and limited localisation precision, the RCC method, which only corrects for translation, is adequate for our purposes. Additionally, due to the short recording time, drift is negligible; therefore, drift correction is omitted in the analysis pipeline.

We identified potential vRNPs using the DBSCAN algorithm (with parameters eps = 117 nm and min_samples = 4) with a more efficient implementation [84] to handle the large data size. This fast parallel version of DBSCAN is designed explicitly for low-dimensional Euclidean spaces. For the exact 2D DBSCAN, the algorithm utilises a grid construction method for point partitioning and employs bichromatic closest pairs to assess connectivity among core points. We set very loose conditions for DBSCAN to include clusters associated with unspecified bindings. This helps us understand the features of false-positive clusters in the subsequent analysis.

3.1.1 False-positive vRNP filtering using change point detection

There are two types of noise associated with high imager concentration: blurred background from diffusing imagers and non-specific binding. PSFs are approximately 5 pixels in size, so the window size needs to be significantly larger than this to

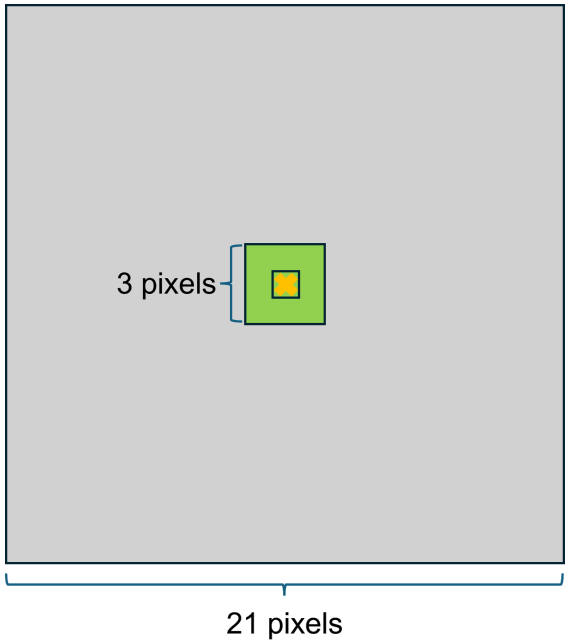

Figure 3.1: background remove method for single-virion stoichiometry assay. The yellow symbol X denotes the pixel where the centroid of a cluster is located. The raw signal and background are determined by calculating the sum and median value within a 3×3 (green area) and a 21×21 (grey area) window centred on this pixel.

preserve the signals, yet not so large as to lose local sensitivity to the background. We tested window sizes of 11, 15, and 21 pixels. A window size of 10 pixels (radius of 5 pixels) was found to be too small, as it disrupted the PSFs. The choice between using 15 or 21 pixels is somewhat arbitrary. Next, we used the x, y coordinates of each cluster to extract the fluorescence time profile from the background-subtracted movie (3.1). The fluorescence value of each frame is the sum of intensities within a 3×3 window, centred at the centroid of the vRNP cluster.

The number of binding events in the fluorescence traces was detected using the pruned exact linear time algorithm (PELT) [85, 86] with the least squares deviation cost function to identify the mean shift in signals (3.2A). To be more specific, for signal $\{y_t\}_t$ on an interval I, the cost function is

$$c(y_I) = \sum_{t \in I} ||y_t - \overline{y}||_2^2 \tag{3.1}$$

where $\overline{y}$ is the mean of $\{y_t\}_{t \in I}$.

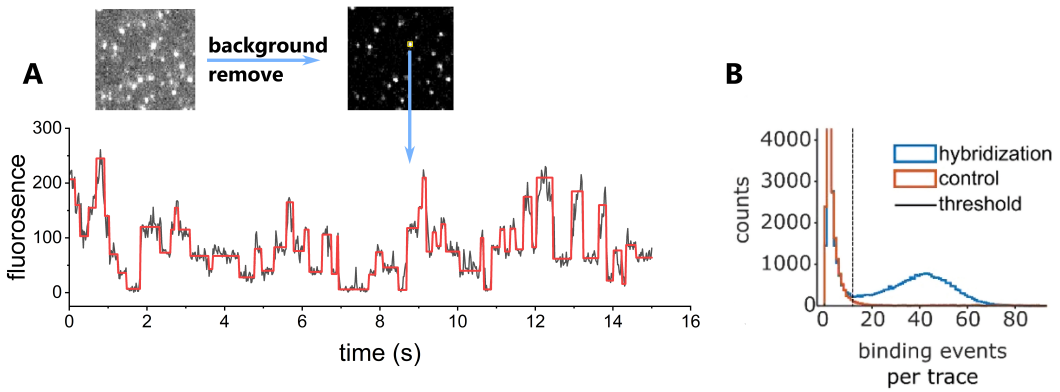

Figure 3.2: false-positive vRNP filtering. A provides a snapshot of a diffraction-limited movie showing NA vRNPs that were imaged with a high imager concentration of 50 nM, before and after the removal of background noise. The fluorescence intensity time trace of the highlighted spot from the background-removed movie (black line) and the results of change point detection using PELT analysis (red line) are also shown. B: This part illustrates the distribution of binding events per spot when imaging NA vRNPs. The blue histogram represents the assay with barcoded probes that are complementary to the NA segment. The red histogram serves as a negative control, having non-complementary sequences to NA vRNA. The vertical black line indicates the threshold for detecting viral segments.

The average computational complexity of the PELT algorithm is $\mathcal{O}(CKn)$, where C represents the complexity of the cost function on a sub-signal, K is the number of change points to detect, and n is the size of the data. PELT achieves computational efficiency by eliminating solution paths that are known not to lead to optimality. It penalises the inclusion of each additional change point by adding a penalty value to the cost function. The reasoning behind this approach is that for a timestamp to be identified as a change point, it must reduce the segmentation cost by more than the penalty value. Thus, selecting the appropriate penalty value is crucial to the effectiveness of PELT. PELT is significantly faster than other exact search methods and more accurate than approximate search methods, allowing us to balance the need for accurate detection and the demand for rapid processing in large datasets.

We compared the distributions of binding event numbers for each trace from the stoichiometry assay and control experiments. The control experiments utilised an

oligo set that contained the same number of sequences, with identical lengths and barcodes. However, the DNAs in this set were not complementary to any vRNA. The population showing a high number of binding events corresponds to specific bindings from the viral segments. In contrast, the population with fewer binding events relates to background noise (3.2B). A normal distribution was fitted to the genuine population to establish a threshold where the probabilities of false positives and false negatives are equal. When signals from the exact location were observed across multiple segment readouts, the likelihood of detecting a genuine viral particle increased with the same count of events. It becomes easier to distinguish between the two components (noise and genuine signals) within the distribution from positions exhibiting persistent signals (3.3). Thus, the event number thresholds for each segment were established based on the number of other segments detected at the same location (3.4). After noise filtering, we identified co-localised vRNPs using agglomerative clustering with single linkage and a distance threshold of 234 nm (under the same conditions as in Chapter 2). So far, we have acquired binary-coded segment combinations of virions.

Approximately 20000 viral particles can be detected per readout using our stoichiometry assay. We initially investigated the correlation of the co-presence rate of segment pairs among three technical duplicates (3.5). Our analysis yielded an overall Pearson's coefficient of about 0.88, indicating the reproducibility of our method.

3.1.2 Abundance and co-presence of vRNPs

We counted the virus particles based on the number of vRNPs they contain, which forms a U-shaped distribution, with the lowest counts observed at those containing 3 or 4 vRNPs (3.6A). This pattern has also been reported in a previous sequential FISH study [53]. It significantly deviates from what would be expected under a random packaging process. We also examined the missing vRNPs from virus

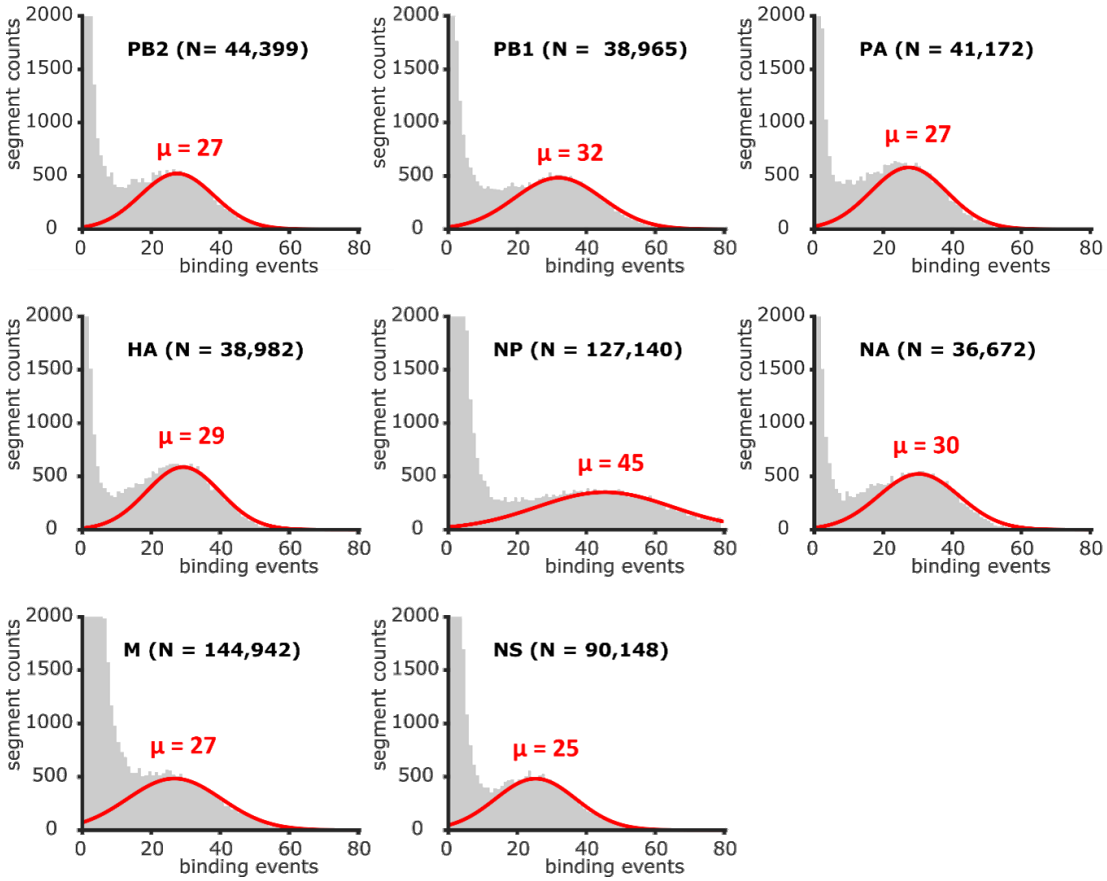

Figure 3.3: distribution of binding event numbers in the presence of other segments. The binding event distributions of spots showing a recurring fluorescent signal from all eight imaging cycles are presented. The red line indicates the Gaussian fit to the population corresponding to vRNPs. μ is the mean number of imager binding events to a vRNP and N is the number of traces analysed (the number of spots that are potential vRNPs).

particles that contain seven segments (3.6B) and the abundance of segments in the virus particles (3.6C). The difference in the number of absent segments is significant, while the quantities of each vRNP are pretty similar. These findings further support the selective packaging model.

Next, we focused on the co-presence rate of segment pairs from virus particles containing two segments (3.6D). This rate is defined as the fraction of the number of segment pairs relative to the mean of the two segment counts. Additionally, the most common five-segment combinations of complexes containing 2-7 vRNPs are displayed in Figure 3.6E. The normalization in Figure 3.6D measures interaction strength between two segments. Meanwhile, panel E shows absolute counts, which

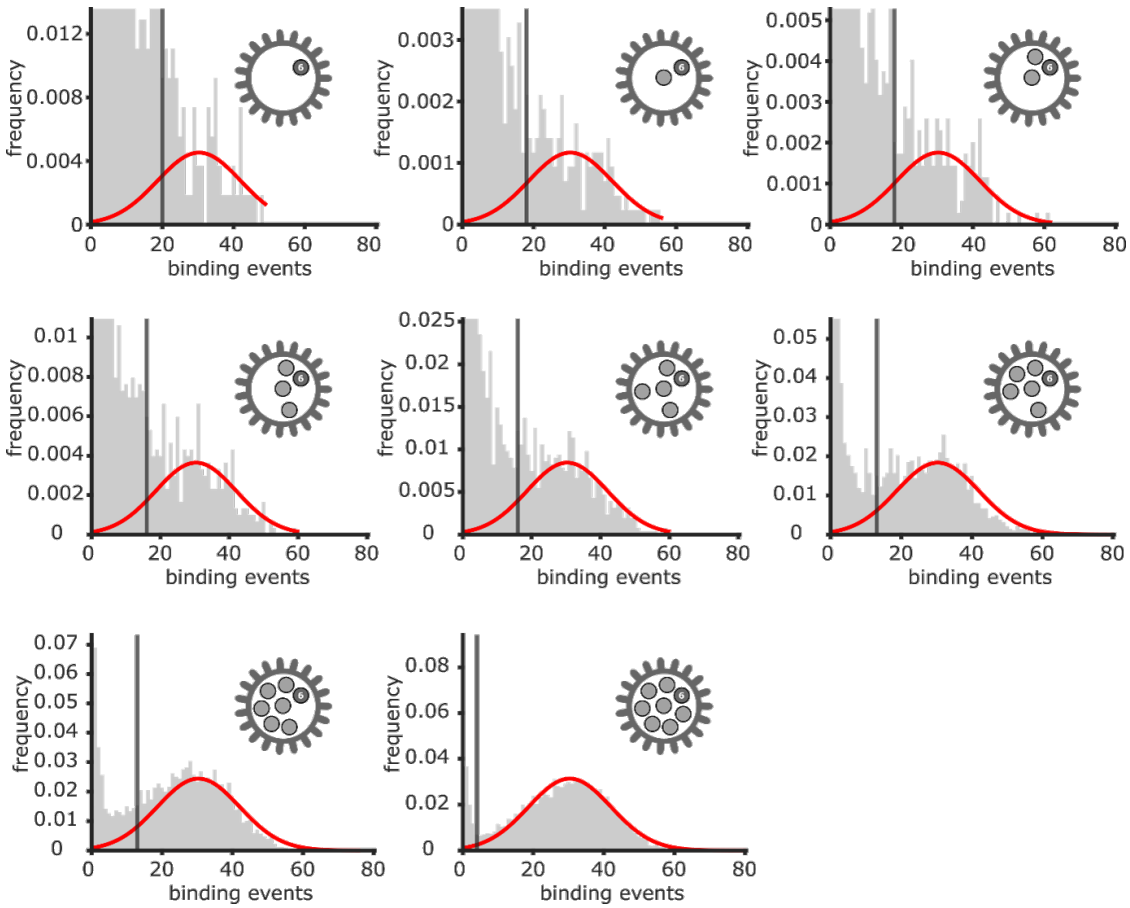

Figure 3.4: determining thresholds of binding event numbers. The detection threshold of a segment (binding events) depends on the number of potential segments. The threshold for the number of binding events in a segment is affected by the presence of fluorescent traces from other detection rounds, which indicate potential segments. This plot illustrates the number of binding events for the NA segment, categorised by the number of potential segments observed in other imaging rounds. The histograms are fitted with Gaussian functions, which were then used to establish the detection threshold, indicated by the black line. As the number of other segments increases, the proportion of the background population decreases in relation to the genuine vRNP population, and the detection threshold shifts to lower values.

are influenced by the varying numbers of different segments. There are no particular combinations that maintain high abundance across all segment counts. This observation suggests that the assembly process is not a straightforward linear process, where one segment is added at a time.

If we rule out the possibility of assembling multiple segments simultaneously, then a virus with three segments can only form via the pathway of 2 + 1. Since

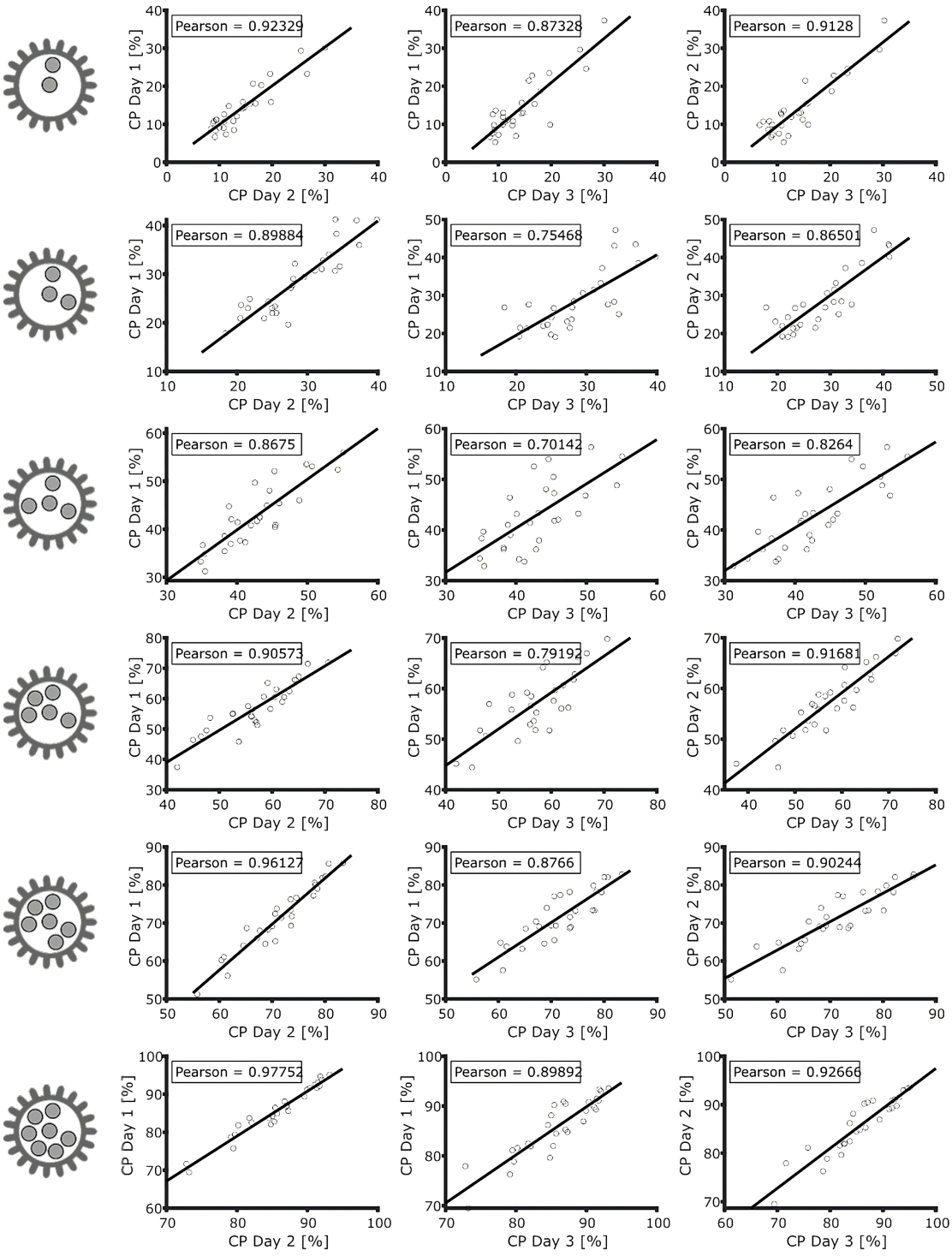

Figure 3.5: co-presence correlation between 3 technical duplicates from virions with different segment counts. Each panel contains 28 data points, representing all possible segment pairs from a total of 8 segments. The Pearson correlation coefficients range from 0.7 to 0.98, with specific values indicated in each panel.

there are no intermediate states for making the virus with two components, the normalised co-presence rate should positively correlate with the interaction strength between the two segments. By comparing the changes in abundance of viruses with 2 and 3 vRNPs (3.6E), it becomes clear that the assembly efficiency between these two states cannot solely be explained by the interaction strengths of the individual components with the newly added member. For instance, the third most abundant complex with two segments (85) becomes the most abundant (385) when segment 3 (PA) is added. In contrast, the 17 complex becomes less plentiful and shifts to the second position when the 2 (PB1) is introduced. However, based on Figure 3.6D, the interactions between 3-8 and 3-5 are weaker than 2-1 and 2-7, respectively. Several factors may contribute to this complexity, including occupied interaction sites and changes in secondary or tertiary structures resulting from vRNP-vRNP interactions in the intermediates.

We further investigated the changes in the normalised co-presence rate of all segment pairs from virions with 2-7 vRNPs. The normalisation is done via the following equation:

$$N = \frac{p/p_{\varnothing}}{(1-p)/(1-p_{\varnothing})} \tag{3.2}$$

where p is the co-presence probability of the segment pair in particles with a specific number of segments and $p_{\varnothing}$ is the average co-presence probability of particles with the same segment count. This equation accounts for the natural increase/decrease in the co-presence rate/mutual exclusion in virions with higher segment count and uses the ratio between co-presence and mutual exclusion to weigh the propensity of co-presence (3.8A).

3.1.3 Analysis of vRNP assembly pathway: association rule mining and community detection

We then expanded the analysis to explore all possible combinations. Excluding the scenario in which a single viral particle contains multiple copies of the same

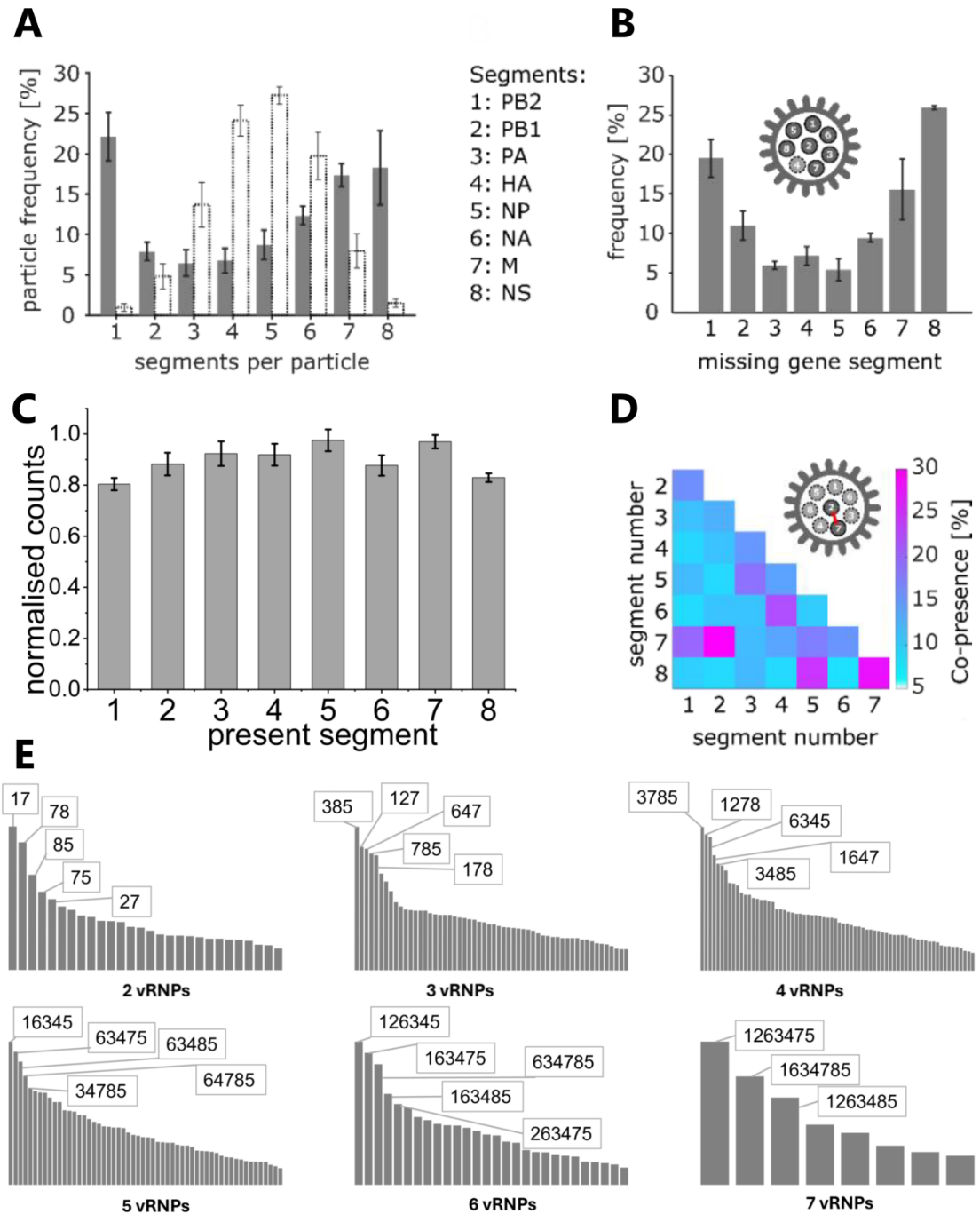

Figure 3.6: analysis of presence or absence of vRNPs. A: counts of virus particles categorised by the number of vRNA contained (grey bars), and comparison with the same datasets with segments shuffled between particles to simulate a stochastic distribution (white bars). B: Frequency of gene segments that are absent from particles with only one segment missing. Legend: Segment numbers that code for genes. C: The count of all detected segments is normalised by the most abundant segment for each assay. D: Co-presence rate in particles with two segments, normalised by the average segment count. E: The counts of different combinations of viral segments for complexes that have 2 to 7 components. The heights of bars are normalised by the highest combination count within each category.

vRNPs, the octameric IAV genome can create 255 distinct complexes and 3025 unique assembly interactions. To investigate these interactions, we decided to utilise the frequent pattern (FP) growth algorithm [87] for association rule mining, and present the statistical results in a network.

Association rule mining reveals the probabilistic relationships between itemsets in an extensive database. This method is widely used in various fields, including market basket analysis, web usage mining, and bioinformatics. The FP-growth algorithm identifies frequent items and utilises a suffix tree structure to encode all samples, allowing for the extraction of all frequent itemsets from the FP tree. An association rule takes the form $A \rightarrow C$, where A and C are disjoint itemsets, with A indicating the antecedent and C as the consequent. For each itemset, various statistical metrics are calculated. To select the most reliable association rules, we analysed the distributions of confidence, lift, and Zhang's metric, establishing appropriate thresholds (3.7). Their definitions are provided in the following equation:

$$\begin{aligned}
&\text{support}(A{\rightarrow}C) = \text{support}(A \bigcup C) \\
&\text{confidence}(A \rightarrow C) = \frac{\text{support}(A \rightarrow C)}{\text{support}(A)} \\
&\text{lift}(A \rightarrow C) = \frac{\text{confidence}(A \rightarrow C)}{\text{support}(C)} \\
&\text{Zhang's metric}(A \rightarrow C) = \frac{\text{confidence}(A \rightarrow C) - \text{confidence}(A' \rightarrow C)}{Max[\text{confidence}(A \rightarrow C), \text{confidence}(A' \rightarrow C)]}
\end{aligned} \tag{3.3}$$

where A' is the complement of the set A. Confidence is the probability of seeing the consequent in a sample given that it contains the antecedent. The lift metric measured how much more often the antecedent and consequent of a rule $A{\rightarrow}C$ occur together than we would expect if they are statistically independent. If A and C are independent, the Lift score will be exactly 1. The value range of Zhang's metric is $[-1, 1]$, while a positive number indicates association and a negative number suggests dissociation. As depicted in Figure 3.7C, there are no negative values in Zhang's metric. In other words, vRNPs and the intermediates of their combinations

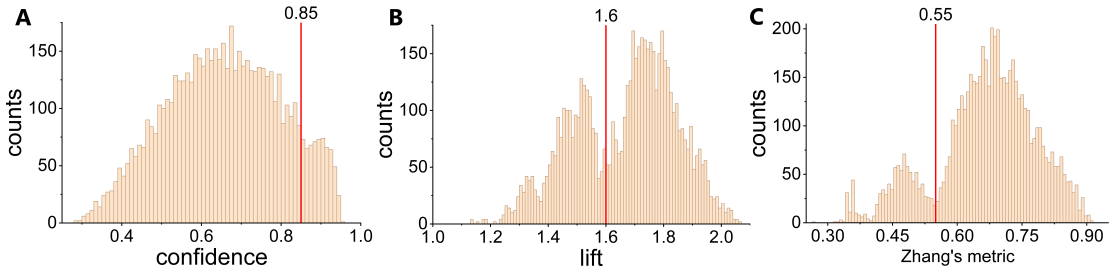

Figure 3.7: Histograms of parameters used to filter association rules. The red lines indicate the thresholds applied for filtering. A: The histogram shows the confidence metric with a threshold of 0.85. B: The metric is lift with a threshold of 1.6. C: The histogram represents Zhang's metric, with a threshold set at 0.55. These parameters are used to evaluate the significance of each association rule. Their definition can bee seen in equation 3.3.

attract each other to varying degrees but do not exhibit any repelling effects.

We present these association rules in a network (3.8B), where arrows indicate the direction from the antecedent to the consequent. The colours of the nodes represent different communities identified using the Leiden algorithm [88] with the modularity method [89]. Community detection functions as a clustering algorithm for a network by identifying nodes that form densely connected subgroups (referred to as communities). The modularity method aims to maximise the difference between the actual number of edges within a community and the expected number of edges. According to the configuration model [90], the expected number of edges is given by $\frac{K_c^2}{2m}$, where K_c represents the total degree of nodes in community c, and m denotes the total number of edges in the network. The definition of modularity is

$$H = \frac{1}{2m} \sum_c (e_c - \gamma \frac{K_c^2}{2m}) \tag{3.4}$$

where e_c is the actual number of edges in the network and γ is the resolution factor. Higher resolution results in the identification of more communities, while lower resolution leads to fewer communities. In plot 3.8B, the optimal partition is obtained with a resolution of 1.0.

In this graph, the item sets (indicated by the cyan boxes in Figure 3.8A) that play central roles in each community contain pairs of segments that have an increasing

relative co-presence rate in a more complete virus. These pairs consist of PA(3) and NP(5) combined with one of three other segments: PB1(2), HA(4), NA(6), or with each other. This suggests that the PA and NP segments have a special role in the genome assembly process. Other non-central item sets could serve as intermediates, particularly those involved in multiple communities, as they may facilitate numerous assembly pathways. Additionally, the number of non-central nodes in each community may provide insights into the flexibility of each potential pathway—the more nodes present, the greater the options available for assembling a more complete complex.

Some combinations, such as NS-M (8-7), exhibit a rapid decline in stoichiometry as the segment number increases (3.8A), suggesting that their initial combination may lead to incomplete packaging. The three most abundant segment pairs (7-2, 8-5, 8-7) found in two-segment virions all fall into this category. Additionally, we observed that other segment pairs, such as 6-4 (M-PB1), although they decrease, maintain a decent value in the co-presence rate as they approach the end. This indicates that while they may not be effective as initial seeds, they may join other pathways as a pair.

In summary, our results dismiss the idea of a single direct or stochastic assembly pathway. Instead, they suggest multiple possible pathways with varying preferred hierarchy orders for vRNPs and their intermediates to join the complex (3.9). The association rules in Figure 3.8 link segment pairs and intermediate complexes, resulting in a high-segment-count genome. However, the association rules that describe the relationships of the non-central components and their intermediates are lost during the filtering process. This indicates that these relationships are not as 'strong', likely due to the large number of possible combinations when the segment count is near four; in the meantime, the number of such virions is significantly lower, or these intermediate processes may experience weaker interactions. There is only one non-central subcomplex with a minimum segment count of 3 that appears in

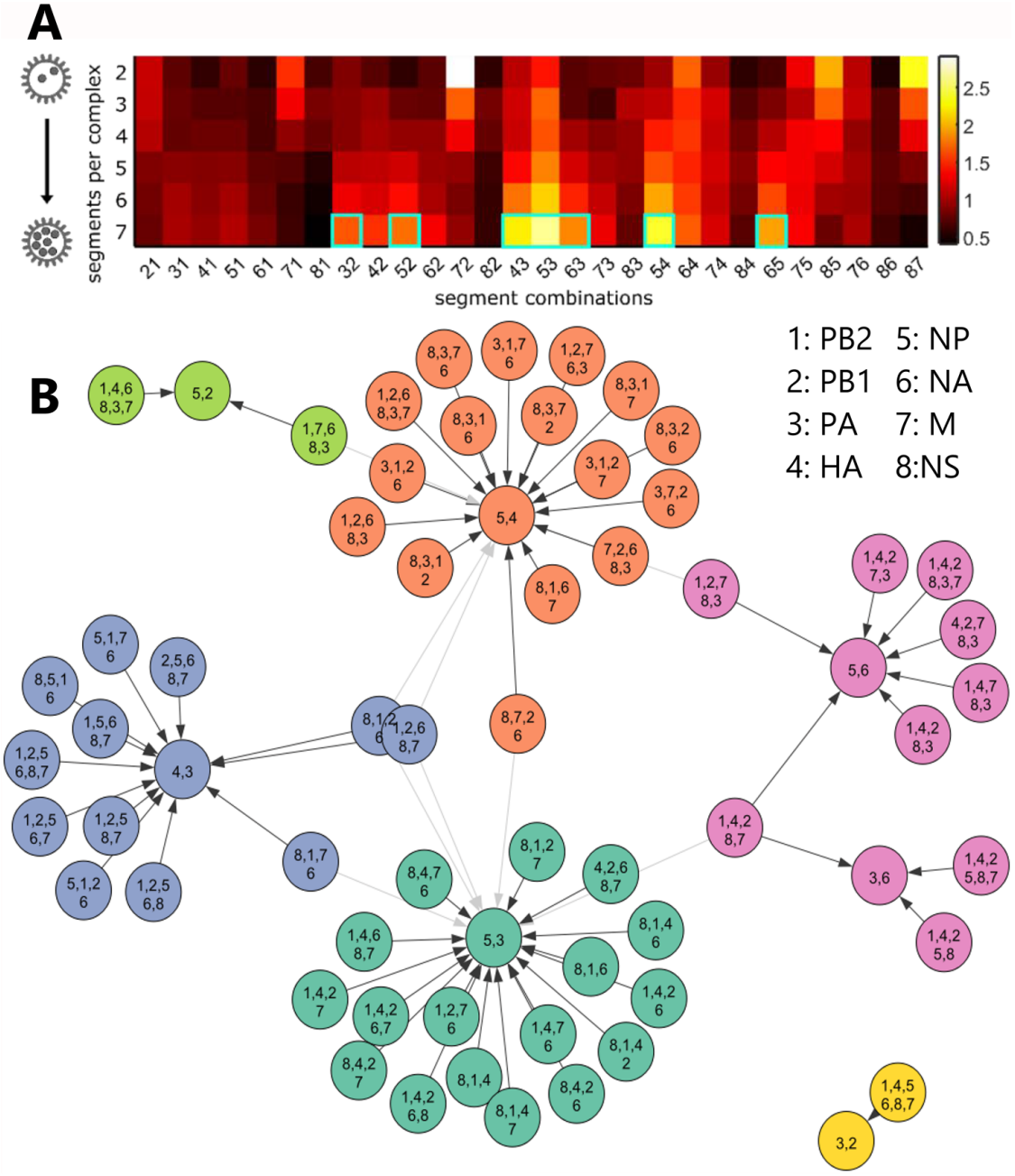

Figure 3.8: segment co-presence rates and association-rule network. A: Normalised segment co-presence in particles with segment counts ranging from 2 to 7. The cyan boxes indicate segment pairs that form association centres in B. B: A network illustrating significant associations between vRNP subcomplexes, with arrows indicating relationships from antecedents to consequents as determined by the FP algorithm in a Fruchterman-Reingold layout. The communities identified by the Leiden Algorithm are represented in different colours.

the network (816 interacts with 53), which suggests a relatively strong interaction compared to other early intermediate-early intermediate interactions that lead to late intermediates.

We also considered the possibility that the incomplete genome observed in the mature virus might result from erroneous pathways. However, similar segment-count distribution patterns (3.6A) have been found in a budding virus as well. Furthermore, we hypothesise that stronger interactions will lead to smaller pairwise distances and greater co-presence rates. The negative correlation between the co-appearance rate and distance among all 28 pairs (2.15E) supports this hypothesis. Therefore, it is speculative that the associations we observed arise from the assembly process being aborted or stalled.

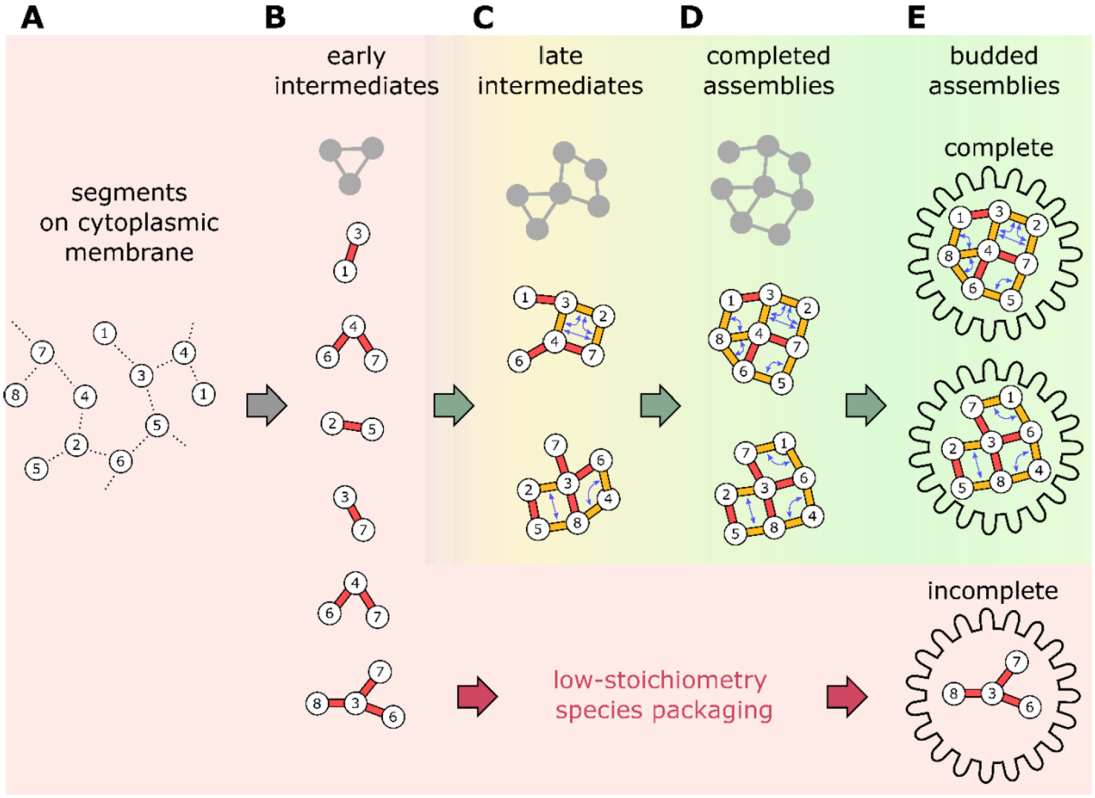

Figure 3.9: a model for influenza genome assembly via multiple pathways. A: Newly synthesised vRNPs are concentrated at a membrane. Unspecific interactions (represented by dashed lines) between all segments help localise the vRNPs, which will later facilitate segment-specific interactions. B: In the early stages of assembly, strong interactions (shown as red sticks) between individual vRNPs and their pairs form low-order subcomplexes. C: During the later stages, weaker interactions (illustrated as orange sticks) cooperatively stabilise each other (indicated by double arrows) to form higher-order complexes (green arrows). This process results in the assembled viral genome (D), which is packaged into budding virus particles (E). The illustrated assembly routes represent conceptual examples, as a multitude of RNA contacts can lead to many alternative assembly pathways (depicted by grey outlines). If lower-order subcomplexes take too long to grow to sizes at which cooperative effects can enhance assembly, incomplete viral genomes may be packaged (indicated by red arrows).

3.2 Fluorogenic DNA-PAINT

DNA-PAINT suffers from a low imaging speed due to its requirement for a low imager concentration. The unbound imagers appear as background fluctuation or erroneous binding events, depending on the exposure time and their diffusion speed. At long exposure times or fast diffusion speeds, they contribute a high background, drowning the signal from molecules, which causes the loss of localisations during detection and deteriorates the localisation precision. The raw images are often processed to remove a heterogeneous background using computational techniques such as the difference of Gaussian, rolling ball algorithm, and wavelet filtering. However, processing the image may disrupt the diffraction pattern of the PSF and have a negative influence on the following localisation step. To suppress background experimentally, researchers use techniques such as TIRF to minimise the illumination volume, and low imager concentration (usually only a few nanomolar). To boost imaging speed, one can optimise the DNA sequence and buffer conditions to reduce the dark time [65]. However, if one wants to increase the imaging speed even more, the exposure time becomes so short that the unbound imagers are no longer blurred by diffusion, which leads to false binding events.

3.2.1 Fluorogenic single-molecule microscopy

An ideal way to address both background and false binding issues is to create imagers that are dim when they diffuse in the solution and bright when they bind to the targets. FRET-based DNA-PAINT has been proposed to achieve this effect. The donor and acceptor are either conjugated to two different imagers or one to a docking strand and the other to an imaging strand. Only when they coincidentally bind to the same docking strand (3.10B2) or the imager binds to the docking strand (3.10B1), the energy from the excited donor is transferred to the acceptor, whose emission is then detected. However, because of the trade-off between energy-transfer efficiency and excitation-emission cross-talk, the localisation precision that has been reported so far is substantially worse than that of regular DNA-PAINT (3.10A) [91,

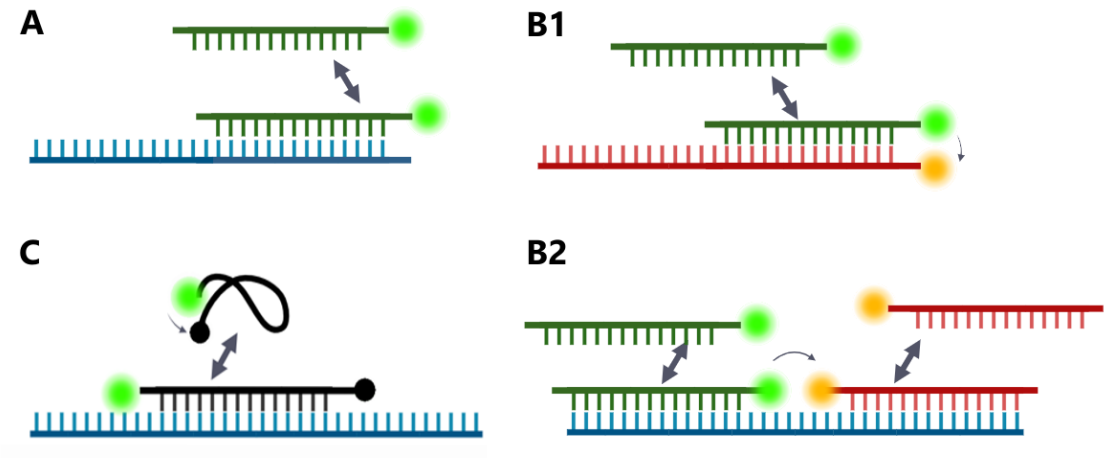

Figure 3.10: probe design of regular, FRET and fluorogenic DNA-PAINT. This plot shows the concept of regular, FRET and fluorogenic DNA-PAINT, comparing their imager probe and docking strand systems. A: Regular DNA-PAINT B: Two different designs for FRET DNA-PAINT. The acceptor fluorophore is only detectable when both donor and acceptor strands are bound to the target together (B1) or when the imager with donor fluorophore binds to the docking strand (B2). C: The fluorogenic probe is quenched in solution and becomes fluorescent in the binding state.

92].

Another approach to achieving this effect is through the use of fluorogenic probes, which change their fluorescence intensity in response to alterations in their micro-environment. Fluorogenic dyes start in a quenched state (non-emissive or weakly emissive) and transform to a fluorescent state upon interaction with or reaction to their targets. These probes can be categorised into two types: those that begin with a quenched state, where interactions restore fluorescence, and those where interactions with the target create fluorescent structures. An important photophysical property of a fluorogenic probe is the ratio of emission intensity between free and bound probes, known as the fluorogenic factor (FF). The binding-induced changes in fluorescence can exceed a 10,000-fold increase [93]. This property enables us to use a considerably higher concentration of imagers compared to regular probes, which accelerates the imaging process while maintaining a low background. In the case of fluorogenic DNA-PAINT, an imager strand is designed to have a dye on one end and a corresponding quencher on the opposite end. In

the unbound state, the imager coils bring the dye and quencher into proximity, resulting in the quenching of the fluorescence signal (3.10C). The efficiency of quenching and the enhancement of fluorescence depend on duplex formation, which can be engineered through various combinations of fluorophores and quenchers, label length, sequence, and other factors.

3.2.2 High-speed DNA-PAINT

We utilised a fluorogenic probe designed to target the PB1 segment, following the same sample preparation protocols as described in 2.4. Dr Mirjam Kummerlin designed two fluorogenic probes: 6ntI_A647N_BHQ1 and 6ntI_A643_BMNQ1. They share the same sequence of 5'-TGGTGG1-3'. These probes (purchased from biomers.net, Ulm, Germany) are 6 nt long, featuring Atto647N or Atto643 at the 5' end and a black hole quencher 1 (BHQ1) or BMN-Q1 at the 3' end. Their docking strand is 5'-CCACCACCACCA-3', which has three complementary sites for binding with the 6nt imagers (3.11B).

For 6nt imagers, the movies were recorded at an exposure time of 20ms, using Nanoimager (Oxford Nanoimaging) under TIRF model with an illumination angle of 54°, 638 nm continuous laser with power of 150mW. To compare the performance of the fluorescent and fluorogenic probes, we also imaged the NA segment with the fluorescent P3 imager (3.11A) using Nanoimager, TIRF model at 54°, 200ms exposure time, 532nm excitation laser, 3.6 mW.

BHQs achieve quenching through a combination of FRET and static quenching. In FRET, the excited fluorophore transfers its energy to BHQ so that the fluorophore returns to the ground state and the energy absorbed by BHQ is dissipated as heat. In static quenching (also known as contact quenching), the fluorophore and quencher combine to form a new, non-fluorescent intramolecular dimer. The static quenching

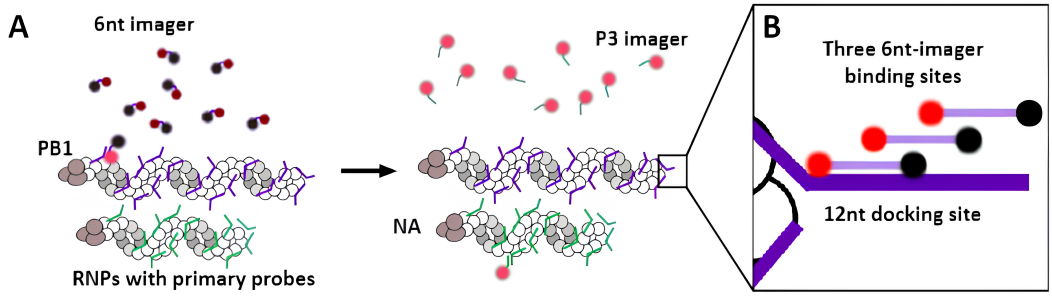

Figure 3.11: protocol of single-virion fluorogenic DNA-PAINT. A is a schematic representation of our fluorogenic DNA-PAINT assay. Viral RNAs are hybridised with primary probes that contain an extension docking sequence. In sequential rounds of imaging, we first visualise the PB1 segment using a fluorogenic 6nt imager (depicted with a purple backbone), followed by imaging the NA segment with the P3 imager (green backbone). B: Arrangement of docking sites for the 6 nt imager.

efficiency depends on the affinity between the fluorophore and the quencher. BMN-Q1 is also a dark quencher having the same characteristics as BHQ.

When the probe length is small, the FRET effect reduces the fluorescence of the probe in binding states. The small overlap between Atto647N and BQH1 leads to a small Förster distance ($R_0 \approx 3nm$) and subsequently has small loss of fluorescence intensity upon hybridisation (3.12A). R_0 is calculated using the formula

$$R_0 = 0.0211(\frac{\kappa^2 \Theta_D J(\lambda)}{n^4})^{1/6} \tag{3.5}$$

where κ^2 is the dipole-dipole orientation factor, which is 2/3 in case of random orientation, n is the refractive index of the medium, Θ_D is the quantum yield of the donor in the absence of acceptor, and J is the spectral overlap integral between the acceptor absorption spectrum and the area-normalised emission spectrum of the donor. (The manufacturer does not provide the spectrum of BMN-Q1, and therefore, the R_0 of Atto643-BMNQ1 is not calculated.)

To calculate the FF of the 6nt imagers, we began by taking the average photon counts from the localisations of the fluorogenic probe (S_{FQ}) and dividing them by the average photon counts from probes that do not contain quenchers (S_F),

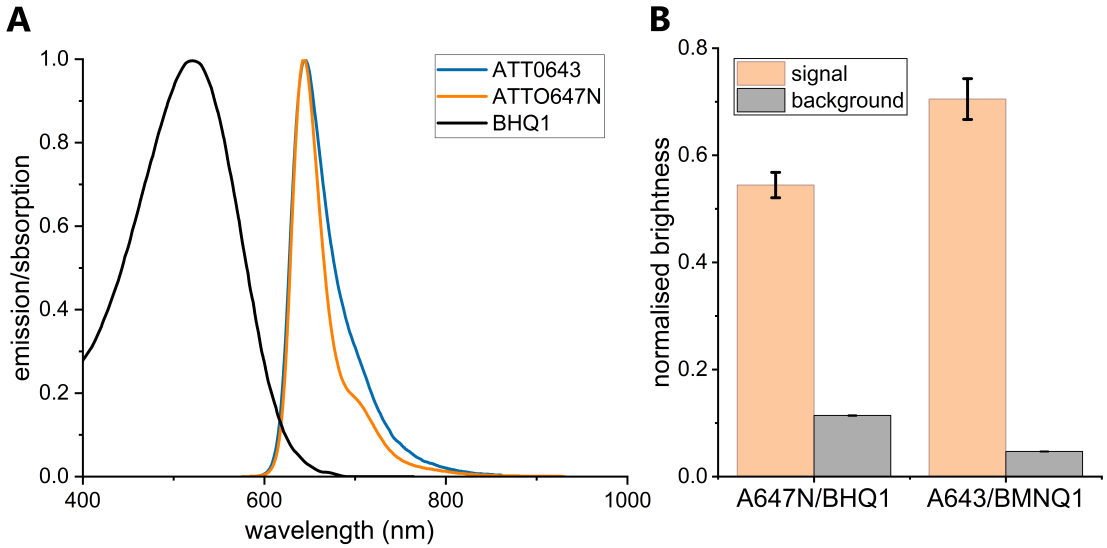

Figure 3.12: spectrum and fluorogenic factors of the imagers. A: The emission spectrum of Atto647N and Atto643, along with the absorption spectrum of BHQ1 (provided by the manufacturer). B: The normalised signal and background of the two 6mer imagers, which gives FFs of 4.77 ± 0.11 (Atto674N-BHQ1) and 15.04 ± 0.57 (Atto643-BMNQ1).

which provides us with the normalised signal (S_N). Due to the sparse nature of the binding events, we estimated the background intensities by using the median value from the central area of the FOV throughout the movie. Next, we performed a linear regression analysis of the background intensities against imager concentration to determine the molar background ($B_{FQ,mol}$). We then normalised this molar background by comparing it to the molar background emission from probes that are only labelled with a fluorophore ($B_{F,mol}$), resulting in the normalised background ($B_{N,mol}$). Finally, the fluorescence factor (FF) was calculated as the ratio of the normalised signal (S_N) to the normalised background ($B_{N,mol}$).

$$FF = \frac{S_N}{B_{N,mol}} = \frac{S_{FQ}/S_F}{B_{FQ,mol}/B_{F,mol}} \tag{3.6}$$

The FFs are about 5 and 15 for Atto647N-BHQ1 and Atto643-BMNQ1, respectively (3.12B).

We conducted a control experiment to examine the specificity of the fluorogenic probe by using a sample without primary probes carrying the docking site for our

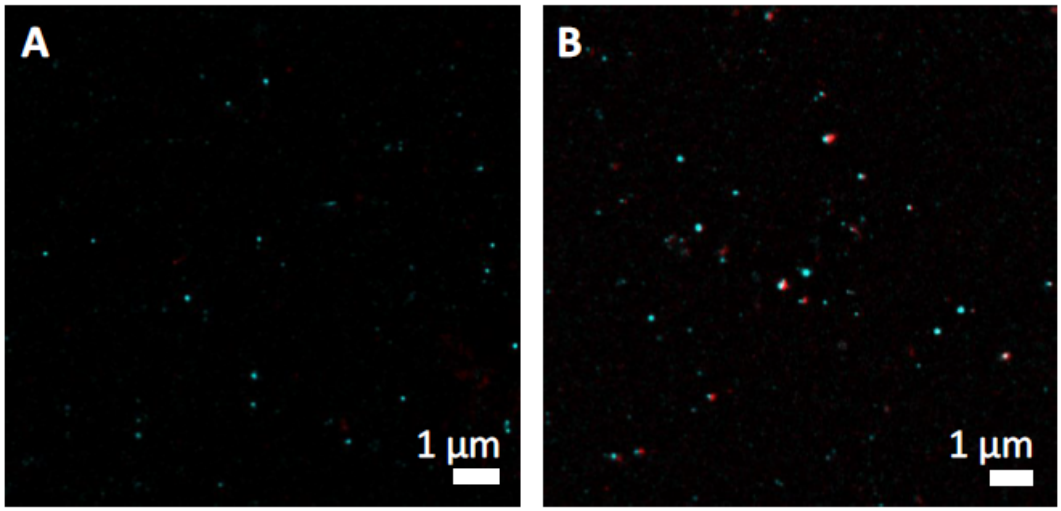

Figure 3.13: negative control for fluorogenic imagers. A: Negative control without primary probes (red) shows few localisations compared to the sample with primary probes(cyan). B: PB1 imager (Cyan) and NA 6nt imager (A647N-BHQ1, red) are co-localised.

6nt imager strand. One can observe that the 6nt signal only exists when the primary probe was applied (3.13A) and PA(red) and PB1 segments are co-localised (3.13B).

Then, we acquired the localisations using Picasso (Box side length: 7, Min. Net gradient: 3000, Baseline: 400, Sensitivity: 2.5, Quantum efficiency: 0.82, Pixel size: 117 nm). Drift correction was performed using the AIM algorithm (segmentation: 100, interaction distance: 20nm, maximum drift in segment: 60 nm), and the potential vRNPs were identified using DBSCAN(eps = 58.5nm, min_samples = 0.01* frame number).

Compared to the P3 imager, both 6nt fluorogenic probes exhibited a higher number of non-specific binding events, resulting in the formation of larger and less dense clusters, in contrast to the more compact viral segments. We quantified the compactness of clusters by the ratio of core samples and non-noise samples identified by DBSCAN. By observing the distributions of core sample ratios (3.14A) from P3 and 6nt imagers, we set the threshold of 0.8 to filter out most false clusters (3.14B). Due to the advantages of higher FF and reduced noise that comes with the high

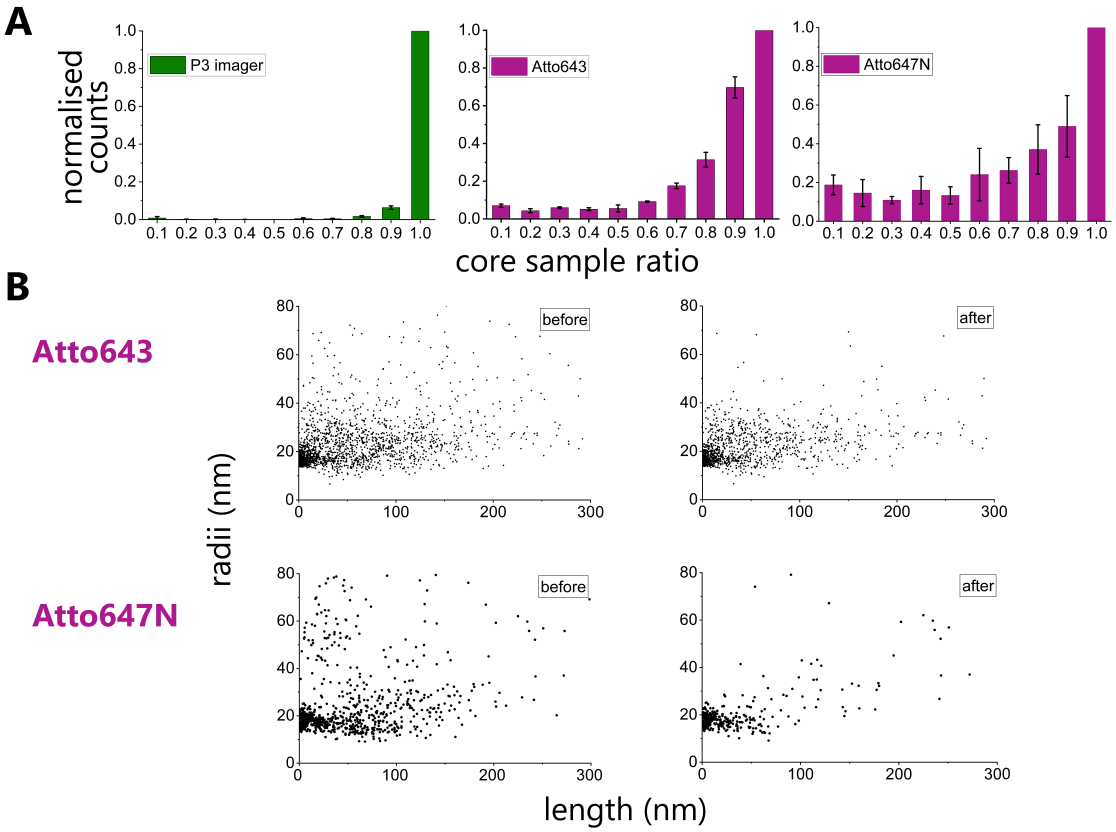

Figure 3.14: false-positive vRNP filtering based on core sample ratio from DBSCAN. A: The average normalised counts of core sample ratio for the P3 imager and the two 6mer imagers, Atto643-BHQ1 and Atto647N-BMNQ1. The error bar is the standard deviation from three technical replicates. B shows the distribution of segment lengths and radii before (left) and after (right) the clustering filtering process, from Atto643-BMNQ1 (top) and Atto647N-BHQ1 (bottom) imagers, respectively.

solubility of Atto643-BMNQ1 fluorophore, we conducted further experiments and analysis using data from this imager.

Super-resolution images of each segment were generated by treating each localisation as a Gaussian function, followed by low-pass filtering, binarisation, and skeletonisation to get the length and width of the segment (detailed process in chapter 2). Most values of vRNP radius acquired from our experiment lie between 10 and 20nm, in agreement with previous studies[94, 95] and our exchange DNA-PAINT experiment in section 2.8.1. Next, we fit the radius distributions using a two-component Gaussian model, where one component with a larger average value represents the remaining noise and the other presents the genuine vRNPs

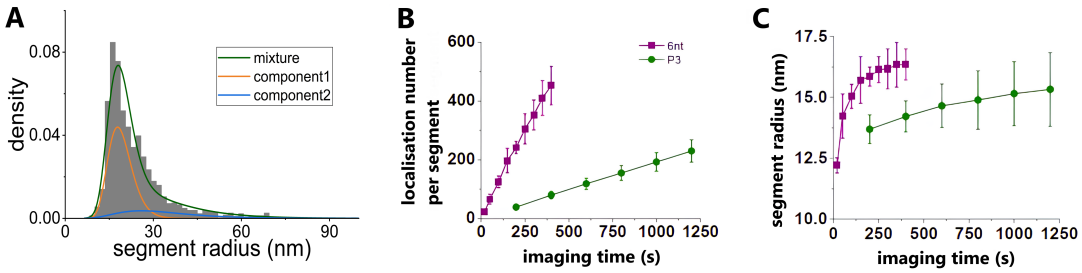

Figure 3.15: convergence of segment radius over imaging time and average binding rates of imagers. A: Two-component Gaussian fit of the radius distribution. B: the change of average localisation number per segment over imaging time. C: The mean value of radius converges with the imaging time increasing.

(3.15A). The use of a Gaussian function is an empirical choice, as these large clusters originate from 'sticky' spots on the coverslip, and there is no specific expectation for the shape of such areas.

We used the centre of the Gaussian to present the radius of the segment at a specific imaging time. It reaches convergence for the 6nt imager at 16 nm and the P3 imager at 15 nm (3.15C). This 1nm difference could be explained by the two extra binding sites on the docking strand of PB1 segment (3.11B). We evaluated the accuracy of the width convergence analysis by randomly splitting the localisations from the same vRNP into two sub-groups and processing them through the same pipeline. The difference in segment radius between the two sub-groups was relatively small, measuring smaller than 0.5 nm for the 6nt imager and 0.25 nm for the P3 imager, across all the imaging times investigated.

Notably, it took approximately 150 seconds and 800 seconds for the radius to reach convergence for the 6nt and P3 imagers, respectively. The faster convergence of 6nt data can be explained by the higher on-rate of the 6mer imager, which can be visualised as the average number of localisations per viral segment. The average localisation number of the 6nt imager is about 6 times higher than that of P3, and both of them show a strong linear relationship with the imaging time (3.15B).

Next, we applied Fourier ring correlation (FRC) to analyse the resolution changes with imaging time. Because SMLM images are rendered from a series of individual fluorophores sparsely located in space and time, these images are fundamentally different from traditional diffraction-limited images. The resolution of structures depends on the localisation precision and the density of fluorescent labels. To answer this question, the FRC was proposed [96]. To compute FRC resolution, the localisations that constitute a super-solved image are split into two statistically independent subsets $f_1(\vec{r})$ and $f_2(\vec{r})$ where $\vec{r}$ is the spatial coordinates. Their Fourier transforms $\widehat{f_1(\vec{q})}$ and $\widehat{f_2(\vec{q})}$ on the perimeter of circles of constant spatial frequency with magnitude $q = |\vec{q}|$ give the RFC

$$FRC(q) = \frac{\sum_{\vec{q}\in\text{circle}} \widehat{f_1(\vec{q})}\widehat{f_2(\vec{q})}}{\sqrt{\sum_{\vec{q}\in\text{circle}} |\widehat{f_1(\vec{q})}|^2}\sqrt{\sum_{\vec{q}\in\text{circle}} |\widehat{f_1 2(\vec{q})}|^2}} \tag{3.7}$$

For low spatial frequencies, the FRC is close to 1. In contrast, at high spatial frequencies—where noise overshadows the actual structure—the FRC declines close to 0. Resolution is defined as the inverse of the spatial frequency at which the FRC falls below a specific threshold. We used a threshold of 1/7, which has been reported as the most suitable for localisation microscopy [97].

We tracked the resolutions of images whose corresponding clusters reach the minimum core sample ratio of 0.8 at the end of the whole movie using the most frequent resolutions as the typical resolution achieved at specific imaging time (3.17A). We summarised such a change over time in Figure 3.17B.

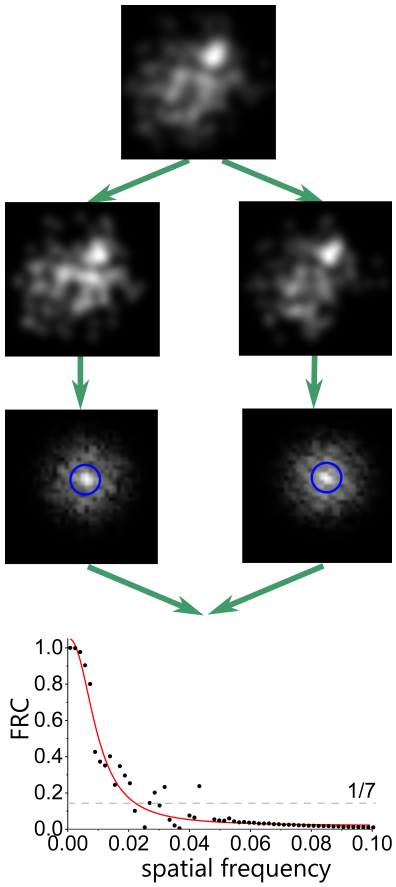

Figure 3.16: procedures of FRC analysis. The process of determining FRC resolution involves creating two super-resolution images from randomly split localisations, calculating the correlation between two rings with a series of radii in Fourier space, fitting the series of spatial frequencies and their corresponding FRC values to a logistic function (red line), and finding the intersection of the fitted function with the threshold of 1/7.

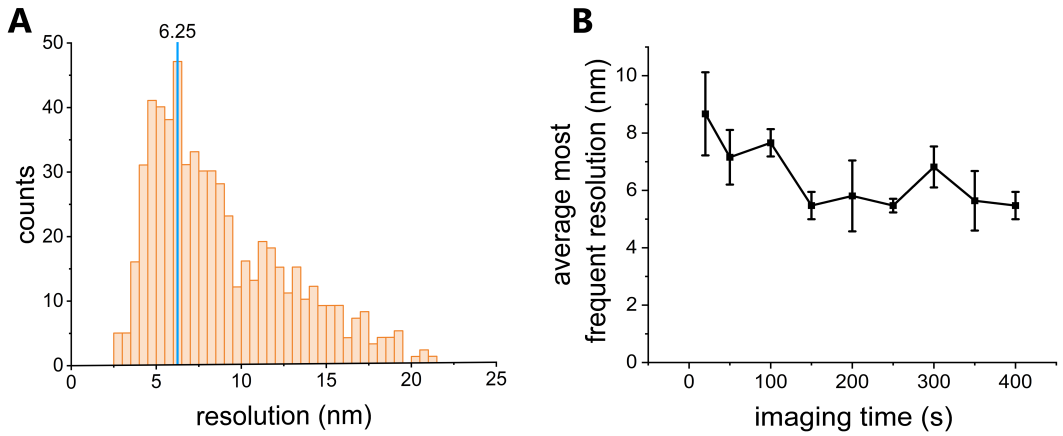

Figure 3.17: FRC resolution converges over time. A: Histogram illustrates the resolution distribution from a 20000-frame movie using 6nt imagers. The blue line indicates the most frequent value, which is selected to represent the resolution achieved at the imaging time. B displays the improvement of resolution as the imaging time increases.

3.3 Summary of the single-virion DNA-PAINT

We utilised DNA-PAINT to investigate the genome of the IAV. This method involves fixing DNA oligos onto the RNP through hybridisation. The extensions of the oligos provide docking sites for DNA imagers labelled with the Cy3B fluorophore. We recorded and localised binding events between imagers and the vRNA with high precision. The super-resolved images and rendered volumes provided structural information about each vRNP and their relative spatial arrangement. We also studied the pair-wise distance and their co-presence rates. Additionally, an effective high-throughput stoichiometry assay allowed us to determine the combination of vRNPs from virus particles, offering insights into the assembly process of IAV's segmented genome. The combined information from both super-resolution and stoichiometry assays ultimately leads to the interpretation of the genome assembly pathway (2.4).

Our findings reveal that the viral RNPs exhibit highly flexible and heterogeneous structures, as well as diverse spatial arrangements. The assembly of the influenza genome occurs through multiple pathways in a selective, redundant, and cooperative manner. The vRNPs likely form medium-sized intermediates, and the subsequent assembly process is completed through interactions with both other intermediates and individual viral RNPs.

The data analysis pipeline used to investigate the IAV samples is summarised in the flowchart 3.18. The two most computationally intensive steps in this pipeline are localisation and clustering filtering through change point detection. To expedite localisation, we fitted the PSF to Gaussian functions using the Levenberg-Marquardt algorithm implemented on CUDA [98]. The PELT algorithm used in change point detection was implemented in C.

In addition to utilising more efficient hardware (such as GPUs or more powerful CPUs) and programming languages, we also adapted more efficient algorithms.

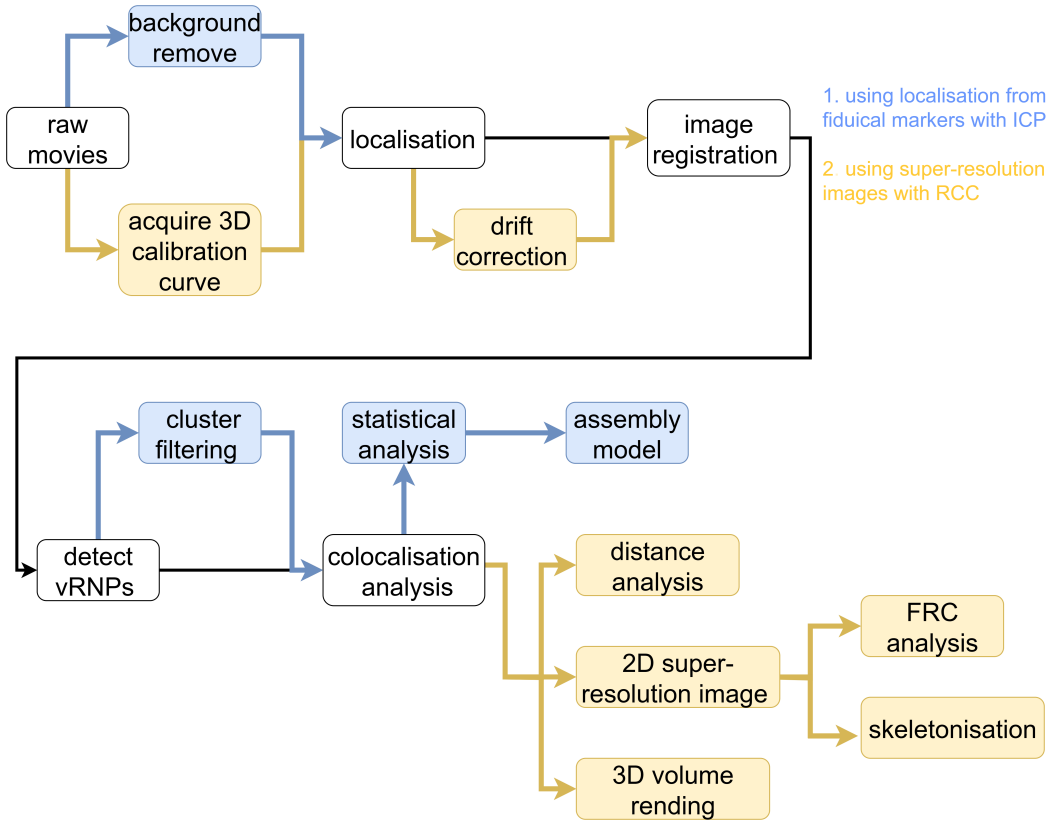

Figure 3.18: data analysis pipeline of single-virion assay.This plot summarises the data analysis methods we applied to study the IAV particles. yellow and blue boxes indicate the methods that are only applied to data from 3D astigmatism super-resolution and stoichiometry assays, respectively. The common steps that are involved in both assays are indicated in the white boxes.

For instance, we utilised a DBSCAN specifically designed for low-dimensional Euclidean spaces, which operates with a time complexity of $\mathcal{O}(n \log n)$ in two dimensions. The octree structure used in volume rendering effectively reduces the number of samples that need to be processed, and PELT is also highly efficient with an average computational complexity that increases linearly with the size of the data.

4

Single-molecule sequencing on gapped DNA

Sequencing techniques play an essential role in basic biological research, allowing scientists to ask questions related to the genome. Traditional DNA sequencing methods, such as the Sanger method, rely on amplifying the template DNA and gel electrophoresis. Although Sanger sequencing is still considered the 'gold standard' for DNA sequencing methods, this first-generation technology is slow and expensive. Over the last decades, next-generation sequencing methods have been developed in response to the demand for cheap, fast, accurate, and easy-to-implement techniques with longer read lengths. Commercial single-molecule, real-time sequencing methods such as Illumina sequencing [99] and Oxford Nanopore Technology [100] have been developed. However, they both require special devices. Here we present a new technique that can sequence gapped DNA molecules at the single-molecule level using a standard wide-field microscope.

This method leverages the variations in the thermodynamic properties of DNA sequences as they interact with their complementary and mismatched strands. The fluorescence profile from labelled oligos indicates the subsequent distinct kinetic behaviours. We developed two types of assays: competitive and non-competitive

sequencing, based on this concept. As a proof of concept, we used them to sequence single, three, and five bases.

4.1 Experiment protocol of base calling

Dr Jagadish Hazra developed the protocols for base calling and conducted the related experiments. I established the data analysis pipeline outlined in this chapter.

The DNA sequences (purchased from Biomer Corp.) and their corresponding strand names are listed in Table A.2. The protocol for preparing a NeutrAvidin-biotin-PEG/mPEG functionalized coverslip is described in section 2.4.

single molecule imaging All single-molecule measurements were conducted using the Nanoimager microscope (Oxford Nanoimaging), equipped with a 100X, 1.4 NA objective, and a sCMOS camera (Orca flash4 v3). The TIRF mode was applied with an illumination angle of 56°. The emission channel was divided into two using a dichroic 640 LP splitter. One is for wavelengths ranging from 498 to 551 nm and 550 to 620nm. The other is for wavelengths 685/40 nm.

For imaging, we added 40 μL solution containing DNA seals in an imaging buffer to the observation wells and incubated for 1 minute. The seal concentration for each experiment is stated in the corresponding parts. Specific parameters for each assay, including exposure time, excitation wavelength, and the number of frames, are detailed in Table 4.1. After one imaging round was completed, the surface was washed using 40 μL of assay buffer (AB: 20 mM HEPES, 200 mM NaCl, 80 mM $MgCl_2$ of pH 7.7) three times to remove the old imagers.

Form Gap-containing DNA substrates To form gapped DNA, oligos were annealed by mixing two complementary strands in a ratio of 1:1 in hybridisation buffer (200 mM Tris-HCl, pH 8.0, 500 mM NaCl, 1 mM EDTA). The mixture was heated to 95°C for 5 minutes to denature the strands, followed by gradual cooling

experiment type	movie type	exposure time (ms)	frame number	excitation wave length (nm)	laser power (mW)
non competing	localiztion movie	200	1000	532	0.2
	sequence interrgation movies	300	1000	638	0.6
competing	localiztion movie	250	1200	532	0.2
	sequence interrgation movies	250	1200	638	0.6

Table 4.1: image acquisition conditions for base calling. This table presents the image acquisition conditions for all base calling methods discussed in this chapter, including exposure time, number of frames, wavelength, and excitation laser power.

to 25°C at a rate of 2°C per minute to facilitate annealing. The resulting DNA duplexes were then placed on ice until further use.

DNA substrates immobilisation First, we prepared NeutrAvidin-biotin-PEG/mPEG functionalised coverslip (#1.5 Menzel/ThermoFisher) using the same approach described before in Section 2.4. Biotinylated DNA constructs were incubated at 50 pM for 30s to 1min to acquire a density of about 2 immobilised molecules per μm^2 and followed by a washing step with 200 μl of AB. Next, 40 μL of FluoSpheres^{TM} biotin-labelled microspheres (Thermo Scientific, 200 nm in diameter) was added as fiduciary markers at a ratio of 1:2000 in AB. Next, we exchanged the buffer in the investigation well with AB three times to remove unattached beads. We added 10 μL 50pM gapped DNA in imaging buffer 1 (IB1: 0.5 M NaCl, 50 mM HEPES pH 7.3, 6 mM BSA, 1 mM TROLOX, 1% Glucose, 40 μg/ml catalase and 0.1 mg/ml glucose oxidase) for 10s, and followed by the other three washes to remove the unattached beads.

Sequencing via non-competitive approach The 8nt gapped DNA was constructed by annealing biotinylated sequence, Cy3B-labelled strand (Gap(1-27)-Top^{Cy3B,18}) and another flanking strand (Gap(36-65)-Top). Four 100nM 8nt non-competitive seals are labelled with Atto647N with one different base at position 3 of the Gap in imaging buffer 2 (IB2: 20 mM HEPES, pH 7.7, 200 mM NaCl, 80 mM $MgCl_2$, 5% dextran sulphate, 10% formamide, 1 mM TROLOX, 1% Glucose, 40 μg/ml catalase and 0.1 mg/ml glucose oxidase) were imaged sequentially.

Sequencing via competitive inhibition For competitive single base calling, our R-seal was Atto647N-labelled 8nt oligo with a degenerated base (equal probability of having any of the four standard bases) opposite to position 3 of the DNA gap (R8-N^3). Next, we added four unlabelled oligos of 8 nt length having A (U8-A^3), T (GU8-T^3), G (U8-T^3) or C (U8-C^3) at a position complementary to the third base of the gapped DNA in a sequential manner. We use 300 nM of the unlabelled oligo (U-seal) and 100 nM of 8 nt labelled R-seal. The imaging was performed in imaging buffer 3 (IB3: 20 mM HEPES, 200 mM NaCl, 80 mM $MgCl_2$ of pH 7.7, 1 mM TROLOX, 1% Glucose, 40 μg/ml catalase and 0.1 mg/ml glucose oxidase, 10% dextran sulphate , 10% formamide).

For the 3-base sequencing experiment, a 13-nt DNA gap was constructed by annealing biotinylated sequence (Gap(1-65)-$B^{Bio,1}$), one 27-nt Cy3B labelled strand (Gap(1-27)-$Top^{Cy3B,25}$), and one 25-nt strand (Gap(41-65)-Top). The R-seal was labelled with Atto647N and BHQ1 quencher on the 5' and 3' ends, respectively, and contained three degenerated positions at the 5th, 6th and 7th positions corresponding to the Gap. We performed the 3-base experiments using 500 nM labelled 13 nt degenerated seal (R3-$N^5N^6N^7$) and 1.5 μM of unlabelled competing seals, in IB3 with 15% formamide.

For 5-base sequencing, we use the same 13-nt DNA gap as for 3-base sequencing. A 13-nt R5-seal labelled with ATTO647N and BHQ1 and 5' and 3' end respectively with five degenerated bases at positions 5,6,7,8,9 (R13-$N^5N^6N^7N^8N^9$) was used. Since we used the same 13-nt gap for 5-base sequencing, we used the same buffer compositions as for 3-base sequencing (i.e., IB3 with 15% formamide). The concentrations of R5-seal and competing unlabelled seals were 1 μM and 2 μM, respectively.

Single base sequencing in a mixed population with prior position knowledge Each of the four different 8-nt DNA gaps was sequentially immobilised

and localised on the surface at a concentration of 50 pM. First, we immobilised Gap-A and registered the locations of the Gap-A molecules by exciting with 0.2 mW of 532 nm laser, followed by photobleaching using a 532 nm laser at a power of 2 mW. We then repeated the same procedures for Gap-T, Gap-G and Gap-C. We then interrogated the Gap molecules using a competitive-inhibition approach, using a mixture of 100 nM R-seal and 300 nM of the unlabelled 8-nt oligos (one U-seal at a time), or 100 nM R-seal for a non-competitive approach.

4.2 Single base calling

First, we will present the concept and results of single-base calling using both non-competitive and competitive assays, introduce the analytical method for automatically sequencing the interrogated position, and present the statistical results of the accuracy rates.

4.2.1 Single base calling via non-competitive approach

In the non-competitive experiment, we utilised four seals labelled with Atto647N that differ by one base at a position opposite the interrogated site on the template. Only the seal that is complementary to the template shows significant binding, resulting in the highest fluorescence and the largest number of detectable localisations (4.1A). We focused on the base at position 3 of an 8-nt gapped DNA molecule immobilised on a surface. For the substrate with guanine at the interrogated position (GapG), only the seal with cytosine at position 3 (S8-C^3) exhibited active fluorescence and subsequently showed a high number of localisations. In contrast, seals S8-A^3, S8-T^3, and S8-G^3 demonstrated significantly lower activity (4.1B, top row).

Similarly, when the thymine is at the interrogated position (GapT), the seal with adenine (S8-A^3) will exhibit the strongest binding, compared to other seals (4.1B, second row). Additional examples from GapA, and GapG are presented in Figure 4.1B as well, which are located in the last two rows. As illustrated in

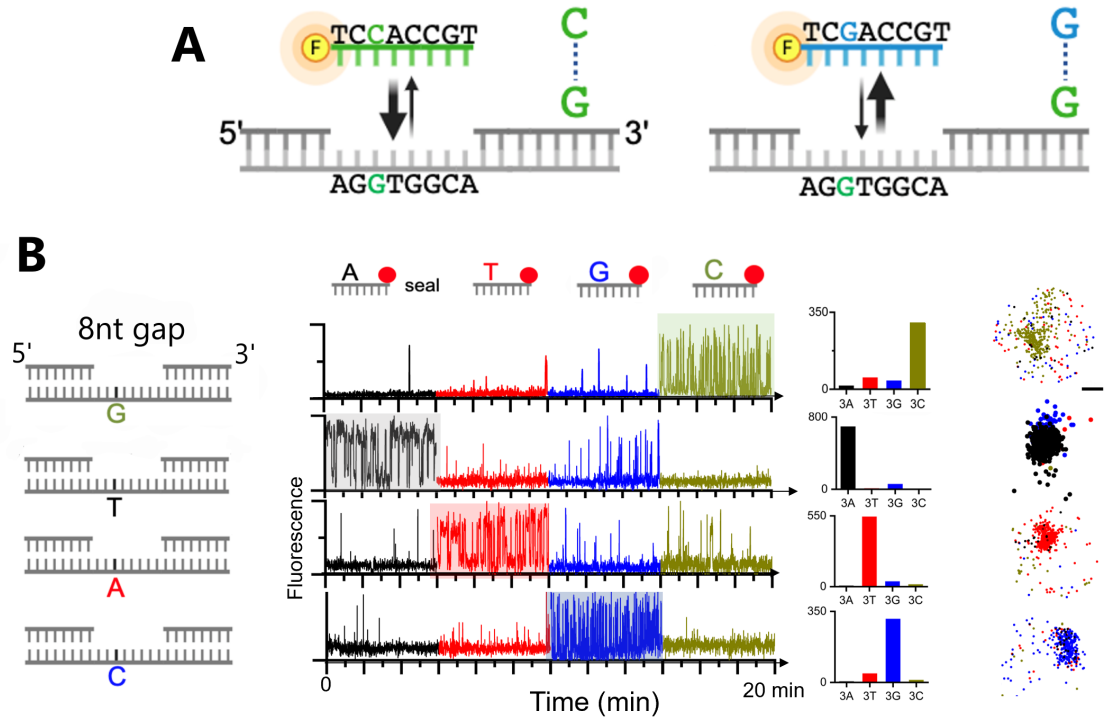

Figure 4.1: concept of non-competitive base calling method and examples of fluorescence data. A: Schematic plots of the non-competitive base calling method. When investigating the third position in the gap, four labelled seals with different nucleotides at the opposite position are used. Compared to the other three seals, the one with a complementary nucleotide at position 3 will exhibit the strongest binding with the gapped substrate and subsequently the strongest fluorescence signals. B: shows the design of gaps and seals for non-competitive single-base calling, examples of the corresponding fluorescence time series, number of detected localisations (parameters for localisation can be seen in Section4.3), and localisations in 2D space from left to right in each row. Colours: green, black, red, and blue represent nucleotide C, A, T, G in seals, and their complementary nucleotide G, T, A, C in gaps. The scale bar is 100 nm.

the scatter plot (4.1B last column), the majority of localizations are clustered around the substrates. This clustering occurs because seals attach to these gapped substrates, leading to the accumulation of localizations and the formation of clusters around them. Conversely, a small fraction of localizations outside the dense area is likely due to noise (unspecific binding).

4.2.2 Single base calling via competitive inhibition

In the competitive experiment, a mixture of DNAs containing degenerated bases (U-seal) at the opposite site of the interrogated position must compete with a

fluorescent reporter seal (R-seal) with a known base at the same site. Since the probability of having any of the four standard bases is equal in a degenerated base (N), only the seal that has the complementary nucleotide to the gap will prevail over the R-seal, suppressing binding of the R-seal to the gap, thus reducing the associated fluorescence (4.2A). In contrast to the non-competitive experiment, the complementary seal in the competitive setup demonstrates the lowest fluorescence activity and the fewest localisations.

We used an Atto647N-labelled oligo of 8nt length with a degenerated base opposite to position 3 of the DNA gap (R-seal; R8-N^3) and four unlabelled oligos of 8 nt length having A (U8-A^3), T (U8-T^3), G (U8-G^3) or C (U8-C^3) at a position opposite to the third base of the gapped DNA. For example, when the integrated position 3 is guanine, U8-C^3 exhibits the lowest fluorescence compared to the other three seals (4.2B, top row). The same results can be observed from other gapped DNA sequences (4.2B, second-to-last rows).

4.2.3 Single base calling on mixture surface

To show our capability to sequence individual bases within a mixed population, we sequentially immobilised four different 8-nt DNA gaps on a surface at a concentration of 50 pM. We recorded their positions and then performed a photobleaching step to eliminate the fluorescence from Cy3B. The gaps were applied to the surface in the following order: GapA, GapT, GapC, and GapG (4.3A).

Next, we interrogated the surface using either a non-competitive (4.3B) or competitive (4.3C) inhibition approach to identify the base of each library molecule, comparing the results with our prior knowledge (the position and identity of each library molecule). Just like the surface, which only contains one type of gapped substrates, the highest and lowest fluorescence occur when seals are complementary in non-competitive and competitive inhibition assays, respectively.

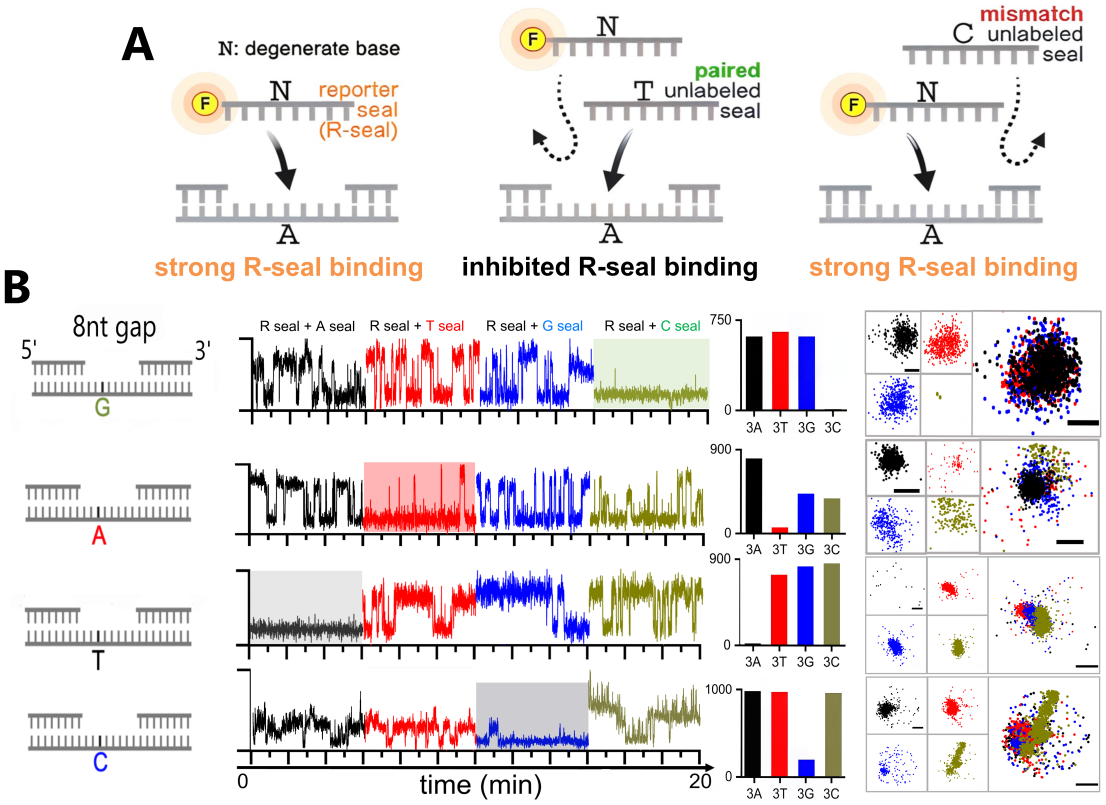

Figure 4.2: concept of competitive inhibition base calling and representative fluorescence signals. A: Schematic plots of the competitive single-base calling method. In each imaging round, U-seals with a known base at the opposite end of the interrogated position and labelled R-seals compete with each other to bind with the gapped DNA. The U-seal will suppress the binding of R-seal when it has the complementary nucleotide. The subsequent decrease in fluorescence will reveal the identity of the integrated base. B: The design of gaps for competitive single-base calling, examples of the corresponding fluorescence time series, number of detected localisations (parameters for localisation can be seen in Section4.3), and localisations in 2D space are laid out from left to right. Colours: green, black, red, and blue represent nucleotide C, A, T, G in seals, and their complementary nucleotide G, T, A, C in gaps. The scale bars are 100 nm.

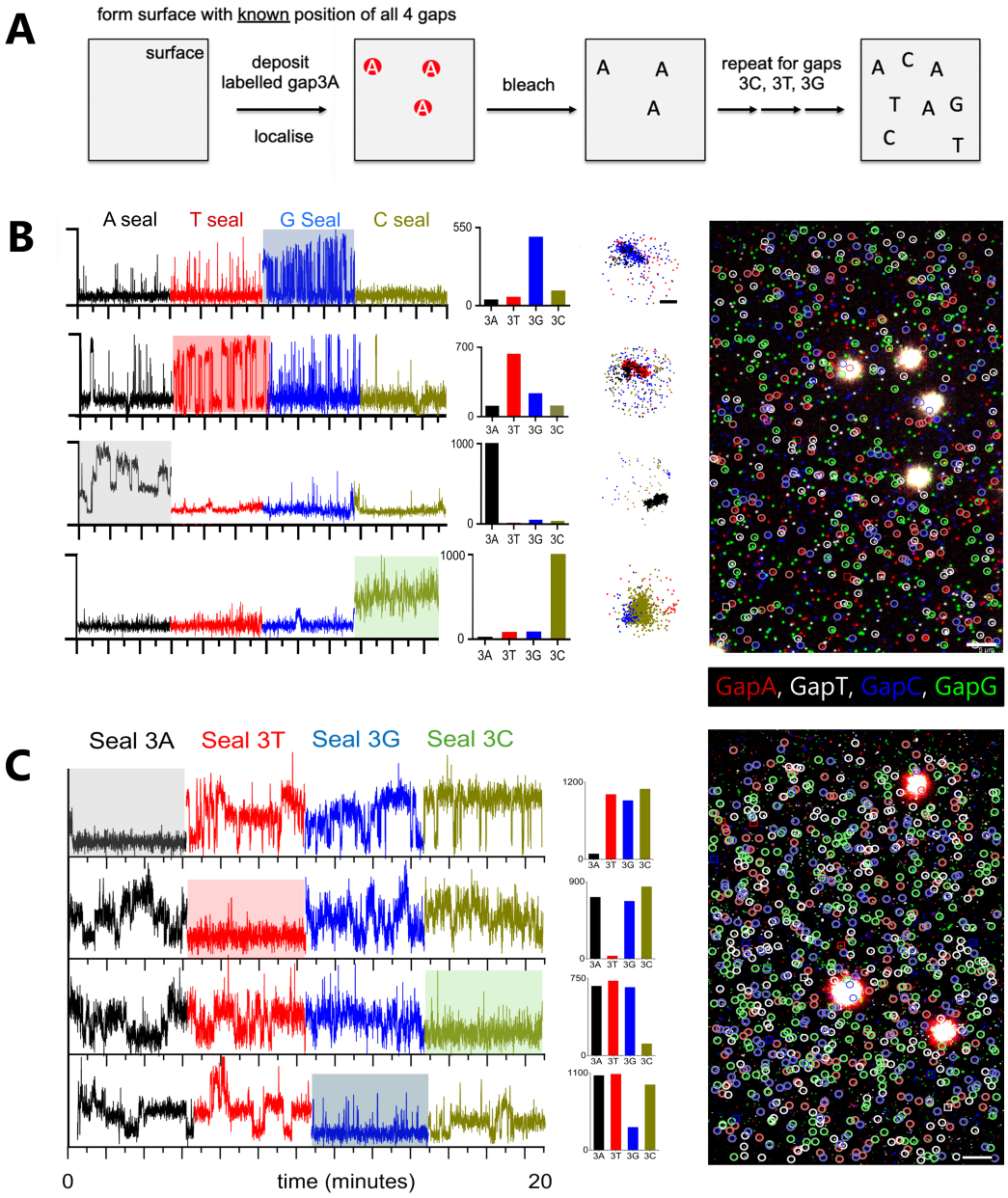

Figure 4.3: single base calling on mixture surface using both non-competitive and competitive methods. A: Schematic plot of the process to create a mixture surface. B: Examples show the binding kinetics of four different DNA substrates and their corresponding base calling results. The overview of sequencing results on a mixture surface using non-competitive method is displayed on the right side of this panel. Circles represent the correct sequencing, and squares indicate the false results. Colours: green, white, red, and blue represent nucleotide C, A, T, G in seals, and their complementary nucleotide G, T, A, C in gaps. C: The similar examples for the competitive inhibition approach. One may notice the presence of circles within the prominent fiducial markers in the images. This phenomenon arises from the enhancement of image contrast, which facilitates the clear visualisation of localizations for the purpose of display. However, this adjustment leads to the occlusion of the relatively weak signals from neighbouring localizations by the fiducial markers. The scale bars in the overview images measure 5 µm, and the scale bar in the scatter plots of localizations is 100 nm.

4.3 Analytical pipeline of base calling

In both competitive and non-competitive cases, we quantify the binding events to identify the unknown base by counting the number of detected localisations in proximity to the library molecules (gapped DNA substrate).

For integration movies, we applied drift correction using the AIM method (segmentation: 100, interaction distance: 20nm, maximum drift in segment: 60 nm) with localisations detected by Picasso (box side length: 5, Min.net gradient: 1000, Baseline: 400, Sensitivity:2.5 Quantum efficiency: 0.82, Pixel size: 117 nm).

Next, movies from different imaging rounds were registered to the gapped DNAs using fiducial markers in the first frame of each movie with a rigid-body transformation. Fiducial markers in different images are selected based on intensity thresholds of 50,000 and 2,000 for the green and red channels, respectively. Only the intensities within a 7×7 window of each fiducial marker were kept and rescaled to 0-255, to remove the impact of other signals and the unwanted weighting effects due to the different brightness of different fiducial markers. The registration matrix was calculated based on the work of Thévenaz, etc [101]. It uses a coarse-to-fine iterative strategy with optimisation carried out using a modified Levenberg-Marquardt algorithm.

To address the issue of lost fiducial markers during the buffer exchange process, we developed a second method for registration using super-resolution images constructed from detected localisations. In this approach, transformation matrices between the red and green channels, as well as between the localisation movie and the initial integration movie, are still calculated using fiducial markers. The calculation of the latter matrix must rely on fiducial markers because the movies recorded for localising gapped DNAs consist of only 10 frames, which is insufficient for creating super-resolution images. After localisation detection, we applied a DBSCAN algorithm (with parameters eps = 117 nm and min_samples = 50)

to eliminate noise. The constructed super-resolution images are then enhanced through a gamma correction to amplify signals from bright pixels, and they are registered using either Thévenaz's method or phase cross-correlation [102]. Phase cross-correlation operates in Fourier space and achieves arbitrary subpixel precision by employing an upsampled matrix-multiplication discrete Fourier transformation.

For each molecule detected (Box side length: 5, Min.net gradient: 400, Baseline: 400, Sensitivity:2.5 Quantum efficiency: 0.82, Pixel size: 117 nm) in the gap substrates, we counted the number of localisations from each integration movie within a radius of 234 nm (2 pixels) from it. This allows us to summarise the interactions between library DNA molecules and different seals into a set of four localisation numbers. Next, we selected the complementary nucleotide from the four standard bases and established a confidence parameter to quantify the differences in fluorescence activities among the four seals. For every gapped DNA, we compared the difference among the four localisation counting (LC) values and defined confidence as:

$$\begin{aligned}
&\text{difference} = \begin{cases} \text{largest count} - \text{second largest count} & \text{non-competitive mehod} \\ \text{second minimum} - \text{minimum} & \text{competitive inhibition} \end{cases} \\
&\text{confidence} = \frac{\text{difference}}{\text{percentile 95 of difference} - \text{minimum of difference}} \\
&\text{clip confidence to the range of 0 - 1}
\end{aligned} \tag{4.1}$$

In this process, we removed substrate molecules with sticky seals by analysing the time distribution of co-localised localisations from all integration movies. We considered localisations in consecutive frames as a bundle. For seals with over 100 localisations, we require at least 5 bundles, ensuring that the longest bundle does not exceed 50% of the total frames for non-competitive sequencing. For competitive assays, we require at least 2 bundles, with the longest bundle being smaller than 80% of the total frames.

Additionally, we determined a minimum LC value for a seal to be considered actively binding. Only molecules that have at least one (for non-competitive method)

or three (for competitive method) LC values exceeding the threshold are considered for base calling. The following steps determine the minimum LC value. First, we pool all the LC values from four integration movies together and plot a histogram. This histogram shows a prominent peak at a low LC value, primarily caused by non-specific binding events. Next, we calculate the first derivative of the histogram and fit an exponential function of the form Ae^{-Bx} using the data from the global minimum onward (x_{start}). This analysis helps us identify the LC value at which the counts stabilise. The minimum number of LC values is determined by the decay length, calculated as $x_{start} + \frac{3}{B}$. This filtering step does not affect the results in the high-confidence range but improves the accuracy rate in the low-confidence region.

By applying the pipeline above, we can achieve an over 99% accuracy rate at a high confidence range at the cost of losing a significant number of molecules. The overall accuracy for single-base calling is 97% when the confidence level is set at 0.2, with the specific accuracy rates for each experiment listed in Table 4.2.

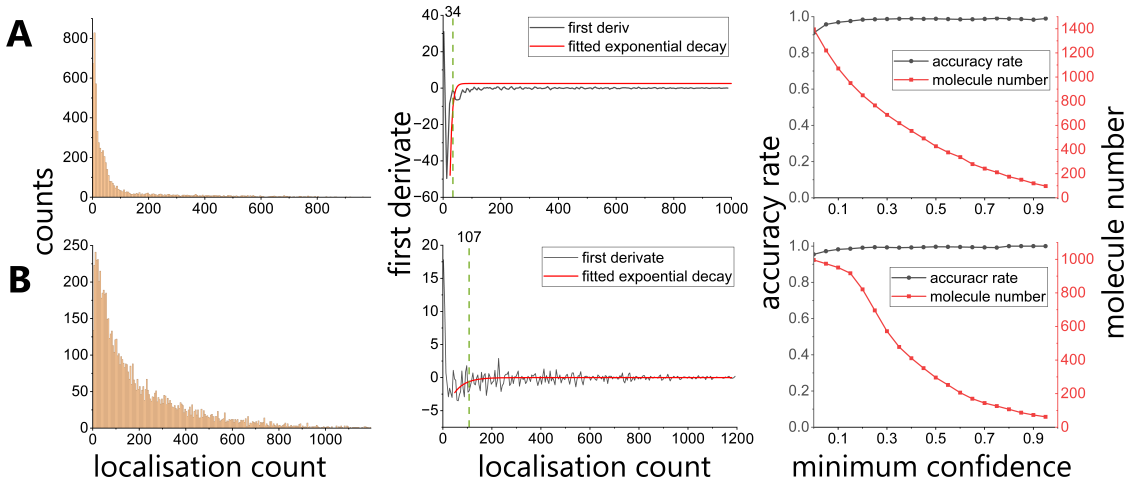

Figure 4.4: analytic pipeline for automatic base calling. This plot illustrates the process of determining the minimum number of localisations needed for a library molecule to be considered as having active binding events with seals. It also shows how accuracy and the number of molecules change with increasing confidence levels. In each panel, moving from left to right, are the following elements: a histogram of LC values for all four seals, the first derivative of the histogram with a red line indicating the fitting result, a green line representing the threshold, and a plot of how the accuracy rate and the number of molecules change as the minimum confidence value increases. A corresponds to non-competitive assay sequencing of GapG, while B displays the same data for the competitive assay of GapG.

4.4 Three- and five-base sequencing by competitive inhibition

Next, we extended the competitive inhibition assay to sequence multiple bases using degenerated bases (equal probability of having any of the four standard bases) to cover the unknown sequence context around the interrogated position. The unknown bases are identified one at a time. During each round, we conducted four tests using unlabelled seals, with only one known base positioned at the targeted location while degenerated bases filled the other positions. This method helps us tackle the challenges of unknown sequence contexts, reduce the cost of the fluorophore, and eliminate the complications associated with using spectrally separated fluorophores.

We constructed the same 13nt gapped DNAs by annealing a biotinylated sequence (Gap(1-65)-$B^{Bio,1}$) with two flanking strands for both 3 and 5 base

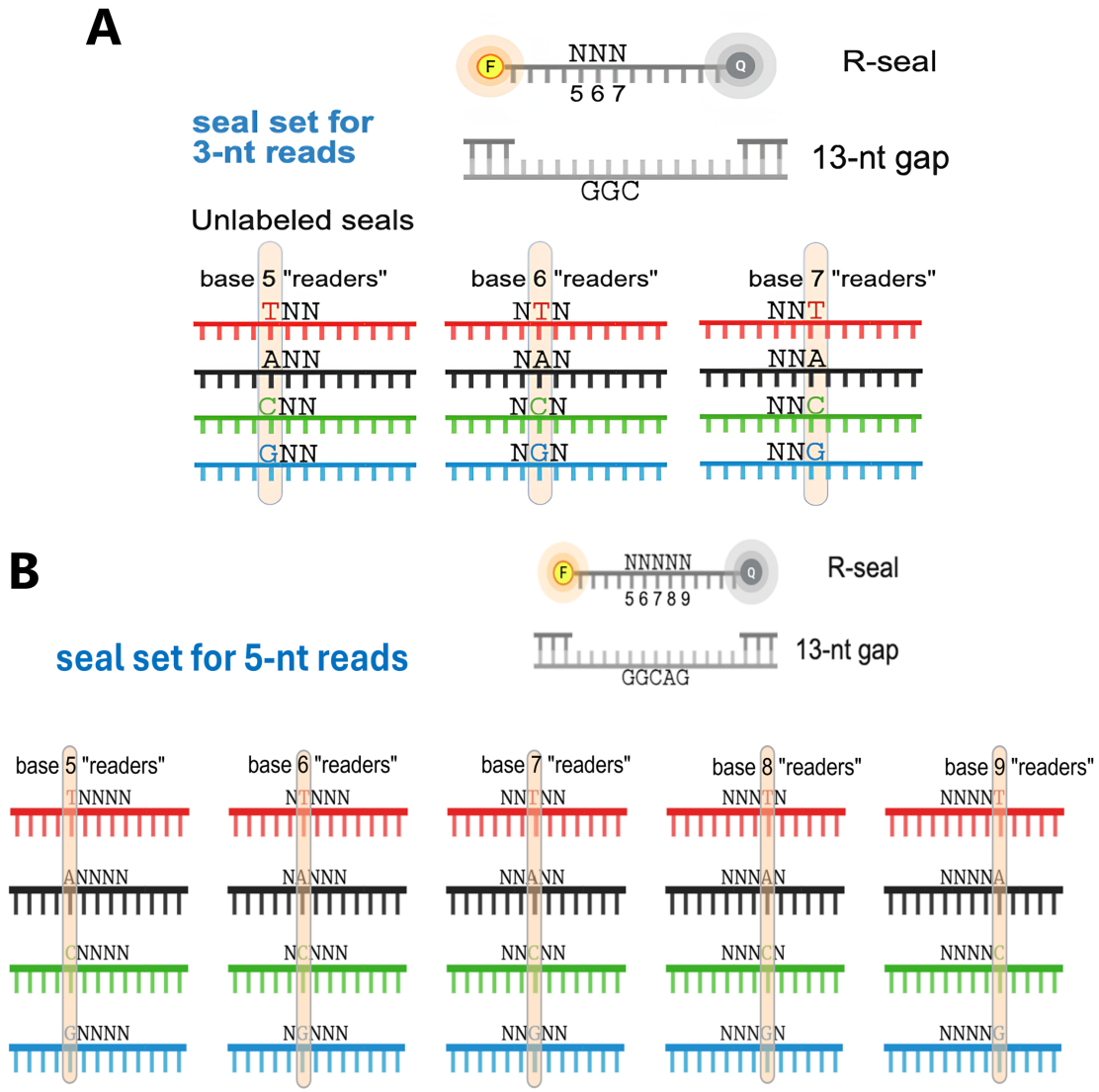

Figure 4.5: DNA design for three- (A) and five-base (B) sequencing assays. The same 13nt gapped DNA was constructed for both experiments. Both R-seals are labelled with Atto647N and BHQ1, but they contain different degenerated bases (indicated as N). The probability of having any of the four standard bases is the same at degenerated bases.

sequencing. One flanking strand was complementary to the first 27 bases of the biotin strand and labelled with Cy3B (Gap(1-27)-Top$^{\text{Cy3B,25}}$), while the other strand was complementary to bases 41-65 of the biotin sequence (Gap(41-65)-Top). For the three-base sequencing, we used the 13 nt R3-seal (R13-N^{5}N^{6}N^7). This seal contained three degenerated positions at the 5th, 6th, and 7th positions, corresponding to the bases being sequenced in the gap (4.5A). For the five-base sequencing, we utilised a 13 nt R5-seal (R13-N^{5}N^{6}N^{7}N^{8}N^9) featuring five degenerated bases at positions 5,

6, 7, 8, and 9 (4.5B). Both R3-seal and R5-seal seals are labelled with Atto647N and BHQ1 at the 5' and 3' ends, respectively. BHQ1s were added to the labelled R-seals to reduce the background that comes with high imager concentrations.

Compared to the 8nt seal used for single base calling, binding of the 13 nt seal will lead to longer dwell times, which in turn inhibits sampling due to decreased binding on-off cycles. We thus decreased the binding stability of the 13-nt seal using formamide. Furthermore, the R3-seal and R5-seal, having 3 and 5 degenerated bases, lead to the exponentially decreased (64-fold and 1024-fold lower than the mixture concentration used) effective concentration of an entirely complementary seal sequence in the mixture. To obtain enough binding, we increased the R3-seal and R5-seal concentrations to 500 nM and 1 μM, respectively, and the concentration for unlabelled seals was 1.5 μM and 2 μM. (The choice of concentrations is established in section 4.5.) It is not to say that only seals with all matching bases provide binding events. There is a chance that for a binding event, the R-seal with an unmatching base at the integrated position will have relatively strong binding due to the matching of positions that are not under integration. Thus, it is critical to ensure there is a large number of binding-unbinding cycles between seals and gapped substrates to ensure that multiple seals join the competition to attach to the gap so that the statistical strength of degenerated bases will be averaged out.

The fluorescence time profile and base calling results obtained using the LC counting method are illustrated in Figure 4.6. In the multi-base experiment, we achieved an overall accuracy rate of 97% as well at the confidence level of 0.2. Details on the specific accuracy rates for each experiment and gap can be found in Table 4.2.

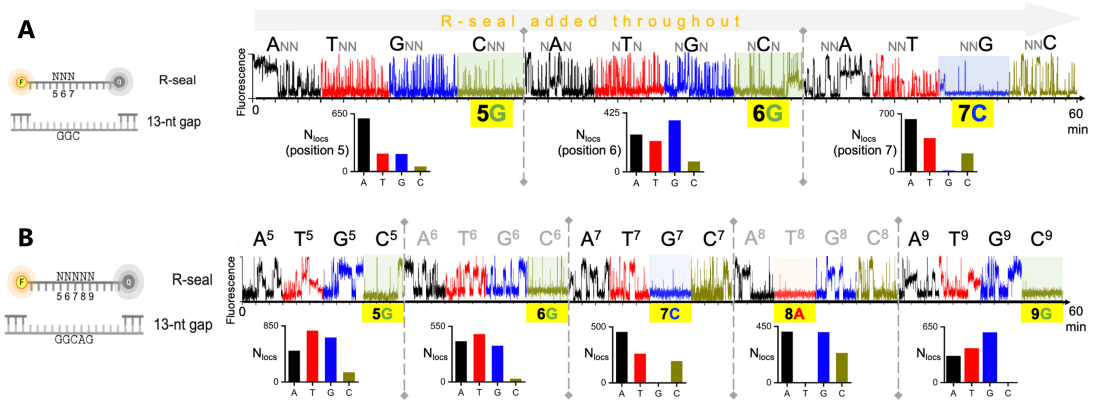

Figure 4.6: representative fluorescence data from multiple base sequencing assays. A and B are representative traces and LC values of three and five-base sequencing using competitive inhibition, respectively. The sequencing result is marked in the yellow boxes at the bottom of the time series.

single base calling				
	non-competitive		competitive	
Base	individual gaps	mixed gaps	individual gaps	mixed gaps
A	93.7(655/699)	97.8(184/188)	94.2 (274/291)	97.9 (184/188)
T	95.6(1464/1531)	98.7(243/246)	97.2 (383/394)	98.8 (243/246)
G	97.9(887/906)	99.6(224/225)	99.6 (500/502)	99.6 (224/225)
C	99.0(1799/1817)	97.0(190/196)	96.6 (736/762)	96.9 (190/196)
overall	97.0(4805/4953)	96.7(841/855)	97.1 (1893/1949)	98.5 (841/855)

multiple base calling		
base number / position	3 bases	5 bases
G^5	96.4 (240/249)	95.1 (195/205)
G^6	98.1 (256/261)	97.9 (464/474)
C^7	95.0 (489/515)	97.1 (438/451)
A^8	NA	94.9 (352/371)
G^9	NA	97.1 (233/240)
overall	96.5 (765/793)	96.6 (1682/1741)

Table 4.2: accuracy rates of all base calling methods. This table summarises the accuracy rates at a confidence level of 0.2 from all assays, including both competitive and non-competitive methods, as well as single and multiple-base sequencing. The accuracy rates range from 94% to over 99%, which demonstrates the validity of our sequencing method. One can get a higher accuracy by setting higher confidence value at the cost of losing some molecules.

4.5 Standardisation of competitive inhibition

The free energy from the interaction between short ssDNAs and gapped DNAs can be adjusted through various experimental conditions, such as sodium concentration, temperature, length, and CG content. We observed changes in the LC values with different concentrations of formamide and U-seals to standardise and optimise their concentrations. An ideal formamide concentration should reduce the stability of the duplex to an appropriate level to provide enough on-off cycles, which enables sufficient competition between R-seals and U-seals, and maintain proper binding between complementary seals and the gapped substrates, so that it can suppress the R-seal and be distinguishable from non-complementary seals.

For U-seals, an appropriate concentration should provide sufficient competition with R-seal, allowing it to effectively suppress fluorescence from R-seal when complementary and prevent excessive non-specific binding that could introduce noise and affect the LC values.

Standardisation of formamide concentration We standardised the choice of formamide concentration for multiple-base sequencing by tracking the binding of 500 nM R3-seal(R13-$N^5N^6N^7$) to a 13-nt Gap in the presence of varying formamide concentrations (10%, 15%, 20% and 25% (v/v)) in the IB3. We observed a negative linear relationship between formamide concentration and the average LC value within the tested range (4.5A). We chose 15% for further experiments to balance the need to maintain a significant number of detectable localisations for R-seals and the need to destabilise the 13-nt seal and increase the total number of binding-unbinding cycles.

Standardisation of U-seal concentration for single base calling We employed the 8nt configuration of GapG (G at position 3). R-seal concentration was kept at 100nM, while the concentration of unlabelled complementary (U8-C^3, C at position 3) and non-complementary(U8-T^3, T at position 3) seals varied from

100 nM to 800 nM. We observed a linear decrease in the average LC values with the increase in both complementary and non-complementary seals. The average LC value of non-complementary seals is significantly higher compared to complementary seals at low concentrations. However, it decreases at a faster rate and intersects with the values from complementary seals at a high concentration of approximately 700nM (4.7B), which means the choice of U-seal concentration must be kept below and away from 7 times the R-seal concentration. We eventually chose 300nM to balance the need to maximise the difference between complementary and non-complementary seals and sufficient competition between R-seal and U-seal.

Standardisation of U-seals for three- and five-base sequencing For standardising the concentration of U-seals to sequence three bases, we kept the R3-seal concentration at 500nM while the concentrations of unlabelled complementary seal (U13-$C^5N^6N^7$) and non-complementary seal (U13-$T^5N^6N^7$) varied between 1 to 3.5 μM. For five bases, we used complementary seal (U13-$C^5N^6N^7N^8N^9$) and non-complementary seal (U13-$T^5N^6N^7N^8N^9$) with concentration from 1 μM to 5μM, while the R5-seal concentration was kept at 1μM. The LC values decreased linearly with the increase of complementary seal concentrations for both 3-base and 5-base scenarios. In comparison, the LC value fluctuates with the concentration of the non-complementary seal (Fig. 4.7C, D). This indicates that the competitive advantage associated with high concentrations reaches its limit. It is noticeable that the distinction between complementary and mismatched seals is significant across tested concentrations. Finally, we chose 1.5μM and 2μM to conduct sequencing for three and five-base assays, ensuring adequate exchange between R-seals and U-seals and fewer unspecific binding events.

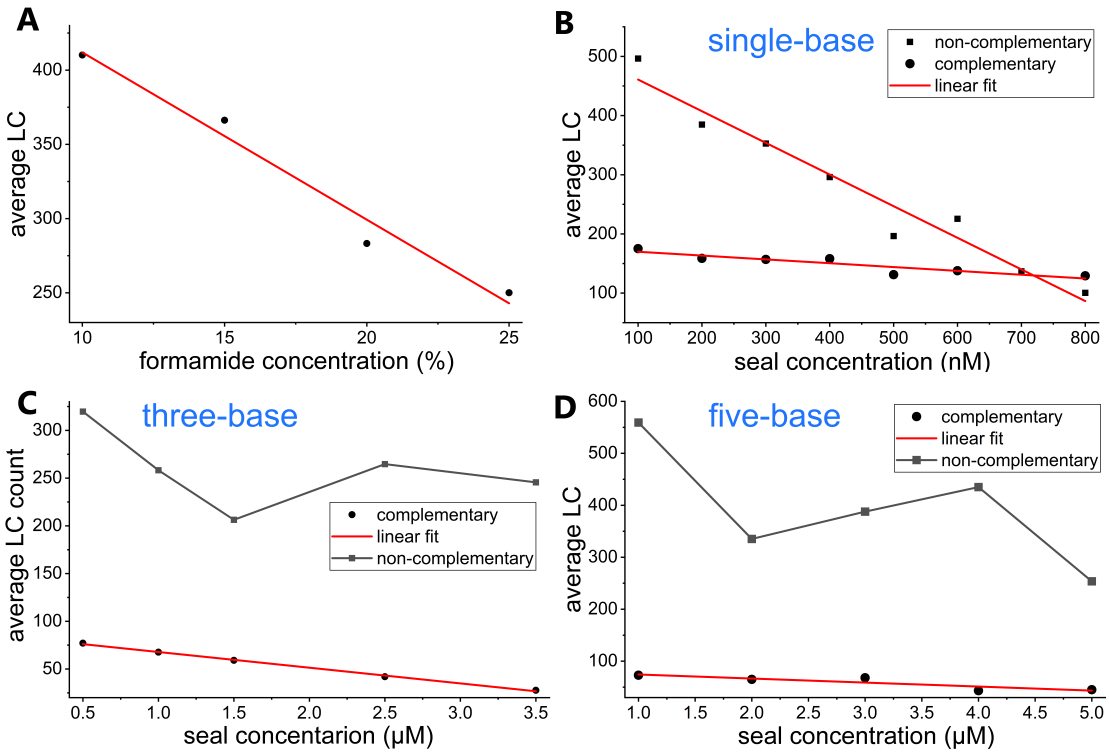

Figure 4.7: standardisation of competitive inhibition assay. Graphs show average LC values at varying concentrations of formamide (A), unlabelled non-complementary and complementary seals in one-base (B), three-base (C), and five-base (D) competitive experiments with the red lines presenting the results from the linear fit.

5

Single-molecule phenotyping and sequencing

Single-molecule techniques are particularly suitable for studying dynamic processes because they avoid ensemble averaging, which can obscure the observation of intermediate states, alternative kinetic pathways, and subpopulations. However, these single-molecule assays can only analyse one sample or condition at a time, making them time-consuming and labour-intensive. Recently, there have been significant advancements in combining single-molecule fluorescence microscopy with single-molecule sequencing. These innovative approaches enable parallel, multiplexed observations of a large number of individual molecules covering a broad sequence space [103, 104]. They have been successfully applied to investigate sequence-dependent activities, such as Cas9-DNA interactions [105], and the kinetics of Holliday junctions [106]. Despite their potential, these methods typically rely on Illumina sequencing and custom-built devices, which can be expensive and not very accessible.

Building on the sequencing method we established in Chapter 4, we developed a sequencing technique to link the kinetic properties of molecule-DNA interactions to the DNA sequence at the single-molecule level. We named this technique Single-

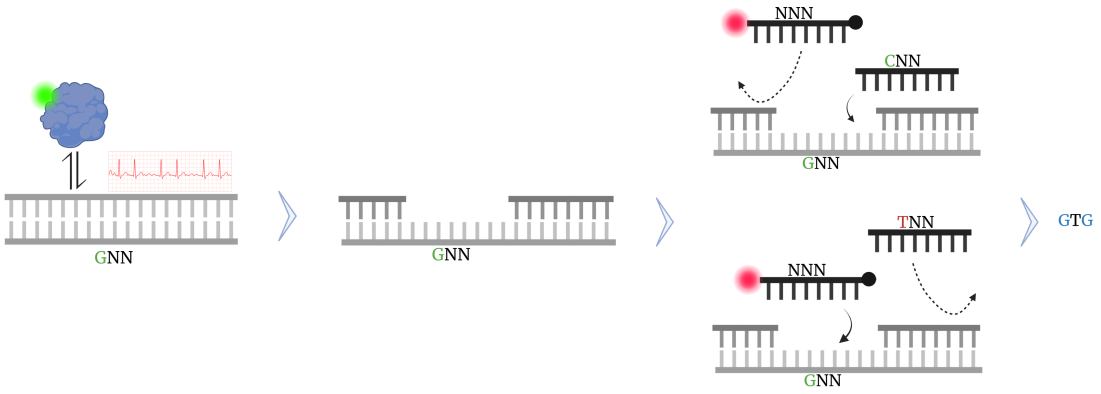

Figure 5.1: abstract of SPIN-Seq method. This plot summarises the concept of the SPIN-Seq method. After recording the interactions between the molecules of interest and DNA, the gapped DNA substrates are constructed and sequenced using the labelled R-seals and competitive U-seals. Therefore, we successfully link the kinetic behaviours of biomolecules with DNA sequence on the single-molecule level, which will aid the study of a variety of sequence-dependent biological processes.

Molecule Phenotyping and In-Situ Sequencing (SPIN-Seq), which can be applied to study sequence-dependent processes using a standard wide-field microscope on a standard coverslip (5.1). We applied the SPIN-Seq method to investigate the E. coli catabolite activator protein (CAP)-consensus DNA interactions and the sequence dependence of bacterial transcription initiation.

To study the phenotype, we applied time series analysis to the fluorescence time profiles extracted from the positions of target molecules. To obtain the fluorescence traces, we constructed an inner circle surrounded by a buffer zone and an outlier circle. The fluorescence signal is quantified as the median value obtained from within the inner circle, adjusted by subtracting the background, which is computed as the median value from the region outside the buffer zone but within the confines of the outlier circle. We employed different time series analysis methods for two experiments, which will be discussed in the corresponding section later.

5.1 Phenotype of CAP on consensus DNA with single-base mutants

CAP is also known as the cyclic Adenosine monophosphate (cAMP) receptor. It is a transcription factor in bacteria that exists as a homo-dimer in solution, with each subunit including a ligand-binding domain at the N-terminus and a DNA-binding domain at the C-terminus. Two cAMP molecules bind dimeric CAP and function as allosteric effectors by increasing its affinity for DNA. In the presence of cAMP, CAP binds to specific DNA sites in or near target promoters and enhances the ability of RNA polymerase holoenzyme to bind and initiate transcription [107]. Transcription activation by CAP serves as a classic model for studying both the structure and the mechanism involved in the process.

CAP recognises DNA sites through a combination of indirect (mediated by sensing of DNA-sequence dependent effects on DNA phosphate position and solvation, or susceptibility to DNA deformation) and direct (mediated by direct hydrogen-bond or van der Waals interactions) readout [108]. CAP interacts with a 22bp sequence 5'-AAATGTGATCTAGATCACATTT-3' [109, 110] called CAP consensus DNA (CAPcons or consDNA) with key positions underlined (5.2A bottom). Sequence preference at three (positions 5, 7 and 8) of these seven positions within each half of DNA is accounted for by direct amino acid-base contact [111–113].

The CAP-DNA complex exhibits two-fold symmetry, where one subunit of CAP interacts with one half of the DNA site while the other subunit interacts with the opposite half. CAP bends the DNA by approximately 80°, wrapping it around the sides of the CAP dimer. This bending is localised at two distinct kinks in each half site: a primary kink of about 40° situated between positions 6 and 7, and a secondary kink of approximately 9° located between positions -1 and 2 [114] (5.2A top).

experiment type		laser mode	frame number	exposure time (ms)	laser power (mW)	excitation wavelength(nm)
CAP-DNA interaction	CAP binding	continuous	1200	200	0.8	638
	base calling	continuous	1800	250	0.6	638

Table 5.1: imaging conditions of CAP-DNA assay. This table shows the imaging conditions of CAP-DNA interactions, including laser mode, frame number, exposure time and laser power.

Here, we investigate the sequence-dependent kinetics of CAP-DNA binding with CAP consensus DNA, which is mutant at position 5 where the guanidinium sidechain of Arg180 forms hydrogen bonds with the guanine O^6 and N^7 atoms of the consensus base pair G•C [113]. We calculated the dwell time, unbound time, and dissociation rates of CAP-DNA binding events and categorised them based on the sequence identified by base calling.

5.1.1 Experiment protocol of CAP-DNA interaction

Dr Jagadish Hazra created the following protocols for CAP binding and conducted the related experiments. I set up the data analysis pipeline described in this section. All the preprocessing steps, including drift correction and movie registration, were conducted in the same way described in Chapter 4 (4.3).

The complete set of DNA sequences (purchased from Biomer Corp), along with their corresponding strand designations, is presented in Table A.2. The protocol for the preparation of NeutrAvidin-biotin-PEG/mPEG functionalized coverslips is detailed in Section 2.4.

All single-molecule measurements were performed using the Nanoimager microscope (Oxford Nanoimaging), equipped with a 100X, 1.4 NA objective, and a sCMOS camera (Orca flash4 v3). The TIRF mode was applied at an illumination angle of 56°. The emission channel was divided into two using a dichroic 640 LP splitter. One is for wavelengths ranging from 498 to 551 nm and 550 to 620nm. The other is for wavelengths 685/40 nm. The imaging conditions are listed in Table 5.1.

Preparation of labelled CAP derivative (Alexa647-CAP) Plasmid pHSCRP-His6H17CC178S, used to express CAP-(Cys17, Ser178), was constructed from pAKCRP-His6 using site-directed mutagenesis with the Q5 Site-Directed Mutagenesis Kit from NEB [115]. Then CAP was purified using the Ni-NTA system, followed by gel-filtration chromatography equipped with Superose®6 Increase 10/300 GL. A 12-hour incubation of 20 μM CAP protein with 400 μM Alexa647 maleimide in buffer A (40 mM HEPES-NaOH (pH 7.3), 100 mM NaCl, 2 mM TCEP and 5% glycerol) on ice for fluorescence labelling. Then we passed the solution through another round of Ni-NTA purification and elution using 500 mM imidazole in buffer B (40 mM Tris-HCl, pH 8, 100 mM NaCl, 0.1 mM TCEP and 5% glycerol). Then we performed buffer exchange of the eluted solution using a 3 KDa MWCO Amicon Ultra-15 centrifugal ultrafilter (Millipore). The purified Alexa647-CAP was stored in aliquots at -80°C.

Library preparation for CAP-DNA interaction We constructed a DNA library of CAP-binding consensus sequences with a mutant at position 25. This library was prepared by annealing 1 μM of biotinylated single strand containing 49 nucleotides and a degenerated (equal probability of having any of the four standard bases) position at 25th nucleotide (CAPcons-(1-49)-N^{25}-$B^{Bio,49}$), 2 μM of Cy3B-labelled 20nt ssDNA sequence (CAPcons-(1-20)-$Top^{Cy3B,11}$) which is complementary to the 30th-49th bases of the biotinylated strand. Next, we conducted a primer extension reaction to complete the sequence using Q5 DNA Polymerase (Thermofisher Scientific) and dNTPs (NEB), while maintaining the melting, annealing, and extension temperatures at 95°C, 62°C, and 72°C, respectively. We purified this library using the QiAquick purification kit (QIAGEN). Next, we performed gel electrophoresis using 20% bi-acrylamide-acrylamide and the Typhoon gel scanning system. The band corresponding to the 49nt dsDNA was then excised from the gel and soaked in TE buffer overnight at room temperature. The product was subjected to a concentration process, which begins by mixing the product with three times its volume of ice-cold ethanol, 1 μL of glycogen, and 20 mM sodium

acetate. We kept this mixture at -20°C for 4 hours, before centrifugation at 4000 rpm for 2 mins. The supernatant was removed, and the precipitated product was dissolved in 50 μL assay buffer (AB: 20 mM HEPES, 80 mM $MgCl_2$, 200 mM NaCl of pH 7.7). The final DNA concentration was determined using a Nanodrop UV-Vis spectrophotometer.

Sample preparation for real-time CAP-DNA interactions The mutant CAPcons were immobilised by incubating 40 μL 50pM DNA on a PEGylated surface, followed by one wash using AB and two washes using CAP-binding buffer (0.2 mM cAMP, 40 mM Tris pH 8, 100 mM KCl, 10 mM $MgCl_2$, 5% glycerol). Next, we added 1.75nM Alexa647-CAP and recorded the binding in cap-binding buffer with oxygen scavenging system (1 mM TROLOX, 1% Glucose, 40 μg/ml catalase and 0.1 mg/ml glucose oxidase).

In-situ DNA gap preparation and sequencing To sequence the library after the CAP-binding assay, we constructed a 9-nt gap with the investigated position (25th) in the middle. The surface was washed three times with Milli-Q (MQ) water before applying 20 mM NaOH for 30s to de-hybridise the dsDNA. Next, we applied another three washes using AB. 2 μM flanking DNAs CAPcons-(30-49)-Top and CAPcons-(1-20)-Top^{Cy3B,11} were added and incubated on the surface for 15 min in AB supplemented with 10% dextran sulphate. After the formation of gapped DNA, we performed a competitive-inhibition sequencing assay, using 200nM R-seal labelled with Cy5 (R9-CAP-N^5) and 600 nM four unlabelled seals (U9-CAP-A^5, U9-CAP-T^5, U9-CAP-G^5, U9-CAP-C^5) in IB3.

5.1.2 Kinetics of CAP-consDNA interactions

Here, we look into the CAP binding with the one-base mutant at position 5 on one half of the consensus sequences. All four pairs (T•A, A•T, C•G (consensus pair), and G•C) are included. We recorded the interactions between DNA and

CAP labelled with Alexa Fluor 647, then removed the top strand using NaOH and constructed 9nt gaps by annealing with two flanking strands for our gap sequencing method (5.2B). The sequencing was conducted using the competitive inhibition method using R-seal labelled with Cy5 (R9-CAP-N^5) and U-seals (U9-CAP-A^5, U9-CAP-T^5, U9-CAP-G^5, U9-CAP-C^5). The examples of fluorescence time profile from CAP-DNA binding with segmentation results and their corresponding integration traces and LC values are displayed in Figure5.2C, D, E, respectively.

The analysis of the fluorescence time profile was conducted using a hybrid method, which involves calculating the predictive mean based on Bayesian statistics [117] (step 1) and applying the PELT algorithm to the predictive mean for change point detection (step 2). Here, we describe the calculation of the predictive mean, and the PELT algorithm is introduced in Section 3.1.1.

Let $x_t \in \mathbb{R}^d$ denote the t-th observation in a data series and $x_{a:b}$ denote the contiguous set of observations between time a and b inclusive. We assume the sequence $x_{1:T}$ can be divided into non-overlapping product partitions ($\rho = 1, 2, ...$) [118]. Data within each partition are independent and identically distributed (IID) from the probability distribution $P(x_t|\eta_\rho)$. The hyperprior parameter η_ρ is IID as well. We estimated the posterior distribution since the last change point (the current run length). The predictive distribution conditional on a given run length r_t and the marginal predictive distribution is calculated by integrating over the posterior distribution [117].

$$P(x_{t+1}|x_{1:t}) = \sum_{r_t} P(x_{t+1}|x_t^{(r)})P(r_t|x_{1:t}) \tag{5.1}$$

where $x_t^{(r)}$ indicates the set of observations associated with run length r_t. The posterior distribution

$$P(r_t|x_{1:t}) = \frac{P(r_t, x_{1:t})}{P(x_{1:t})} \tag{5.2}$$

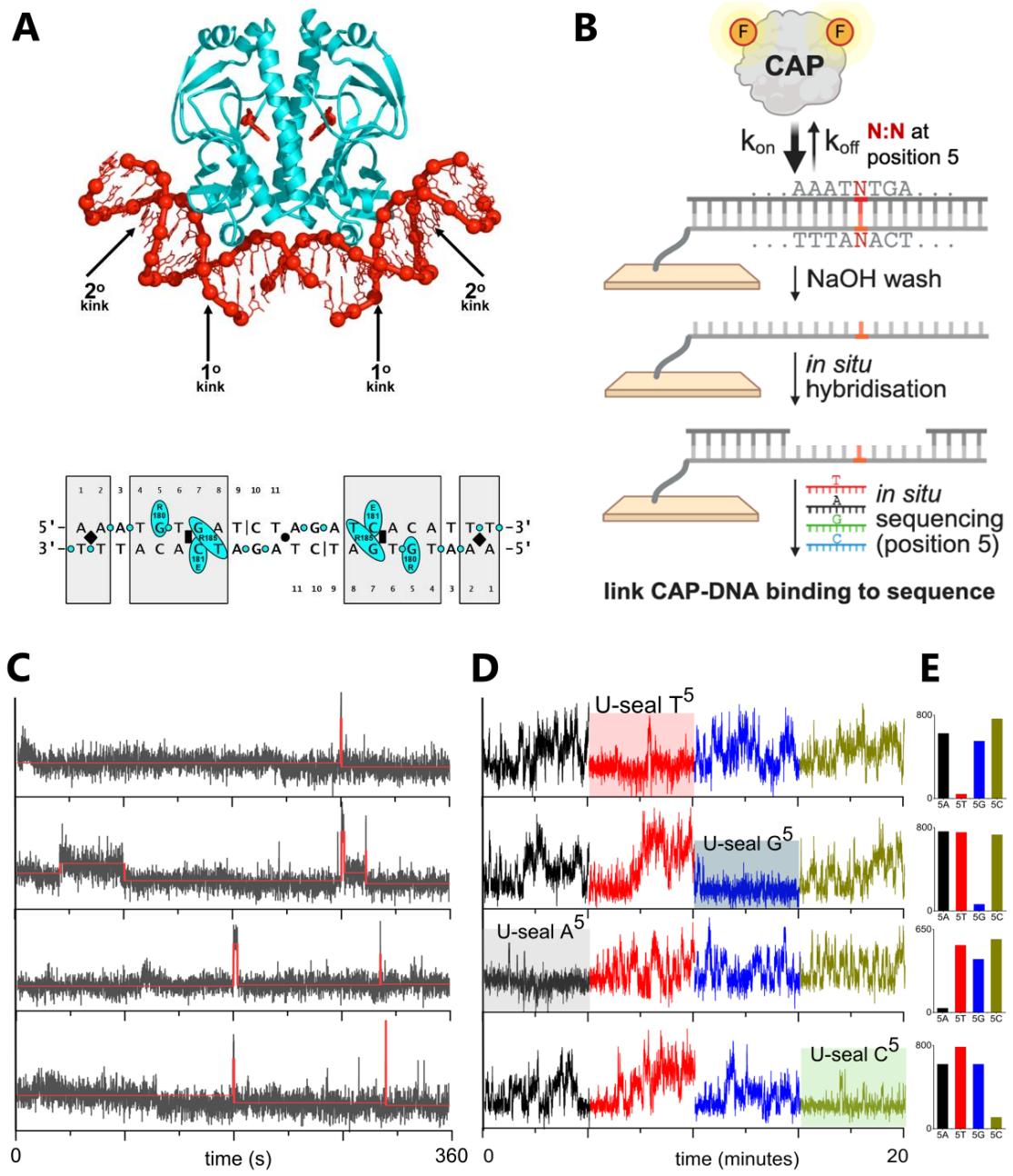

Figure 5.2: Linking kinetics of CAP protein-consDNA interactions to DNA sequence. A: Structure of CAP in complex with its consensus DNA sites. The bottom sequence is the summary of CAP-DNA interactions. Shaded boxes indicate positions where CAP exhibits strong sequence preference. The black vertical lines indicate the positions of single-phosphate gaps present in the crystallisation DNA fragment. The cyan circles and cyan ovals indicate amino acid–phosphate and amino acid–base contacts, respectively. Black rectangle, diamonds, and circles indicate the primary kink sites, the secondary kink sites and the twofold-symmetry axis, respectively. B: Summary of SPINSeq assay to study the CAP-consDNA interaction. Step 1: recording CAP interacting with immobilised CAPcons DNA. Step 2: Denaturing dsDNA, followed by the synthesis of gapped DNA via in situ hybridisation of two DNA strands. Steps 3: Base calling for DNA at position 5 in the gapped substrate. Examples of fluorescence profiles (C) with red lines illustrating the trace analysis results and their corresponding traces from the same molecule in sequence integration movies (D) are also provided, along with LC values (E). (A adapted from [116])

is updated recursively by a message-passing algorithm for joint distribution over current run length and data.

$$\begin{aligned} P(r_t, x_{1:t}) &= \sum_{r_{t-1}} P(r_t, r_{t-1}, x_{1:t-1}) \\ &= \sum_{r_{t-1}} P(r_t, x_t | r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \\ &= \sum_{r_{t-1}} P(r_t | r_{t-1}) P(x_t | r_{t-1}, x_t^{(r)}) P(r_{t-1}, x_{1:t-1}) \end{aligned} \tag{5.3}$$

The conditional prior on change point $P(r_t|r_{t-1})$

$$P(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}+1) & \text{if } r_t = 0 \\ 1 - H(r_{t-1}+1) & \text{if } r_t = r_{t-1}+1 \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

as r_t can only increase by one or drop to zero, depending on whether time t is a change point. $H(\tau)$ is the hazard function,

$$H(\tau) = \frac{P(g=\tau)}{\sum_{t=\tau}^{\inf} P(g=t)} \tag{5.5}$$

$P(g)$ is a discrete priori probability distribution over the interval between change points. When $P(g)$ is a geometric distribution with time scale λ, $H(\tau) = 1/\lambda$ is called memoryless because the hazard function does not depend on time. Here, we chose λ equal to the movie length to cope with the small number of binding events and the traces extracted from false library molecules, which have no change point. We used a Gaussian model (one of the conjugate-exponential models) as the underlying statistical model for generating the data within a run length. Exponential family likelihoods enable inference with a finite number of sufficient statistics, making them very convenient for integration with this scheme.

To address the change points indicated by the resetting of the run length, selecting the timestamp where $r_t = 0$ has the maximum probability is too sensitive for our data, and establishing an appropriate threshold for probability is not a straightforward process. Thus, we applied the PELT algorithm to the predictive mean to detect the change point, as it effectively smooths out the noise (5.3).

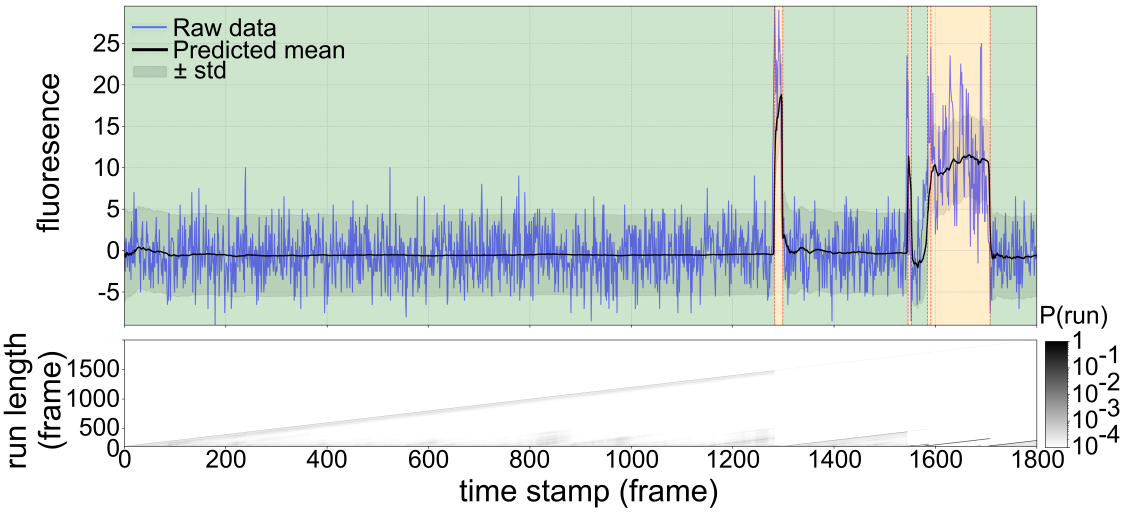

Figure 5.3: illustration of change point detection method. In the upper half, the blue line represents the raw data, the black line indicates the predictive mean, and the red dashed line marks the change points detected by PELT. The yellow and green backgrounds differentiate between the segments considered bound and unbound, respectively. The lower plot displays the posterior probability of the current run, $P(r_t|x_{1:t})$, at each time step, with a logarithmic colour scale.

Next, we applied this trace analysis method to the fluorescence time profiles extracted from the CAPcons library. The results of dwell and unbound time distribution are displayed in Figure 5.4. The distributions of dwell time were fitted with a two-component exponential decay, and the unbound distributions were fitted with a single exponential decay.

The affinity properties were quantified using the binding and unbinding rates, as well as the dissociation constant, K_d, which was calculated using formulas:

$$\begin{aligned} k_{off} &= \frac{1}{\tau_{on}} \\ k_{on} &= \frac{1}{[E] \times \tau_{off}} \\ K_d &= \frac{k_{off}}{k_{on}} \end{aligned} \tag{5.6}$$

where $[E]$ is the concentration of CAP, τ_{on} and τ_{off} are the binding and unbound time respectively. The specific kinetic parameters for each sequence are presented in Table 5.2. Among the four variants, the consensus sequence (C•G) demonstrates the shortest unbound time, the longest dwell time, and the subsequent smallest

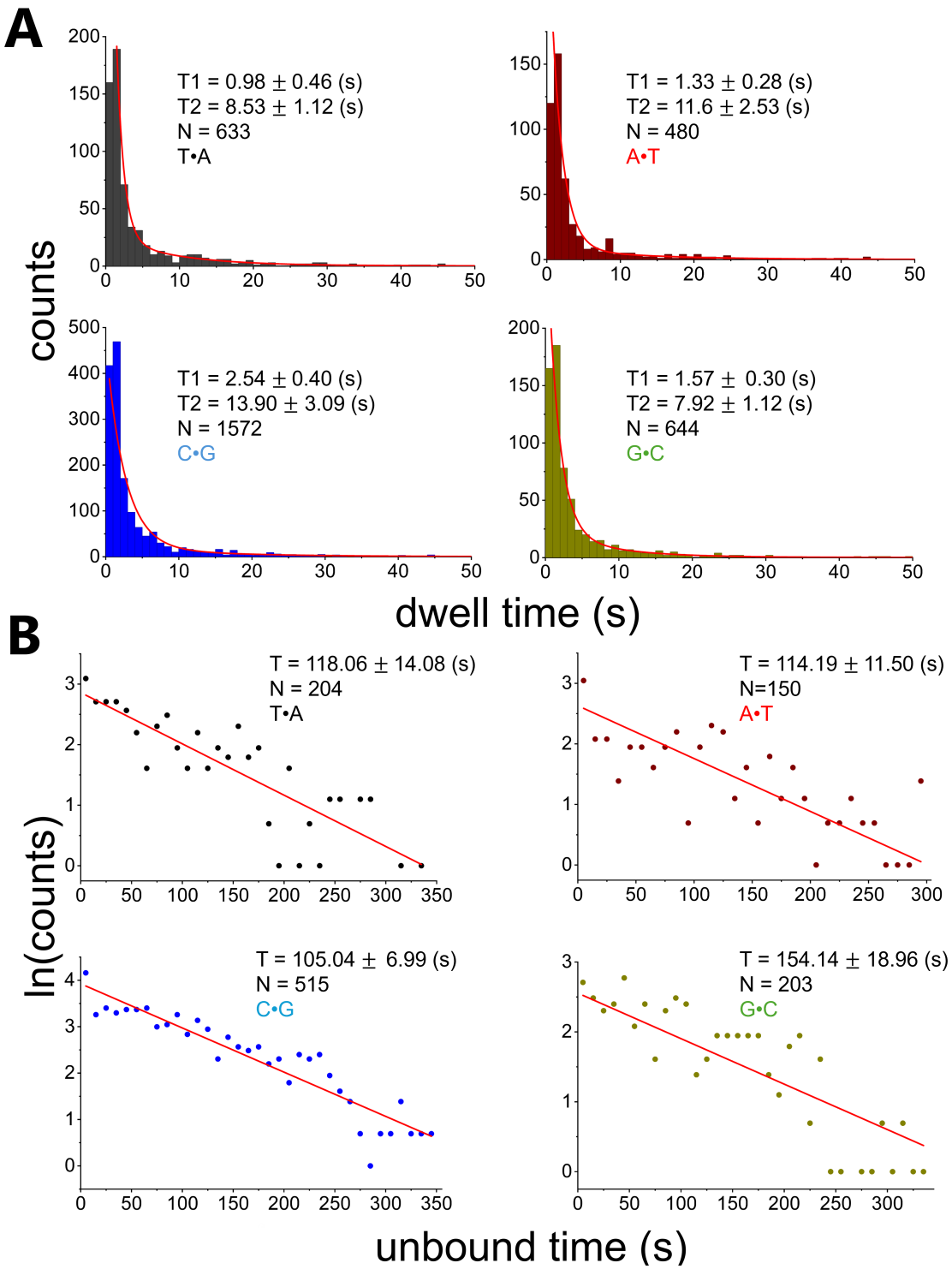

Figure 5.4: dwell and unbound time distributions of CAP on DNA variants. A is the histogram and fitting results of dwell time of CAP binding on four DNA variants. B displays the distribution of unbound time plotted against the natural logarithm of counts. The linear fit on a logarithmic scale shows single-exponential decay when analysing the original counts. The specific values of dwell time from exponential fitting and their corresponding sequence are shown within each panel and N is the number of bound or unbound states included in the statistics. To accurately capture the duration, one unbound time can be obtained when two binding events are detected. Therefore, the number of unbound events is significantly less than that of bound events.

sequences	A•T	T•A	C•G	G•C
on rate ($ms^{-1}*mM^{-1}$)	5.00±0.50	4.84±0.58	5.44±0.36	3.71±0.46
off rate (ms^{-1})	86.2±18.8	117.2±15.4	71.9±16.0	126.3±17.9
K_d (nM)	17.24±4.14	24.21±4.13	13.22±3.07	34.04±6.41

Table 5.2: kinetic parameters of CAP-DNA variant interaction. This table shows the on, off, and dissociation rate of CAP on DNA variants. The rank order of binding preference is C•G > A•T > T•A > G•C. CAP shows strongest specificity to consensus sequences (C•G).

dissociation rate, while the G•C pairing exhibits the lowest binding affinity overall.

5.1.3 Verification of gap formation

To verify in-situ the formation of gapped DNA, we deposited gapped DNA (GapG used in single base calling), performed transient hybridisation with complementary strands, and recorded the localisations for 10 frames. We removed the strands flanking the gap by washing the surface three times with MQ water, incubating it with 20 mM NaOH for 20seconds, and washing it three times using AB. Next, we incubated the surface using 100nM flanking strands Gap(1-27)-$\text{Top}^{\text{Cy3B,18}}$ and Gap(36-65)-Top for 15 minutes. Then we imaged the surface again for another 10-frame movie. Next, we performed a DBSCAN (eps = 100 nm, min_samples = 3) on localisations (Box side length: 5, Min Net gradient: 400, Baseline: 400, Sensitivity: 2.5, Quantum efficiency: 0.82, Pixel size: 117 nm) identified from the two movies. If the centroids of clusters in the two movies were closer than two pixels, the two clusters were considered as co-localised, which means successful attachments of the flanking strands onto the same biotin strands.

We find approximately 70% co-localisation of the initially deposited GAP with the in-situ formed GAP (5.5). We also observed more localisations in the in-situ formed GAP, which may be due to the sticking of Cy3B-modified flanking strands on the surface, or some of the initially deposited GAP may not have had both flanking strands. Except for the false-positive initial GAP, we deduced that during

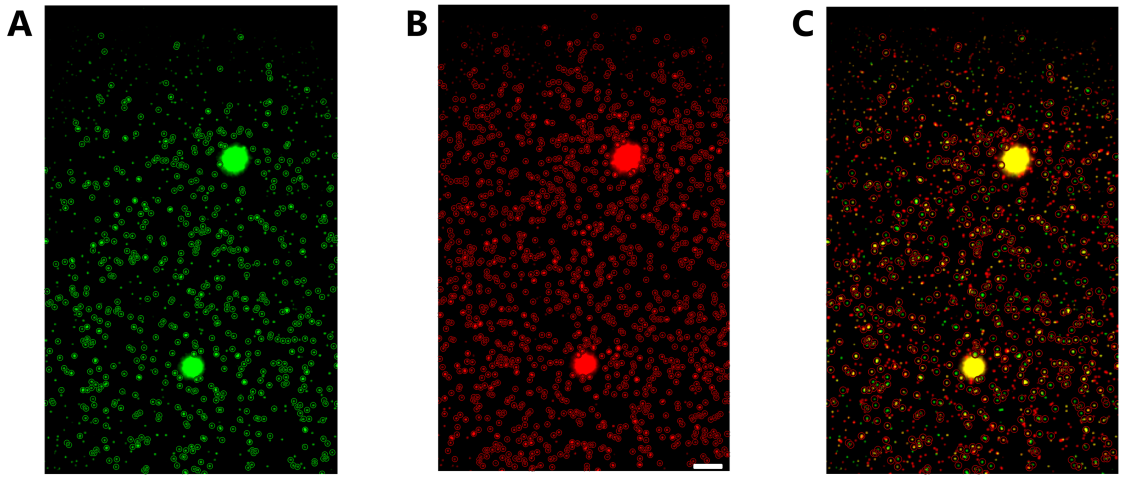

Figure 5.5: verification of in-situ gap formation. A: initial deposited Gap G B: In-situ formed Gap G C: Overlap of A and B to visualise the co-localisation rate. The scale bar is 5 μm.

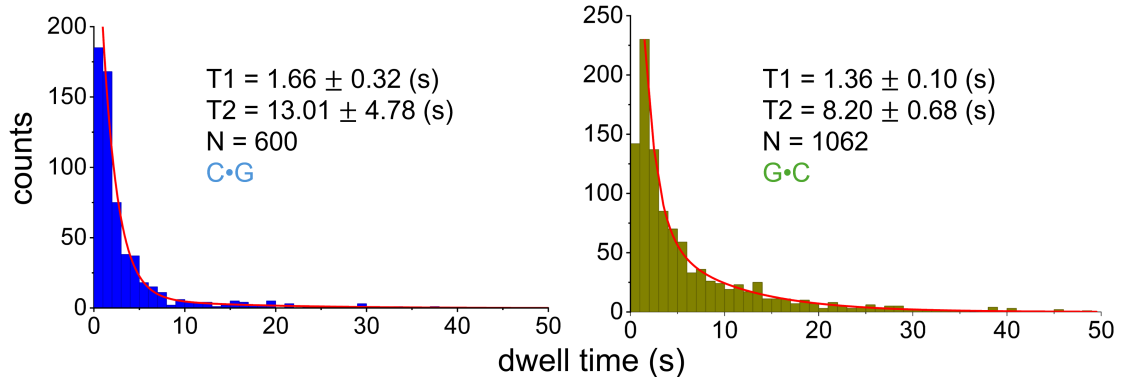

Figure 5.6: dwell time distribution of CAP on known base pairs. Histogram of CAP dwell time and the fitting results of the double exponential function from known base pairs: C•G and G•C. The highly similar results to those identified by base calling demonstrate the reliability of our method. N is the number of detected bound or unbound stages.

in-situ GAP formation, where the top strand is not labelled, the gap formation should succeed in more than 90% of molecules due to high concentration (1-2 μM) of flanking strands in incubation and similar Gap concentration.

5.1.4 Verification of sequence-dependent CAP dwell time

We also conducted the same analysis method described above on data with two known sequences: one with consensus base pair C•G, the other having pair G•C at position 5 and acquired highly similar results (5.6).

5.2 Bacterial transcription initiation

The multi-subunit RNA polymerase (RNAP) plays a central role in converting stored information in DNA to an active RNA transcript. Bacteria are an excellent system to study transcription due to their simplicity. In bacteria, RNAP binds a σ factor, generating the holoenzyme, which locates and unwinds promoter DNA, forming the transcription bubble of the open promoter complex (RP_o). The wound template strand is then able to enter the active centre cleft, and synthesis of the DNA template-directed RNA chain begins.

Maizels et al. observed in 1973 that RNA synthesis by RNAP is discontinuous [119]. RNAP typically adds nucleotides to the growing RNA transcript rapidly, with times ranging from 10 to 50 ms at most DNA positions. However, at certain positions, the process takes significantly longer—ranging from seconds to minutes. These slower steps are referred to as pauses. Pausing during transcription plays a crucial role in regulating transcription elongation, enabling RNAP to detect and respond to signals from both intracellular and extracellular environments. The occurrence of pauses is influenced by specific DNA and RNA sequences, as well as structural features, which happen at specific locations and times to facilitate interactions with small molecules and transcription factors essential for regulatory processes. Furthermore, transcription regulators can either directly target RNAP or influence the accessibility and affinity of promoters for RNAP [120]. More specifically, the initial pause has been reported to coordinate transcription and translation [121], providing opportunities for regulatory factors to bind the elongation complex [122], contributing to transcription termination [123], and aiding the proper folding of nascent RNA transcripts [124]. The pausing mechanism of bacterial RNAP has been under intensive study. The prevailing opinion at the moment is that a common unproductive conformation of the RNAP active site is the primary cause of transcriptional pauses.

During the initial RNA synthesis, the strong interaction with DNA holds RNAP at the promoter, which causes a scrunching state of the downstream DNA [125]. Next, the RNAP can pause for seconds, and the transcription proceeds with a productive, abortive, or futile-cycling path (5.7A). In the productive pathway, the RNAP escapes the promoter when the nascent RNA becomes 9-11nt long, which relaxes the scrunched DNA. In the abortive pathway, the short nascent RNA dissociate and the initial transcribing complex (ITC) resets to RP_o [126]. It has been reported that the ratio between productive and abortive pathways is dependent on the promoter and initial transcribed sequence [127]. Furthermore, in the cyclic pathway, ITC is temporarily trapped in a catalytically inactive state where it interconverts between pre-translocating and backtracking [128].

Here we studied the sequence-dependent initial transcription pausing on lac promoter (lacCONS) with its 16 variants at positions +6 and +7 using SPIN-Seq (5.7C). We looked into the propensity of RNAP to enter each pathway on variants and the time of ITC stayed in the inactive state.

5.2.1 Experiment protocol of transcription initiation

Dr Jagadish Hazra created the following protocols for transcription initiation and conducted the related experiments. The software used for the HMM fit and manual correction was developed by Dr Piers Turner(https://github.com/piedrro/napari-molseeq). The preprocessing steps, including drift correction, movie registration, and fluorescence trace extraction were conducted in the same way described in Chapter 4 (4.3).

All the DNA sequences (Biomer Corp) and corresponding strand names are listed in Table A.2. The E. coli RNAP holoenzyme was purchased from New England Biolabs. The protocol for preparing NeutrAvidin-biotin-PEG/mPEG functionalised coverslips is described in Section 2.4.

experiment type		laser mode	frame number	exposure time (ms)	laser power (mW)	excitation wavelength (nm)
transcription initiation	transcription	alternating	1200	200	0.8	638
	base calling	continuous	1200	250	0.6	638

Table 5.3: imaging conditions of the transcription initiation assay. This table shows the imaging conditions of the transcription initiation assay, including laser mode, frame number, exposure time and laser power.

All single-molecule measurements were performed using the Nanoimager microscope (Oxford Nanoimaging), equipped with a 100X, 1.4 NA objective, and a sCMOS camera (Orca flash4 v3). The TIRF mode was applied at an illumination angle of 56°. The emission channel was divided into two using a dichroic 640 LP splitter. One is for wavelengths ranging from 498 to 551 nm and 550 to 620nm. The other is for wavelengths 685/40 nm. The imaging conditions are listed in Table 5.3.

Library preparation for initiation pause assay We designed a library of lacCONS promoters containing sequences from -39 to +25 with variants of all possible combinations at position +6 and +7. Here we annealed 2 μM of biotinylated non-template strand from -39 to -10 positions having Cy3B at -15 position (lacCONS(-39/-10)-Top$^{\text{Cy3B,-15; Bio,-39}}$) with 1 μM of a fragment of template strand (lacCONS(-39/+10)-N^{+6}N^{+7}-B) from -30 to +10 position having degenerated base at +6 and +7 positions. Primer extension was performed using Q5 DNA polymerase (Thermofisher Scientific), keeping melting temperature, annealing temperature, and extension temperature at 95°C, 62°C and 72°C, respectively. Next, we annealed the second fragment of the template strand (lacCONS(+11/+49)-B$^{\text{Atto647N,+20}}$) with Atto647N labelled at position +20 to the second fragment of the non-template (lacCONS(+11/+49)-Top). This annealed strand was ligated to the primer-extended product using 4000 units of T4 DNA ligase (NEB), incubated overnight at 16°C. Next, we conducted an electrophoresis on a 20% bis-acrylamide-acrylamide native gel, and the band corresponding to 64-bp sequences with Cy3B and Atto647N fluorophore was excised from the gel and soaked in TE buffer overnight. We precipitated the gel-extracted products by mixing the solution with ice-cold ethanol three times the volume, 1 μL of glycogen, and 20 mM sodium acetate. The

precipitated DNA was kept at -20 °C for 4 hours and dissolved in assay buffer (AB: 20 mM HEPES (pH 7.7), 80 mM $MgCl_2$, 200 mM NaCl).

Transcription initiation-pause assay To form the open complex, we incubated 100 nM of DNA library with 150 nM of RNAP holoenzyme in KG7 buffer (40 mM HEPES, 100 mM potassium glutamate, 10 mM $MgCl_2$, 1 mM DTT, 100 μg/mL BSA, 5% glycerol) for 30 min at 37 °C. Next, 1 μL of 1 mg/mL heparin was added to the solution, which was then incubated for 1 min and centrifuged at 2000 rpm for 2min. We added the 0.5 μL collected supernatant and 30 μL KG7 onto a neutravidin-coated slide. The unbound open complex was washed off from the surface using AB, followed by 10-minute incubation with 500 μM ApA dinucleotide (Jena Bioscience), which was diluted in KG7 buffer with the oxygen scavenging system (1% Glucose, 40 μg/ml catalase and 0.1 mg/ml glucose oxidase). We added 200 μM NTPs using a home-built magnetically controlled tubing positioner approximately 15 seconds after the start of imaging.

In-situ DNA gap preparation and sequencing after transcription assay After the transcription assay, we determined the sequence at the +6 and +7 positions using a competitive inhibition assay. The surface was washed with 40 μL MQ water before adding 40 μL of 20 mM NaOH for 30s to dehybridize the dsDNA strands, followed by three washes using AB. Next, we created a gap around the sequence of interest (position +6 and +7) by incubating two 2 μM flanking ssDNA (lacCONS(+11/+49)-B and lacCONS(-39/+2)-B) for 15min in IB3 (20 mM HEPES, 200 mM NaCl, 80 mM $MgCl_2$ of pH 7.7, 1 mM TROLOX, 1% Glucose, 40 μg/ml catalase and 0.1 mg/ml glucose oxidase, 10% dextran sulphate , 10% formamide). The sequencing was performed using 200 nM 8-nt R-seal (R8-lacCONS(+3/+9)-$N^{+7}N^{+6}$-B), which was labelled with Atto647N at the 5'end and contained two degenerated bases (equal probability of having any of the four standard bases) at positions corresponding to positions +6 and +7 on the promoter and 600nM unlabelled U-seals in imaging buffer 4 (IB4: 20 mM HEPES (pH 7.7), 150 mM

$MgCl_2$, 200 mM NaCl, 10% dextran sulphate, 5% formamide and 200 μM of freshly prepared spermidine hydrochloride).

5.2.2 Sequence dependence of transcription pathway

Here, we applied the SPIN-Seq method to investigate how the DNA sequence at positions +6 and +7 of the lac consensus promoter affects initiation pausing, using a library containing 16 variants at these two positions (5.7C). To monitor RNA synthesis and RNAP pausing, we employed a single-molecule Förster resonance energy transfer (smFRET) strategy under alternating laser excitation. The FRET donor and acceptor were added at position -15 on the non-template strand and +20 on the template strand, respectively.

We immobilised promoter library with RNAP on PEGylated surfaces to form RNAP-DNA complex and imaged them at the single molecule level. An FRET efficiency (E*) of around 0.2 was observed before transcription started. After adding 20 μM NTPs, FRET activities changed in most molecules, indicating transcription occurrence. After recoding the transcription for 4 min, we created an 8-nt gap centred around positions +6 and +7 by removing the enzyme and template DNA strand and annealing two flanking strands. Then, we performed a competitive base calling method using 8nt R-seal (R8-lacCONS(+3/+9)-$N^{+7}N^{+6}$-B) with degenerated bases (equal probability of having any of the four standard bases) at +6 and +7 positions and the eight corresponding U-seals.

After acquiring the sequence of library molecules, we can investigate the relations between FRET patterns and different sequences. Drift correction, movie registration, fluorescence time profile extraction and base calling were conducted as described before (4.3). After acquiring the fluorescence traces of donor emission upon donor excitation (I_{DD}), and acceptor emission upon donor excitation (I_{DA}), we calculated

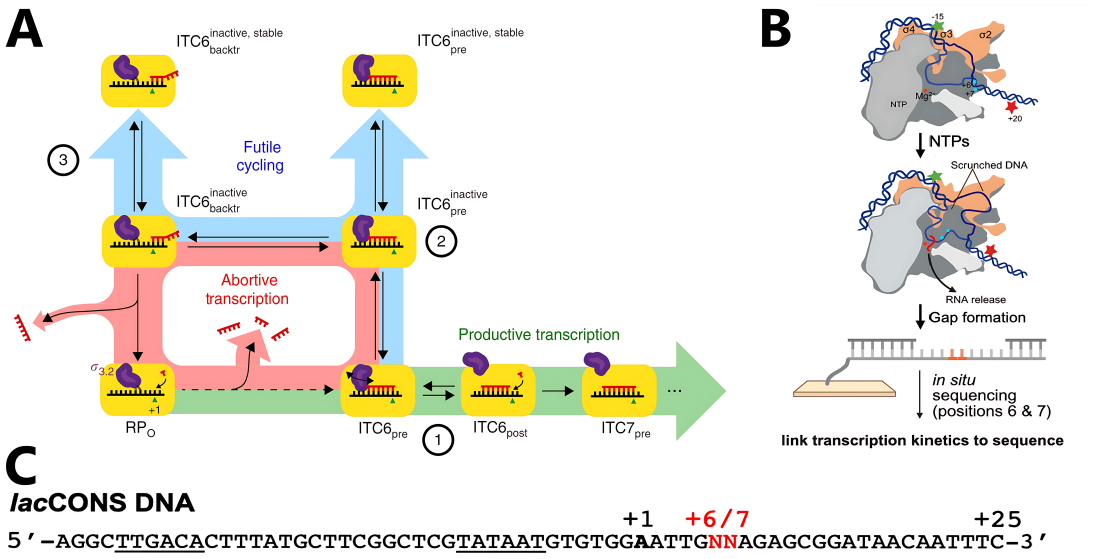

Figure 5.7: linking bacterial transcription initiation branching with DNA sequences. A: The progress of initial transcription has three competing pathways. Productive pathway leads to promoter escape and produces full-length RNA (indicated in green). Red abortive transcription leads to the synthesis and dissociation of short nascent RNAs. In futile cycling (marked in blue), ITC is temporarily trapped in a catalytically inactive state, interconverting between pre-translocated and backtracked states. B: Schematic plot of studying transcription complexes with a 16-variant library using SPIN-Seq. After the addition of NTPs to open complexes, transcription leads to FRET signal changes due to the movement of the DNA downstream of the transcription start site. After recording transcription-associated FRET signals, the protein and template strand are removed, and the sequence at positions +6 and +7 is read using the competitive inhibition assay. C: lac consensus DNA with the -35 and -10 promoter elements indicated using underline, and the variants are at positions +6 and +7. (A adapted from [128])

the FRET efficiency (E*) as

$$E^*_{FRET} = \frac{I_{DA}}{I_{DA} + I_{DD}} \tag{5.7}$$

The FRET trace segmentation of E^* and the classification were performed with hidden Markov models (HMM) with manual correction. A few examples of the FRET signals and their sequencing results from library molecules are displayed in Figure 5.8.

More than 90% of molecules exhibited high E* related to initial transcription pausing. They can be classified into four types, each corresponding to a different initial transcription pathway. In class I, E* increases to about 0.8 through one or

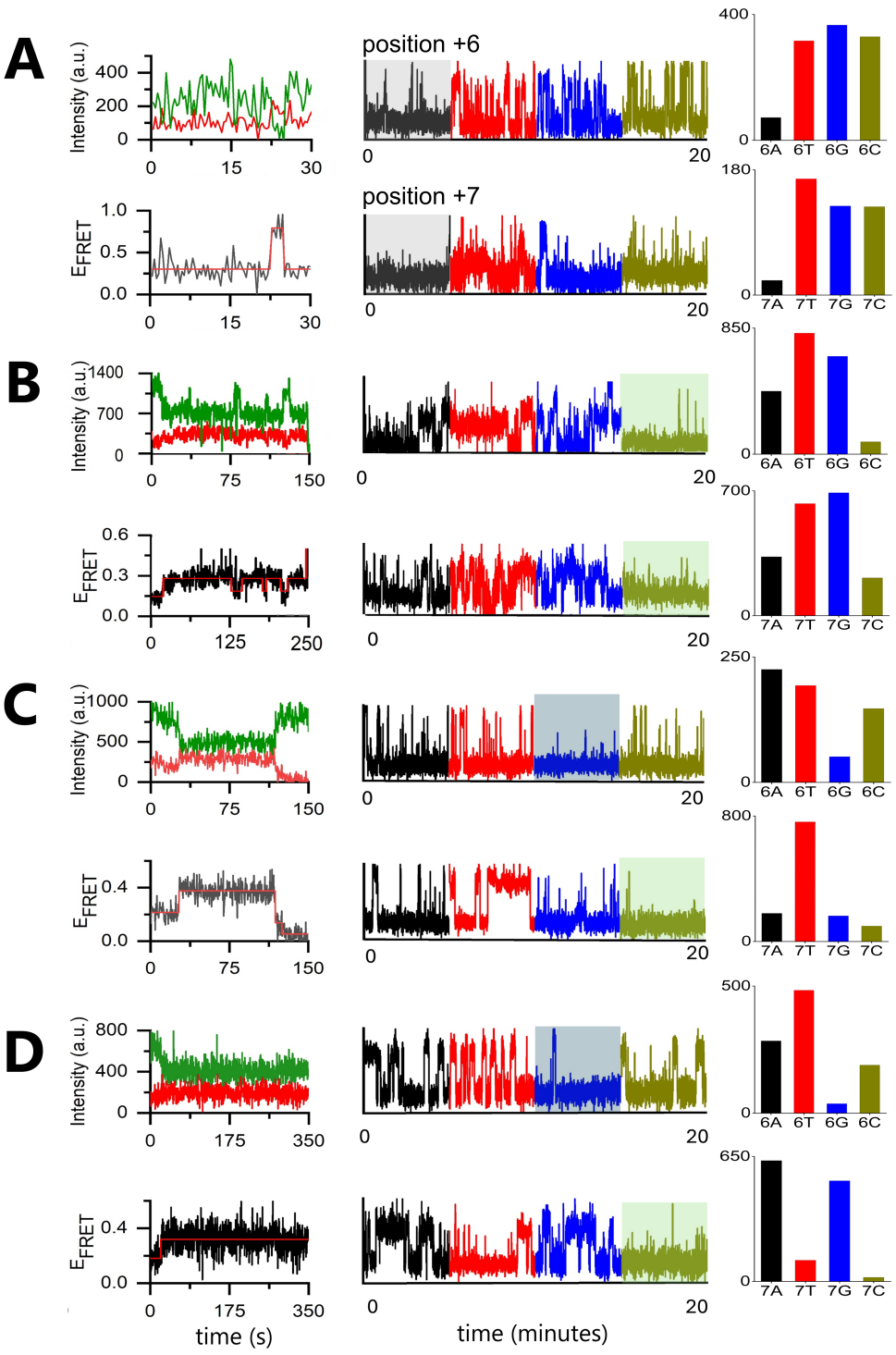

Figure 5.8: examples of FRET traces and their corresponding base calling results. A-D show four examples of FRET traces along with their corresponding base calling results. In each panel, the donor and acceptor signals are displayed in the top left corner, indicated by green and red colours, respectively. The corresponding FRET efficiency is shown in the bottom left corner. The fluorescence time profile from position 6 is presented at the top middle, while the LC values are located in the top right corner. Similarly, the data for base 7 is displayed in the bottom middle and bottom right corners.

multiple steps and decreases back to the low level at 0.2, which is consistent with a productive path where promoter escape and transcription complete on the entire template. Class II molecules switch between states with E* of 0.2 and 0.4. These correspond to the futile cycles, in which the short nascent RNA is released and RNAP restart the synthesis again. Class III molecules have a single pause at about 0.4, which is likely to be the inactive ITC with backtracked RNA. Class IV reaches an E* level of 0.4 and maintains this level for the rest of the trace. These molecules are the stably scrunched DNAs that are reported to be the major intermediates in lac consensus initial transcription [128, 129] (5.9A, B).

Furthermore, we looked into the probability of each variant entering each class. $A^{+6}A^{+7}$ and $A^{+6}T^{+7}$ sequences are more likely enters into cycling path, while $C^{+6}G^{+7}$ is the least to enter this path. This finding supports the previous proposal that cycling RNAP leaves the paused state via RNA backtracking, followed by RNA entry in the NTP-entry channel, RNA releases, and restart of RNA synthesis [66, 129]. Since the A•T base implies low duplex stability, easier opening of RNA-DNA hybridisation, and matches the determinant for RNA backtracking [130, 131]. Sequences $C^{+6}G^{+7}$ and $T^{+6}G^{+7}$ are over-presented in class IV, which is likely to reflect RNA backtracking into NTP entry channel and transcription arrest (5.9C).

We explored the pause time of states where E* is approximately 0.4 for each sequence form molecules in class I, II, and III. The dwell time is acquired by fitting the histogram with a single exponential decay function (5.10A). $C^{+6}G^{+7}$ has the longest pause of approximately 27.5s, while $G^{+6}A^{+7}$ has the shortest pause of about 8.4s (5.10B). Consistent with previous work [128, 129], sequence $T^{+6}G^{+7}$ which is the native lac promoter has a relative long pause of roughly 26.4s. Furthermore, we classify the single pause FRET traces into long pausing (longer than 10s) and short pausing (1-10s). The preference of different variants towards long and short pause is significant. $T^{+6}G^{+7}$ and $C^{+6}G^{+7}$ prefers long pausing and $A^{+6}A^{+7}$ prefers short

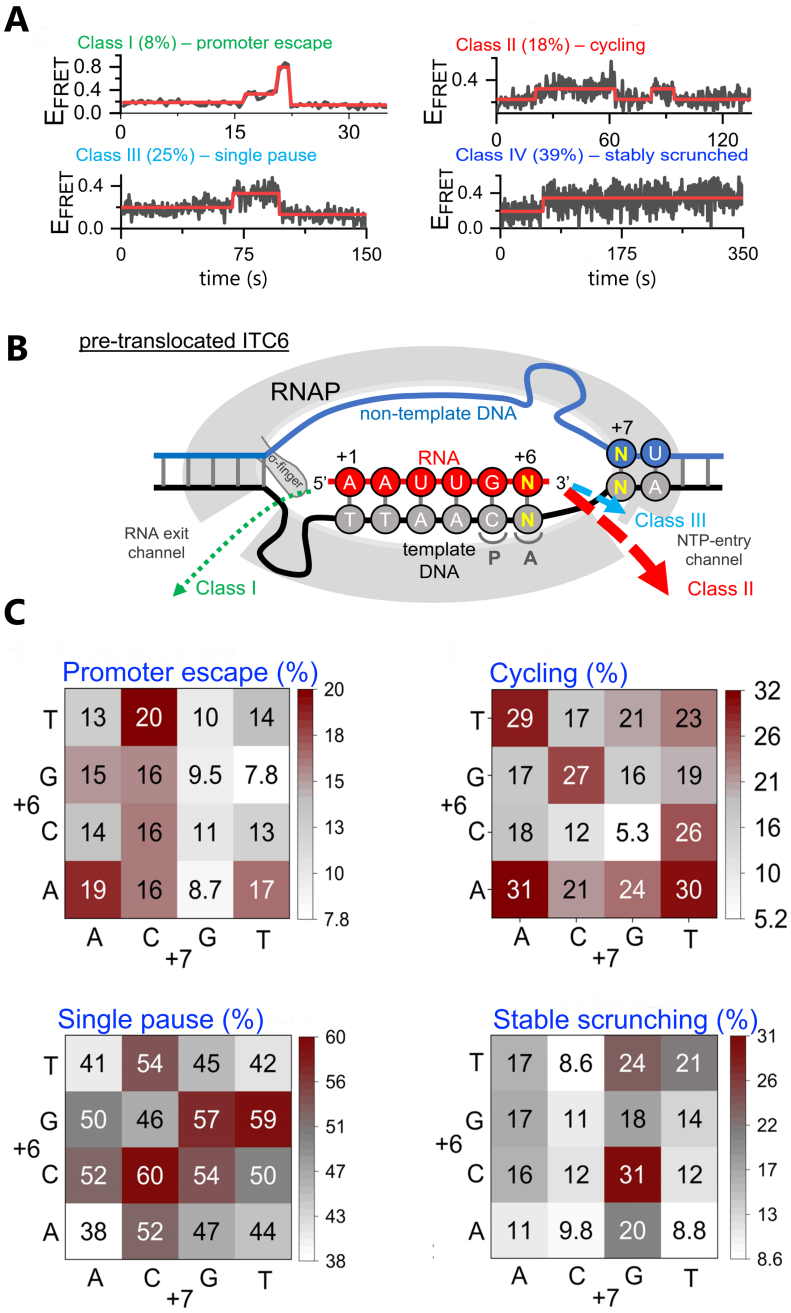

Figure 5.9: four FRET classes and their corresponding transcription initiation branches. A: Examples of four types of FRET behaviours observed that represent different initial transcription activities. B: Schematic plot of the pre-translocated ITC6 state with nucleic acids. 6mer RNA overcomes the σ singer and continues to elongate (class I), or backtracks into the NTP-entry channel and dissociates (class II), backtracks and stays in the NTP-entry channel (class III). Moreover, class IV involves inactive ITC6, which may reflect an inactive RNAP conformation. At the beginning of the nucleotide addition cycle, the 3' end of the RNA transcript occupies the product site (P site). To extend the RNA, a NTP enters the active site and base pairs with the template DNA in the acceptor site (A-site). The 3'-OH of RNA attacks the correctly paired NTP substrate to extend the transcript by one base. C: shows the percentage of all sequence combinations belonging to class I, II, III, and IV.

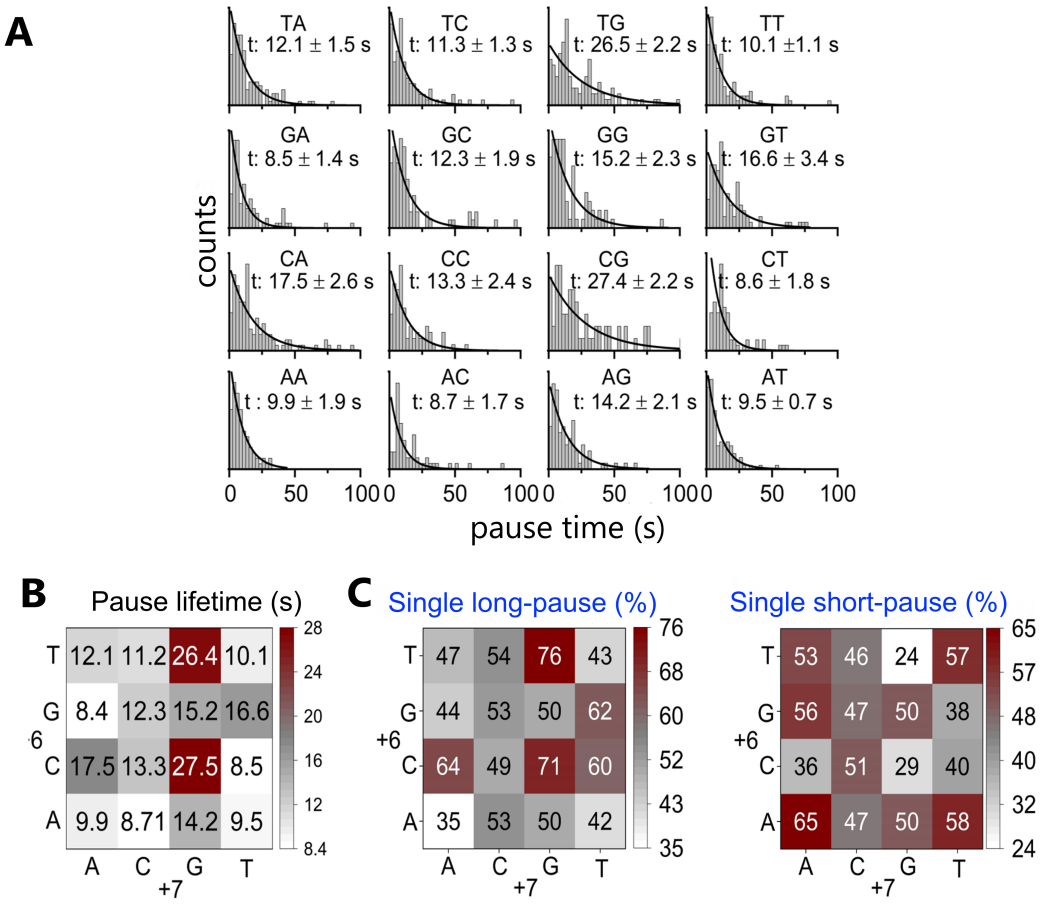

Figure 5.10: sequence-dependence of pause time. A: the histograms of pause time and the result of a single exponential fit for all 16 variants. B: Heatmap of the average pause time for all variants. C: The ratio between long (more than 10s) and short (less than 10s) pause from traces having a single pause for all variants.

pausing (5.10C).

To understand the overall relationship between the sequence variants of LacCONS promoters and the initial transcription pathway. We overlapped all FRET traces that belong to each variant as heatmaps (5.11A). Moreover, we can see the patterns of each class in their average traces (5.11B). For example, $C^{+6}G^{+7}$ has a high probability of staying in scrunching, and its average trace does not have a clear decay with the increase of time. To demonstrate the difference, the four average traces that reach the average E^* of about 0.27 at 15s when adding NTPs, are displayed in Figure 5.11B. The average $T^{+6}G^{+7}$ trace exhibits a slight decrease, indicating a high probability of entering the stably scrunched state. In contrast,

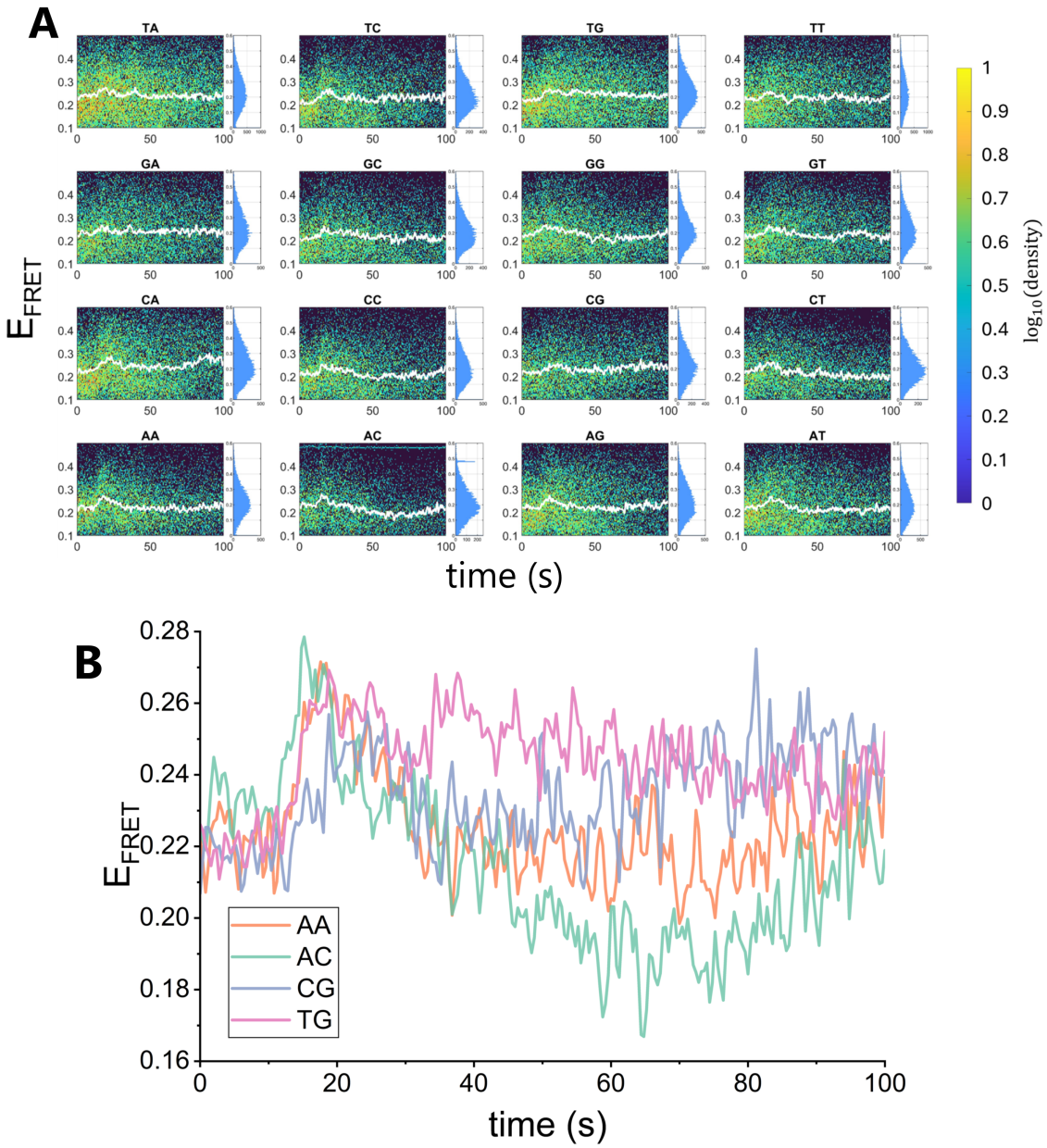

Figure 5.11: overlaid and average FRET efficiencies from variants of LacCONS promoters. A: This plot shows the overlap of all FRET traces belonging to each variant. The white line indicates the average FRET efficiencies. B: Overlap of the average FRET traces from four sequence variants ($A^{+6}A^{+7}$, $T^{+6}G^{+7}$, $A^{+6}C^{+7}$, $C^{+6}G^{+7}$).

$A^{+6}C^{+7}$ has a significant decay corresponding to a high probability of a single short pause.

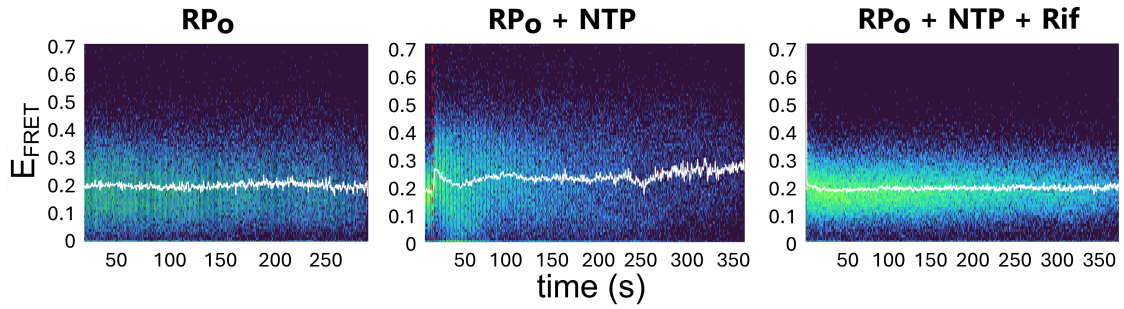

Figure 5.12: FRET signals from experiments at different conditions. This plot illustrates the overlap of FRET efficiencies under three conditions: without NTP, with NTP, and with both NTPs and rifampicin added, from left to right.

5.2.3 Verification of transcription activities

Additionally, a control experiment was conducted to block RNA extension beyond 3nt by adding rifampicin. Our investigation included three distinct experimental conditions: (1) the absence of NTPs, (2) the introduction of NTPs, and (3) the presence of both rifampicin and NTPs. The results showed no variation in FRET efficiency under conditions 1 and 3. In contrast, a significant change in FRET efficiency was observed approximately 15 seconds after recording when NTPs were added in condition 2. This finding supports the conclusion that the observed FRET activities were indeed a result of transcriptional processes.

5.2.4 Verification of sequence-dependence of transcription initiation

We selected the variants of lacCONS-$C^{+6}G^{+7}$ and lacCONS-$A^{+6}T^{+7}$, which were associated with long and short pause duration, respectively, from our sequencing method and performed the same experiment and analysis upon them. The results of both dwell time and propensity to enter different classes from the two known sequences are very similar to those from base calling, which proves the validity of our method (5.13).

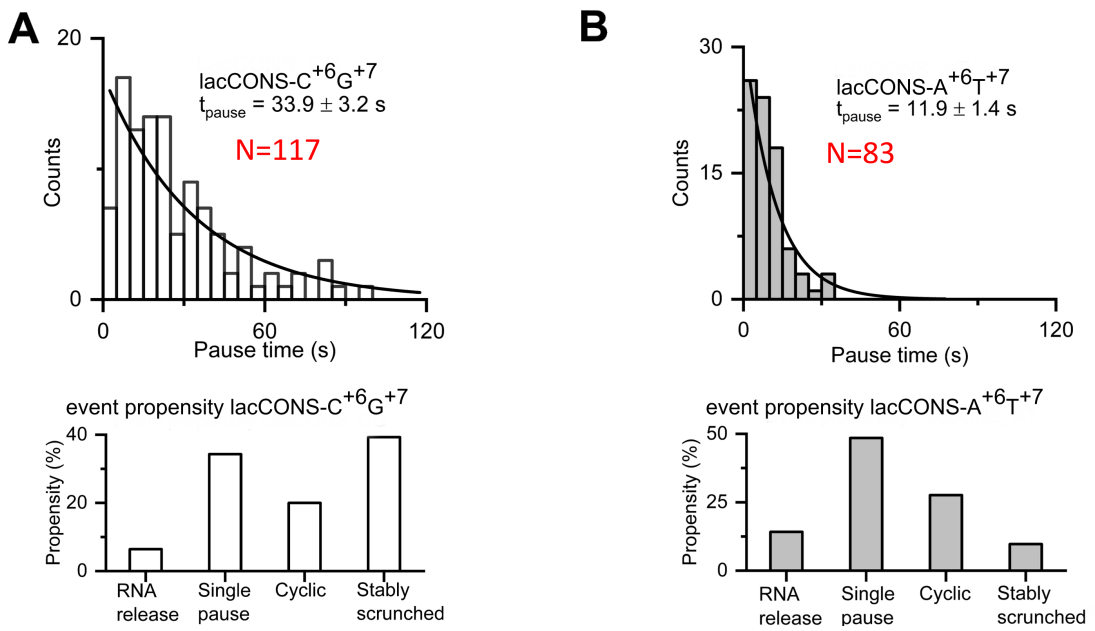

Figure 5.13: transcription initiation on known sequences. This plot shows the dwell time (top) and propensity (bottom) of sequences $C^{+6}G^{+7}$ (A) and $A^{+6}T^{+7}$ (B) to enter each initial class. N is the number of molecules in the dataset.

5.3 Summary of SPIN-Seq technique

We developed the SPIN-Seq technique, which can connect biomolecule-DNA interactions to DNA sequences at the single-molecule level. After recording the activities of molecules of interest, a gapped DNA was constructed, and the positions in the centre of the gap can be sequenced by exploiting the thermodynamic differences between complementary DNA sequences and those with mismatches on the gapped substrates (5.1). This approach can be applied to study the interactions of a wide range of biomolecules with nucleic acids in vitro, as long as they can be labelled with fluorophores. SPIN-seq only requires a standard wide-field fluorescence microscope, making it affordable and accessible to researchers.

We applied this method to study the sequence-dependent CAP-DNA interaction, as well as initial transcription pauses. When examining the CAP-DNA interaction, we investigated the kinetics of binding and confirmed that CAP forms more stable bonds with the consensus base pair compared to mutants. As for the initial

transcription, we used 16 variants and noted distinct propensities for entering different classes, as well as a more than threefold difference in pausing time resulting from only two base mutations. For the single-molecule sequencing aspect, as a proof of concept, we tested our approach to sequence one, three, and five bases, achieving an average accuracy of over 97%.

The data analysis pipeline for the SPIN-Seq technique is summarised in the flowchart 5.14. In response to the requirement of colleagues, drift correction and channel registration were not only applied to the localisations but also to the raw movies, which significantly increased the computational cost and were therefore conducted on CUDA.

Due to high (μM range) imager concentrations, despite the usage of fluorogenic probes, we still faced background problems, which reduced the SNR and imposed challenges to the time series analysis. After initial background removal by subtracting the background of the local environment, we employed a Bayesian approach to effectively smooth out the noise while preserving the abrupt changes that we sought to detect.

To cope with the loss of fiducial markers, we developed an image registration method based on constructed super-resolution images. These images are essentially spots randomly scattered on the image (lack of continuous structures); thus, the conventional intensity-based registration method could fail sometimes. Phase cross-correlation can cope with this type of image more robustly, as it focuses on the locations of the spots rather than their brightness.

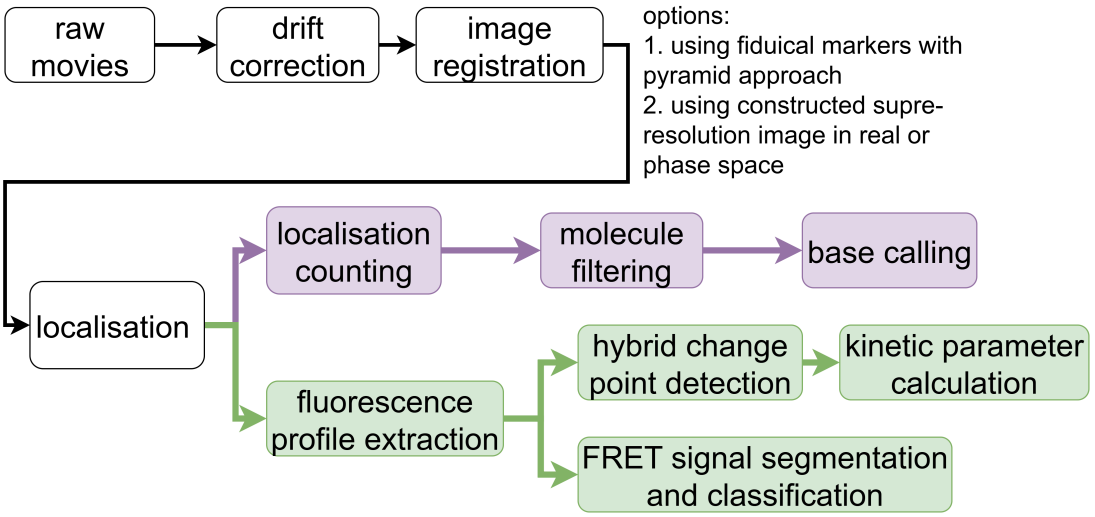

Figure 5.14: data analysis pipeline for SPIN-Seq. This plot summarises the data analysis methods used in the SPIN-Seq technique. The green boxes represent the methods for extracting fluorescence signals and calculating the kinetic parameters for CAP binding and FRET efficiency for transcription initiation assays. The purple boxes illustrate the sequencing methods employed. Common steps are indicated in white.

6

Conclusion and future work

In this project, we mined the spatial and temporal information from DNA-DAN or protein-DNA interactions. Here, I would like to discuss the general methods used in both single-virion PAINT and SPIN-Seq and possible future work before proceeding to separate discussions.

6.1 Connection between localisations and fluorescence traces

We processed two types of data: localisation data and fluorescence time profiles. Current popular analysis methods tend to treat them separately by considering the localisations as point clouds and fluorescence traces as sequential data. Evidently, they do possess such inherent features. By exploring the SMLM data as point clouds, one can apply 2D/3D image analysis methods after image/volume rendering, or work directly with localisations. Both approaches yield the geometrical properties of the molecules of interest, including the precise distances between them and the density of labelled molecules, among others. These properties can provide valuable insights: clusters are often linked to biological structures, and the co-localisation of different molecules may indicate a functional relationship between them. Sequential datasets (also referred to as ordered data) can be analysed by various methods

based on Bayesian statistics, such as the online change-point detection method we used. Moreover, this order usually helps us reduce the search space significantly for the optimisation algorithm.

However, both data types describe the same events, just emphasising different aspects. While we focus on one type of information, the other remains accessible to some extent, albeit in a different form. For instance, we can determine the dwell time of a binding event from localisations. This can be achieved through a linking process that connects localisations within a specific distance and across consecutive frames, allowing some tolerance for dark time. By implementing this linking process, each localisation is assigned an additional property: its duration of existence. Conversely, when we concentrate on the time profile of a binding event, a pooling process replaces the diffraction pattern with statistical features, such as average or maximum values. This approach sacrifices the high-precision positional information that comes with the PSFs but reduces the dimensionality of the input data. It also provides a level of regularisation, making the data invariant to small fluctuations or distortions. The coordinates are still available at a pixel-limited precision, making this method suitable for traditional diffraction-limited microscopy.

The localisation table contains much richer information than just coordinates; it includes parameters such as frame (timestamp), photon count (similar to intensity values), widths of point PSFs, etc. These parameters are both valuable and accessible in the localisations. For instance, we filtered out molecules with LC value sets that were affected by sticky seals using a binary-coded time series. Similarly, we used time series analysis to identify the localisation clusters that correspond to genuine virus particles.

Previous studies already show the improvement by connecting temporal and spatial information together, such as achieving sub-pixel resolution in conventional wide-field microscopes [132], enhancing localisation accuracy in SMLM [133], etc.

With advancements in computer science and increased computational resources, we can preserve information within raw data better for analysis. Deep neural networks (DNNs) offer such capabilities to analyse videos directly. For SMLM data, we could train the network on a small area that encompasses the diffraction pattern across all frames. In this way, all related information is retained while keeping the size of the training data easy to adjust. Video content analysis has been applied in surveillance systems, social and medical science, using convolutional neural networks, video transformers, spatio-temporal graph neural networks, and hybrid models [134–137]. However, this will be a challenging approach. Since this type of analysis has not been applied to microscopy video yet, modifications of the current network architecture are probably required.

6.2 Pre-processing methods

For standard preprocessing steps like drift correction and image registration, there are several reliable and effective methods available. Except for the AIM drift correction technique used in our analysis, Cnossen et al. [11] proposed a drift estimation method that minimises a bound on the entropy of localisations. Compared to the AIM algorithm, this method provides a smoother drift trajectory, which presents a more realistic situation, as it is unlikely for drift to exhibit vibrations, as long as the microscope is kept away from such a source. In both AIM and another popular drift correction method, RCC, these small vibrations in the estimation caused by noise usually do not harm our following analysis. These three methods yield similar results on a large scale, and the entropy method requires significantly more computation. It would be beneficial to enhance the AIM method by incorporating some regulation to suppress such vibration in drift estimation.

As for registration, we have currently achieved approximately 10 nm registration precision using 200 nm fiducial markers in SPIN-Seq. This precision can be improved by utilising smaller beads. Additionally, we also developed a registration method that corrects solely for translation in Fourier space. We also attempted to address

the slight rotation between the two channels by remapping the images into polar space and applying phase cross-correlation. However, we failed to detect any rotation. This issue could potentially be resolved by upsampling and optimising the image construction method.

So far, the precision of localisation and drift correction has not posed any issues for accurately assigning localisations to molecules. However, noise from slow-moving imagers between adjacent molecules can blur the boundaries of the molecules. Some of this noise can be removed by analysing the shape of the PSFs.

6.3 Imager performance

The performance of imagers and their docking strand plays an important part in the data quality. The sequences can be engineered to achieve the desirable dwell time, on/off rates, and specificity. We utilised fluorogenic imagers to alleviate the background issues, which also allows us to boost the imaging speed with higher imager concentration. The size of our fluorogenic probe ranged from 6 to 13 nt. The average length of a base in ssDNA is approximately 0.63 nm, with a reported persistence length of a few nanometres [138, 139]. The short length of the imagers (smaller than the persistence length) can impede effective contact quenching and reduce fluorescence intensity in the binding state due to FRET. To achieve higher FF, it is advisable to consider longer sequences to mitigate stiffness and potentially take advantage of secondary structures.

Additionally, the elastic properties of ssDNA are closely linked to ion concentration and the composition of the DNA sequence, providing more control for optimisation. Various theoretical models, such as the freely jointed chain and the worm-like chain, are based on the premise that a polymer is stretched within its entropic regime. For dsDNA, the elastic characteristics resemble those of a non-interacting (non-self-avoiding) polymer until the molecule exceeds 50 μm in length [140]. Thus, in practice, dsDNA behaves like an ideal polymer. However,

ssDNA is much more flexible, and self-avoiding interactions play a more significant role; thus, predicting its behaviour relies on the development of a more suitable polymer physics model.

Solubility is an important factor in determining the tendency for non-specific binding. As we tested in the fluorogenic DNA-PAINT experiment, the combination of Atto643-BMNQ1 has higher FF and fewer false-positive vRNPs compared to Atto647N-BHQ1. By updating the fluorophore and quencher we used in SPIN-Seq to the Atto643-BMNQ1, we should be able to reduce the background and have fewer sticky seals.

6.4 Throughput increase

To enhance the throughput of an assay, several strategies can be employed. Utilising spectrally separated fluorophores or increasing the concentration of imagers can reduce the imaging time required. Additionally, raising the concentration of particles on the surface, using a microscope with a larger FOV, or implementing a waveguide chip [141, 142] can further boost assay throughput. However, denser molecules on the surface and imagers in the buffer can both cause issues in assigning localisation to molecules in proximity and overlaps among PSFs. They also place greater demands on the precision required for localisation, drift correction, and image registration. Recent advancements in the development of software designed for high-density SMLM in both 2D and 3D settings [143, 144] provide an excellent foundation for conducting high-density experiments.

Our current method and setup in SPIN-Seq enable us to identify approximately 2,000 molecules per FOV. However, it is crucial to increase this output to explore all possible permutations of longer sequences. In addition to the previously mentioned methods, Nanoimager's multiple FOV function can alleviate this issue. Nonetheless, the total imaging time remains unchanged, and this method cannot be used for time-sensitive samples. For instance, transcription initiation, which we observed, occurs

only once after the addition of NTPs. As a result, sequentially recorded movies from different FOVs cannot capture this process, except for the first FOV. Additionally, conducting overnight experiments using a microfluidic device is also beneficial. While this approach does not improve experimental efficiency, it significantly reduces labour and allows for the collection of much larger datasets without significant changes to the current setup.

6.5 Future work in single-virion DNA-PAINT

The combined data from both super-resolution and stoichiometry assays support our hypothesis regarding the correlation between the strength of vRNP-vRNP interactions, their distances, and co-presence rates. We found highly flexible spatial arrangements and structures of vRNPs, as well as a selective, redundant, and cooperative assembly process for IAV's segmented genome.

It is the first time that the vRNP complexes were visualised and identified inside virus particles. Unlike previous studies that focused on identifying packaging signals or evaluating how mutations in RNA or nuclear proteins influence packaging efficiency, our approach directly examines the combinations of vRNPs, which allows us to evaluate the overall outcomes of all interactions.

A comprehensive understanding of the selectivity of packaging is essential for elucidating the mechanisms underlying reassortment. DNA-PAINT on mutant strains may quantify the effect of individual sequences motifs more precisely than determining the number of infectious cells or quantifying virus materials using bulk measurements such as immunoassays.

Both our experimental and analytic methods can be applied to study the overall effects among vRNPs for other viruses with segmented genomes as well. For a virus with fewer RNP molecules, such as influenza C, influenza D and the Cystoviridae family, due to the smaller number of possible combinations for the intermediates,

it is possible to paint a more detailed picture of the assembly pathway using our method.

Both our super-resolution and stoichiometry assays can be improved by the fluorogenic probes we designed at a later stage. These methods resulted in two publications [145, 146], focusing on IAV particles and the design of fluorogenic imagers, respectively.

It would be beneficial to develop a solution-based immobilisation approach to further preserve the samples and test strains that are mutant in previously identified packaging signals. Additionally, comparing the packaging intermediates at earlier stages of infection would provide valuable insights.

The development of fluorogenic probes can not only alleviate the background ans slow imaging issues, its high signal specificity will posses great advantages in cell-based experiments. Our current method utilises TIRF microscopy, which minimises the impact of fluorescence background but results in a limited illumination depth. If we were to adapt this method for cell-based experiments, we would risk losing a significant number of segment complexes. Therefore, it is essential to establish new experimental protocols that can capture more segments at varying heights within the cells. Techniques such as confocal scanning microscopy could be utilised, and subsequently, we would need to develop new analysis methods based on the features of the new data. Additionally, the cell unroofing technique has been reported to be effective with Cryo-EM, AFM, and fluorescence microscopy without the need for membrane permeabilisation [147–149]. This method may help us capture budding viruses without extensive modifications to our current PAINT and analysis methods.

6.6 Future work in SPIN-Seq

Connecting DNA sequences with molecular-DNA interactions is fundamental to deciphering the genetic code and gene regulation. DNA is a dynamic blueprint that is read and interpreted by various molecules. Transcription factors regulate gene expression by binding to specific sequences. Epigenetic regulation is mediated by processes such as DNA methylation, which recruits proteins involved in gene repression or inhibits the binding of transcription factors.

Single-molecule techniques provide in-depth information about the interplay among function, structure, and sequence. However, it is usually limited to investigating a small number of different samples. As a new parallel method that maps the function-sequence landscape at the single-molecule level, SPIN-Seq provides a way to explore large sequence spaces and acquire a comprehensive understanding of biological processes. Previous parallel multiplexed studies across sequence space require Illumina chips and are suboptimal for single-molecule imaging [103, 106]. SPIN-Seq utilises standard coverslips and a wide-field microscope, and is compatible with various nucleic acid systems, out-of-equilibrium reactions and fluorescence-based single-molecule approaches. Overall, SPIN-Seq has significant potential to aid a wide range of research, including fields such as pharmacogenomics and aptamer development.

It is the first truly single-molecule 'phenotype + genotype' platform. Illumina provides single-molecule information, but actually reads cluster signals from bridge amplification. The SPIN-Seq method has not been officially published yet (submitted to Nature Communications). We hope it will aid many other fluorescence-based research projects, making massively parallel analysis accessible.

An essential step of the further development of SPIN-Seq is to increase the throughput. As the number of unknown sequences increases, both the imaging time and the number of DNA strands involved will rise, while the effective concentration

of entirely complementary seals dramatically decreases. It adds complexity and increases the cost of the experiment.

A molecular dynamics simulation could be a valuable step for this project, as it can help guide the selection of experimental conditions and estimate the limits of the approach. For example, this simulation could address questions such as the probability of mismatched DNA strands exhibiting stronger binding due to fluctuations, and it could find other possible DNA configurations (such as configurations with no gaps or one stacking site) that might provide detectable differences between complementary and mismatched seals.

For the calculation of the predictive mean using the Bayesian method, thanks to the relatively small number of frames (approximately 1,000), we did not encounter the memory usage issue of this algorithm, which grows quadratically with the number of timestamps. The next step in this analysis could focus on resolving the memory usage issue, developing new methods for denoising, or exploring a completely different segmentation method.

The most significant missing piece in our pipeline is the automation of segmenting and classifying FRET signals. Despite years of research and application, these two tasks remain challenging due to the confounding factors such as the intensity changes, bleaching and blinking due to photophysics, low SNR, and the complex underlying biological dynamics. The conventional FRET trace idealisation methods are based on HMM. However, HMM requires substantial knowledge of the system, such as the number of states and assumptions about the transition probabilities. More recent attempts to analyse FRET traces use deep learning methods with frameworks such as long short-term memory, and convolution units [150–152]. In our case, one can build a DNN with similar architectures using simulated data and refine it with manually labelled data from this project.

To increase the accuracy of the localisation counting method, the next step could be adding a filtering step for localisation, using criteria such as SNR, PSF shape, and duration of each localisation after linking. It would also be beneficial to reconstruct the parameter confidence to accurately reflect the probability of correct calling, rather than the relative difference compared to other molecules in the dataset.

References

[1] E. Neher and B. Sakmann. "Single-channel currents recorded from membrane of denervated frog muscle fibres." eng. In: *Nature* 260 (5554 1976), pp. 799–802.

[2] Mickaël Lelek et al. "Single-molecule localization microscopy". In: *Nature Reviews Methods Primers* 1.1 (2021), p. 39. URL: https://doi.org/10.1038/s43586-021-00038-x.

[3] Philipp Blumhardt et al. "Photo-Induced Depletion of Binding Sites in DNA-PAINT Microscopy". In: *Molecules* 23.12 (2018). URL: https://www.mdpi.com/1420-3049/23/12/3165.

[4] Ralf Jungmann et al. "Single-Molecule Kinetics and Super-Resolution Microscopy by Fluorescence Imaging of Transient Binding on DNA Origami". In: *Nano Lett.* 10.11 (Nov. 2010), pp. 4756–4761. URL: https://doi.org/10.1021/nl103427w.

[5] Joerg Schnitzbauer et al. "Super-resolution microscopy with DNA-PAINT". In: *Nature Protocols* 12.6 (2017), pp. 1198–1228. URL: https://doi.org/10.1038/nprot.2017.024.

[6] Raimund J. Ober, Sripad Ram, and E. Sally Ward. "Localization Accuracy in Single-Molecule Microscopy". In: *Biophysical Journal* 86.2 (Feb. 2004), pp. 1185–1200. URL: https://doi.org/10.1016/S0006-3495(04)74193-4.

[7] Kim I. Mortensen et al. "Optimized localization analysis for single-molecule tracking and super-resolution microscopy". In: *Nature Methods* 7.5 (2010), pp. 377–381. URL: https://doi.org/10.1038/nmeth.1447.

[8] Alex Small and Shane Stahlheber. "Fluorophore localization algorithms for super-resolution microscopy". In: *Nature Methods* 11.3 (2014), pp. 267–279. URL: https://doi.org/10.1038/nmeth.2844.

[9] Yina Wang et al. "Localization events-based sample drift correction for localization microscopy with redundant cross-correlation algorithm." eng. In: *Optics express* 22 (13 2014), pp. 15982–91.

[10] Frank J. Fazekas et al. "A mean shift algorithm for drift correction in localization microscopy." eng. In: *Biophysical reports* 1 (1 2021).

[11] Jelmer Cnossen et al. "Drift correction in localization microscopy using entropy minimization". In: *Opt. Express* 29.18 (2021), pp. 27961–27974. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-29-18-27961.

[12] Hongqiang Ma et al. "Toward drift-free high-throughput nanoscopy through adaptive intersection maximization." eng. In: *Science advances* 10 (21 2024), eadm7765.

[13] Brian D Ripley. "Modelling spatial patterns". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.2 (1977), pp. 172–192.

[14] Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Knowledge Discovery and Data Mining*. 1996. URL: https://api.semanticscholar.org/CorpusID:355163.

[15] Franz Aurenhammer. “Voronoi diagrams—a survey of a fundamental geometric data structure”. In: *ACM Comput. Surv.* 23.3 (Sept. 1991), 345–405. URL: https://doi.org/10.1145/116873.116880.

[16] Jeremy A. Pike et al. “Topological data analysis quantifies biological nano-structure from single molecule localization microscopy”. In: *Bioinformatics* 36.5 (Mar. 2020), pp. 1614–1621. URL: https://doi.org/10.1093/bioinformatics/btz788.

[17] Patrick Rubin-Delanchy et al. “Bayesian cluster identification in single-molecule localization microscopy data”. In: *Nature Methods* 12.11 (2015), pp. 1072–1076. URL: https://doi.org/10.1038/nmeth.3612.

[18] Bennett Kalafut and Koen Visscher. “An objective, model-independent method for detection of non-uniform steps in noisy signals”. In: *Computer Physics Communications* 179.10 (2008), pp. 716–723. URL: https://www.sciencedirect.com/science/article/pii/S0010465508002282.

[19] F. Aboul-ela et al. “Base-base mismatches. Thermodynamics of double helix formation for dCA3XA3G + dCT3YT3G (X, Y = A,C,G,T).” eng. In: *Nucleic acids research* 13 (13 1985), pp. 4811–24.

[20] R. Bruce Wallace et al. “Hybridization of synthetic oligodeoxyribonucleotides to Φ_X 174 DNA: the effect of single base pair mismatch”. In: *Nucleic Acids Research* 6.11 (Aug. 1979), pp. 3543–3558. eprint: https://academic.oup.com/nar/article-pdf/6/11/3543/7063438/6-11-3543.pdf. URL: https://doi.org/10.1093/nar/6.11.3543.

[21] H. T. Allawi and J. Jr SantaLucia. “Thermodynamics and NMR of internal G.T mismatches in DNA.” eng. In: *Biochemistry* 36 (34 1997), pp. 10581–94.

[22] Aleksey Lomakin and Maxim D Frank-Kamenetskii. “A theoretical analysis of specificity of nucleic acid interactions with oligonucleotides and peptide nucleic acids (PNAs)11Edited by I. Tinoco”. In: *Journal of Molecular Biology* 276.1 (1998), pp. 57–70. URL: https://www.sciencedirect.com/science/article/pii/S0022283697914972.

[23] David Yu Zhang, Sherry Xi Chen, and Peng Yin. “Optimizing the specificity of nucleic acid hybridization”. In: *Nature Chemistry* 4.3 (2012), pp. 208–214. URL: https://doi.org/10.1038/nchem.1246.

[24] Fabian Kilchherr et al. “Single-molecule dissection of stacking forces in DNA”. In: *Science* 353.6304 (2016), aaf5508. URL: https://www.science.org/doi/abs/10.1126/science.aaf5508.

[25] Jibin Abraham Punnoose et al. “High-throughput single-molecule quantification of individual base stacking energies in nucleic acids”. In: *Nature Communications* 14.1 (2023), p. 631. URL: https://doi.org/10.1038/s41467-023-36373-8.

[26] Abhinav Banerjee et al. “Single-molecule analysis of DNA base-stacking energetics using patterned DNA nanostructures”. In: *Nature Nanotechnology* 18.12 (2023), pp. 1474–1482. URL: https://doi.org/10.1038/s41565-023-01485-1.

[27] Erik de Oliveira Martins and Gerald Weber. "Nearest-neighbour parametrization of DNA single, double and triple mismatches at low sodium concentration". In: *Biophysical Chemistry* 306 (2024), p. 107156. URL: https://www.sciencedirect.com/science/article/pii/S0301462223002077.

[28] J. Hooyberghs, P. Van Hummelen, and E. Carlon. "The effects of mismatches on hybridization in DNA microarrays: determination of nearest neighbor parameters." eng. In: *Nucleic acids research* 37 (7 2009), e53.

[29] Dipanwita Banerjee et al. "Correction to 'Improved nearest-neighbor parameters for the stability of RNA/DNA hybrids under a physiological condition'". In: *Nucleic Acids Research* 49.18 (Sept. 2021), pp. 10796–10799. eprint: https://academic.oup.com/nar/article-pdf/49/18/10796/40538359/gkab780.pdf. URL: https://doi.org/10.1093/nar/gkab780.

[30] Timothy S. Hall et al. "Sequence Context and Thermodynamic Stability of a Single Base Pair Mismatch in Short Deoxyoligonucleotide Duplexes". In: *J. Am. Chem. Soc.* 123.47 (Nov. 2001), pp. 11811–11812. URL: https://doi.org/10.1021/ja016360j.

[31] Luciana M. Oliveira et al. "Melting temperature measurement and mesoscopic evaluation of single, double and triple DNA mismatches". In: *Chem. Sci.* 11 (31 2020), pp. 8273–8287. URL: http://dx.doi.org/10.1039/D0SC01700K.

[32] M. Peyrard and A. R. Bishop. "Statistical mechanics of a nonlinear model for DNA denaturation". In: *Phys. Rev. Lett.* 62 (23 1989), pp. 2755–2758. URL: https://link.aps.org/doi/10.1103/PhysRevLett.62.2755.

[33] Suxiang Tong et al. "New world bats harbor diverse influenza A viruses." eng. In: *PLoS pathogens* 9 (10 2013), e1003657.

[34] Rimjhim Kanaujia et al. "Avian influenza revisited: concerns and constraints." eng. In: *Virusdisease* 33 (4 2022), pp. 456–465.

[35] Annie Kalonda et al. "Influenza A and D Viruses in Non-Human Mammalian Hosts in Africa: A Systematic Review and Meta-Analysis." eng. In: *Viruses* 13 (12 2021).

[36] Florian Krammer and Peter Palese. "Advances in the development of influenza virus vaccines". In: *Nature Reviews Drug Discovery* 14.3 (2015), pp. 167–182. URL: https://doi.org/10.1038/nrd4529.

[37] Haye Kester et al. "The NS1 Protein of a Human Influenza Virus Inhibits Type I Interferon Production and the Induction of Antiviral Responses in Primary Human Dendritic and Respiratory Epithelial Cells". In: *Journal of Virology* 83.13 (July 2009), pp. 6849–6862. URL: https://doi.org/10.1128/jvi.02323-08.

[38] Robert E. O'Neill, Julie Talon, and Peter Palese. "The influenza virus NEP (NS2 protein) mediates the nuclear export of viral ribonucleoproteins". In: *The EMBO Journal* 17.1 (Jan. 1998), pp. 288–296–296. URL: https://doi.org/10.1093/emboj/17.1.288.

[39] Toby Carter and Munir Iqbal. "The Influenza A Virus Replication Cycle: A Comprehensive Review." eng. In: *Viruses* 16 (2 2024).

[40] Aartjan J. W. te Velthuis and Ervin Fodor. "Influenza virus RNA polymerase: insights into the mechanisms of viral RNA synthesis". In: *Nature Reviews Microbiology* 14.8 (2016), pp. 479–493. URL: https://doi.org/10.1038/nrmicro.2016.87.

[41] W. Compans Richard, Content Jean, and H. Duesberg Peter. "Structure of the Ribonucleoprotein of Influenza Virus". In: *Journal of Virology* 10.4 (Oct. 1972), pp. 795–800. URL: https://doi.org/10.1128/jvi.10.4.795-800.1972.

[42] Valerie Le Sage et al. "Mapping of Influenza Virus RNA-RNA Interactions Reveals a Flexible Network". In: *Cell Reports* 31.13 (June 2020). URL: https://doi.org/10.1016/j.celrep.2020.107823.

[43] Nara Lee et al. "Genome-wide analysis of influenza viral RNA and nucleoprotein association." eng. In: *Nucleic acids research* 45 (15 2017), pp. 8968–8977.

[44] Graham D. Williams et al. "Nucleotide resolution mapping of influenza A virus nucleoprotein-RNA interactions reveals RNA features required for replication". In: *Nature Communications* 9.1 (2018), p. 465. URL: https://doi.org/10.1038/s41467-018-02886-w.

[45] Takeshi Noda. "Selective Genome Packaging Mechanisms of Influenza A Viruses." eng. In: *Cold Spring Harbor perspectives in medicine* 11 (7 2021).

[46] Takeshi Noda et al. "Importance of the 1+7 configuration of ribonucleoprotein complexes for influenza A virus genome packaging". In: *Nature Communications* 9.1 (2018), p. 54. URL: https://doi.org/10.1038/s41467-017-02517-w.

[47] Nakatsu Sumiho et al. "Influenza C and D Viruses Package Eight Organized Ribonucleoprotein Complexes". In: *Journal of Virology* 92.6 (Feb. 2018), 10.1128/jvi.02084–17. URL: https://doi.org/10.1128/jvi.02084-17.

[48] Takeshi Noda et al. "Three-dimensional analysis of ribonucleoprotein complexes in influenza A virus". In: *Nature Communications* 3.1 (2012), p. 639. URL: https://doi.org/10.1038/ncomms1647.

[49] Julia R. Gog et al. "Codon conservation in the influenza A virus genome defines RNA packaging signals". In: *Nucleic Acids Res* 35.6 (Mar. 2007), pp. 1897–1907. URL: https://doi.org/10.1093/nar/gkm087.

[50] Yukiko Muramoto et al. "Hierarchy among Viral RNA (vRNA) Segments in Their Role in vRNA Incorporation into Influenza A Virions". In: *Journal of Virology* 80.5 (2006), pp. 2318–2325. URL: https://journals.asm.org/doi/abs/10.1128/jvi.80.5.2318-2325.2006.

[51] Gao Qinshan et al. "The Influenza A Virus PB2, PA, NP, and M Segments Play a Pivotal Role during Genome Packaging". In: *Journal of Virology* 86.13 (July 2012), pp. 7043–7051. URL: https://doi.org/10.1128/jvi.00662-12.

[52] Yi-ying Chou et al. "One influenza virus particle packages eight unique viral RNAs as shown by FISH analysis." eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 109 (23 2012), pp. 9101–6.

[53] Ivan Haralampiev et al. "Selective flexible packaging pathways of the segmented genome of influenza A virus". In: *Nature Communications* 11.1 (2020), p. 4355. URL: https://doi.org/10.1038/s41467-020-18108-1.

[54] Étori Aguiar Moreira et al. "A conserved influenza A virus nucleoprotein code controls specific viral genome packaging." eng. In: *Nature communications* 7 (2016), p. 12861.

[55] Naoki Takizawa et al. "Local structural changes of the influenza A virus ribonucleoprotein complex by single mutations in the specific residues involved in efficient genome packaging". In: *Virology* 531 (2019), pp. 126–140. URL: https://www.sciencedirect.com/science/article/pii/S0042682219300698.

[56] Bolte Hardin et al. "Packaging of the Influenza Virus Genome Is Governed by a Plastic Network of RNA- and Nucleoprotein-Mediated Interactions". In: *Journal of Virology* 93.4 (Feb. 2019), 10.1128/jvi.01861–18. URL: https://doi.org/10.1128/jvi.01861-18.

[57] Bernadeta Dadonaite et al. "The structure of the influenza A virus genome". In: *Nature Microbiology* 4.11 (2019), pp. 1781–1789. URL: https://doi.org/10.1038/s41564-019-0513-7.

[58] Fang Huang et al. "Video-rate nanoscopy using sCMOS camera-specific single-molecule localization algorithms." eng. In: *Nature methods* 10 (7 2013), pp. 653–8.

[59] Thomas J. Etheridge, Antony M. Carr, and Alex D. Herbert. "GDSC SMLM: Single-molecule localisation microscopy software for ImageJ." eng. In: *Wellcome open research* 7 (2022), p. 241.

[60] Bo Huang et al. "Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy." eng. In: *Science (New York, N.Y.)* 319 (5864 2008), pp. 810–3.

[61] Grégory CLOUVEL, Audrius JASAITIS, and Xavier LEVECQ. "Deep 3D PALM/STORM imaging: MicAO 3DSR–the key to combining depth and highest resolution". In: (2015).

[62] Robert A. Lamb and Griffith D. Parks. "Orthomyxoviridae: The Viruses and Their Replication." In: *Fields Virology*. 1996. URL: https://api.semanticscholar.org/CorpusID:82630032.

[63] Jürgen J. Schmied et al. "DNA origami nanopillars as standards for three-dimensional superresolution microscopy." eng. In: *Nano letters* 13 (2 2013), pp. 781–5.

[64] Chenxiang Lin et al. "Submicrometre geometrically encoded fluorescent barcodes self-assembled from DNA". In: *Nature Chemistry* 4.10 (2012), pp. 832–839. URL: https://doi.org/10.1038/nchem.1451.

[65] Florian Schueder et al. "An order of magnitude faster DNA-PAINT imaging by optimized sequence design and buffer conditions". In: *Nature Methods* 16.11 (2019), pp. 1101–1104. URL: https://doi.org/10.1038/s41592-019-0584-7.

[66] Eitan Lerner et al. "Backtracked and paused transcription initiation intermediate of Escherichia coli RNA polymerase". In: *Proceedings of the National Academy of Sciences* 113.43 (2016), E6562–E6571. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1605038113.

[67] P.R. Selvin and T. Ha. *Single-molecule Techniques: A Laboratory Manual*. G - Reference, Information and Interdisciplinary Subjects Series. Cold Spring Harbor Laboratory Press, 2008. URL: https://books.google.co.uk/books?id=2w6CswEACAAJ.

[68] Paul J. Besl and Neil D. McKay. "A Method for Registration of 3-D Shapes". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992), pp. 239–256. URL: https://api.semanticscholar.org/CorpusID:21874346.

[69] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: *ArXiv* abs/1801.09847 (2018). URL: https://api.semanticscholar.org/CorpusID:34198369.

[70] P. Walker Alexander, Sharps Jane, and Fodor Ervin. "Mutation of an Influenza Virus Polymerase 3' RNA Promoter Binding Site Inhibits Transcription Elongation". In: *Journal of Virology* 94.13 (June 2020), 10.1128/jvi.00498–20. URL: https://doi.org/10.1128/jvi.00498-20.

[71] Tony Ridler. "Picture thresholding using an iterative selection method." In: *IEEE Transactions on Systems, Man, and Cybernetics* 8 (1978), pp. 630–632. URL: https://api.semanticscholar.org/CorpusID:118436421.

[72] B.J.H. Verwer. "Improved metrics in image processing applied to the Hilditch skeleton". In: *[1988 Proceedings] 9th International Conference on Pattern Recognition*. 1988, 137–142 vol.1.

[73] Stan Birchfield. "Image processing and analysis". In: *(No Title)* (2018).

[74] Audray Harris et al. "Influenza virus pleiomorphy characterized by cryoelectron tomography." eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 103 (50 2006), pp. 19123–7.

[75] Qiuyu J. Huang et al. "Quantitative structural analysis of influenza virus by cryo-electron tomography and convolutional neural networks." eng. In: *Structure (London, England : 1993)* 30 (5 2022), 777–786.e3.

[76] Zach Marin et al. "PYMEVisualize: an open-source tool for exploring 3D super-resolution data". In: *Nature Methods* 18.6 (2021), pp. 582–584. URL: https://doi.org/10.1038/s41592-021-01165-9.

[77] S. Schaefer and J. Warren. "Dual marching cubes: primal contouring of dual grids". In: *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.* 2004, pp. 70–76.

[78] Desirée Salas et al. "Angular reconstitution-based 3D reconstructions of nanomolecular structures from superresolution light-microscopy images". In: *Proceedings of the National Academy of Sciences* 114.35 (2017), pp. 9273–9278. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1704908114.

[79] Hamidreza Heydarian et al. "3D particle averaging and detection of macromolecular symmetry in localization microscopy". In: *Nature Communications* 12.1 (2021), p. 2847. URL: https://doi.org/10.1038/s41467-021-22006-5.

[80] Christian Sieben et al. "Multicolor single-particle reconstruction of protein complexes." eng. In: *Nature methods* 15 (10 2018), pp. 777–780.

[81] Sjors H. W. Scheres et al. “Maximum-likelihood multi-reference refinement for electron microscopy images.” eng. In: *Journal of molecular biology* 348 (1 2005), pp. 139–49.

[82] Guang Tang et al. “EMAN2: an extensible image processing suite for electron microscopy.” eng. In: *Journal of structural biology* 157 (1 2007), pp. 38–46.

[83] J. M. de la Rosa-Trevín et al. “Xmipp 3.0: an improved software suite for image processing in electron microscopy.” eng. In: *Journal of structural biology* 184 (2 2013), pp. 321–8.

[84] Yiqiu Wang, Yan Gu, and Julian Shun. “Theoretically-Efficient and Practical Parallel DBSCAN”. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (2019). URL: https://api.semanticscholar.org/CorpusID:209370510.

[85] Rebecca Killick, Paul Fearnhead, and Idris Arthur Eckley. “Optimal detection of changepoints with a linear computational cost”. In: *Journal of the American Statistical Association* 107 (2011), pp. 1590 –1598. URL: https://api.semanticscholar.org/CorpusID:5627005.

[86] Charles Truong, Laurent Oudre, and Nicolas Vayatis. “Selective review of offline change point detection methods”. In: *Signal Processing* 167 (2020), p. 107299. URL: https://www.sciencedirect.com/science/article/pii/S0165168419303494.

[87] Jiawei Han, Jian Pei, and Yiwen Yin. “Mining frequent patterns without candidate generation”. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. SIGMOD ’00. Dallas, Texas, USA: Association for Computing Machinery, 2000, 1–12. URL: https://doi.org/10.1145/342009.335372.

[88] Vincent Antonio Traag, Ludo Waltman, and Nees Jan van Eck. “From Louvain to Leiden: guaranteeing well-connected communities”. In: *Scientific Reports* 9 (2018). URL: https://api.semanticscholar.org/CorpusID:53041707.

[89] M. E. J. Newman and M. Girvan. “Finding and evaluating community structure in networks”. In: *Phys. Rev. E* 69 (2 2004), p. 026113. URL: https://link.aps.org/doi/10.1103/PhysRevE.69.026113.

[90] Jörg Reichardt and Stefan Bornholdt. “Statistical mechanics of community detection”. In: *Phys. Rev. E* 74 (1 2006), p. 016110. URL: https://link.aps.org/doi/10.1103/PhysRevE.74.016110.

[91] Alexander Auer et al. “Fast, Background-Free DNA-PAINT Imaging Using FRET-Based Probes”. In: *Nano Lett.* 17.10 (Oct. 2017), pp. 6428–6434. URL: https://doi.org/10.1021/acs.nanolett.7b03425.

[92] Jongjin Lee, Sangjun Park, and Sungchul Hohng. “Accelerated FRET-PAINT microscopy”. In: *Molecular Brain* 11.1 (2018), p. 70. URL: https://doi.org/10.1186/s13041-018-0414-3.

[93] Labros G. Meimetis et al. “Ultrafluorogenic coumarin-tetrazine probes for real-time biological imaging.” eng. In: *Angewandte Chemie (International ed. in English)* 53 (29 2014), pp. 7531–4.

[94] Rocío Arranz et al. “The structure of native influenza virion ribonucleoproteins.” eng. In: *Science (New York, N.Y.)* 338 (6114 2012), pp. 1634–7.

[95] Arne Moeller et al. “Organization of the influenza virus replication machinery.” eng. In: *Science (New York, N.Y.)* 338 (6114 2012), pp. 1631–4.

[96] Robert P. J. Nieuwenhuizen et al. “Measuring image resolution in optical nanoscopy”. In: *Nature Methods* 10.6 (2013), pp. 557–562. URL: https://doi.org/10.1038/nmeth.2448.

[97] Roland Beckmann et al. “Alignment of Conduits for the Nascent Polypeptide Chain in the Ribosome-Sec61 Complex”. In: *Science* 278.5346 (Dec. 1997), pp. 2123–2126. URL: https://doi.org/10.1126/science.278.5346.2123.

[98] Adrian Przybylski et al. “Gpufit: An open-source toolkit for GPU-accelerated curve fitting”. In: *Scientific Reports* 7.1 (2017), p. 15722. URL: https://doi.org/10.1038/s41598-017-15313-9.

[99] Michael A. Quail et al. “A large genome center’s improvements to the Illumina sequencing system”. In: *Nature Methods* 5.12 (2008), pp. 1005–1010. URL: https://doi.org/10.1038/nmeth.1270.

[100] Yunhao Wang et al. “Nanopore sequencing technology, bioinformatics and applications”. In: *Nature Biotechnology* 39.11 (2021), pp. 1348–1365. URL: https://doi.org/10.1038/s41587-021-01108-x.

[101] P. Thévenaz, U. E. Ruttimann, and M. Unser. “A pyramid approach to subpixel registration based on intensity.” eng. In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 7 (1 1998), pp. 27–41.

[102] Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup. “Efficient subpixel image registration algorithms”. In: *Opt. Lett.* 33.2 (2008), pp. 156–158. URL: https://opg.optica.org/ol/abstract.cfm?URI=ol-33-2-156.

[103] J. Aguirre Rivera et al. “Massively parallel analysis of single-molecule dynamics on next-generation sequencing chips.” eng. In: *Science (New York, N.Y.)* 385 (6711 2024), pp. 892–898.

[104] Carolien Bastiaanssen et al. “Single-molecule parallel analysis for rapid exploration of sequence space”. In: *Nature Protocols* (2025). URL: https://doi.org/10.1038/s41596-025-01196-y.

[105] M. Panfilov et al. “Multiplexed single-molecule characterization at the library scale”. In: *Nature Protocols* (2025). URL: https://doi.org/10.1038/s41596-025-01198-w.

[106] Ivo Severins et al. “Single-molecule structural and kinetic studies across sequence space”. In: *Science* 385.6711 (2024), pp. 898–904. URL: https://www.science.org/doi/abs/10.1126/science.adn5968.

[107] S. Busby and R. H. Ebright. “Transcription activation by catabolite activator protein (CAP).” eng. In: *Journal of molecular biology* 293 (2 1999), pp. 199–213.

[108] Gary Parkinson et al. “Aromatic hydrogen bond in sequence-specific protein DNA recognition”. In: *Nature Structural Biology* 3.10 (1996), pp. 837–841. URL: https://doi.org/10.1038/nsb1096-837.

[109] R. H. Ebright, Y. W. Ebright, and A. Gunasekera. "Consensus DNA site for the Escherichia coli catabolite gene activator protein (CAP): CAP exhibits a 450-fold higher affinity for the consensus DNA site than for the E. coli lac DNA site." eng. In: *Nucleic acids research* 17 (24 1989), pp. 10295–305.

[110] A Gunasekera, Y.W. Ebright, and R.H. Ebright. "DNA sequence determinants for binding of the Escherichia coli catabolite gene activator protein." In: *Journal of Biological Chemistry* 267.21 (1992), pp. 14713–14720. URL: https://www.sciencedirect.com/science/article/pii/S0021925818420996.

[111] R H Ebright et al. "Role of glutamic acid-181 in DNA-sequence recognition by the catabolite gene activator protein (CAP) of Escherichia coli: altered DNA-sequence-recognition properties of [Val181]CAP and [Leu181]CAP." In: *Proceedings of the National Academy of Sciences* 84.17 (1987), pp. 6083–6087. URL: https://www.pnas.org/doi/abs/10.1073/pnas.84.17.6083.

[112] Richard H. Ebright et al. "Mutations that alter the DNA sequence specificity of the catabolite gene activator protein of E. coli". In: *Nature* 311.5983 (1984), pp. 232–235. URL: https://doi.org/10.1038/311232a0.

[113] Gary Parkinson et al. "Structure of the CAP-DNA Complex at 2.5 Å Resolution: A Complete Picture of the Protein-DNA Interface". In: *Journal of Molecular Biology* 260.3 (1996), pp. 395–408. URL: https://www.sciencedirect.com/science/article/pii/S002228369690409X.

[114] Steve C. Schultz, George C. Shields, and Thomas A. Steitz. "Crystal Structure of a CAP-DNA Complex: the DNA Is Bent by 90°". In: *Science* 253.5023 (1991), pp. 1001–1007. URL: https://www.science.org/doi/abs/10.1126/science.1653449.

[115] Achillefs N. Kapanidis et al. "Mean DNA Bend Angle and Distribution of DNA Bend Angles in the CAP-DNA Complex in Solution". In: *Journal of Molecular Biology* 312.3 (2001), pp. 453–468. URL: https://www.sciencedirect.com/science/article/pii/S0022283601949769.

[116] Catherine L. Lawson et al. "Catabolite activator protein: DNA binding and transcription activation." eng. In: *Current opinion in structural biology* 14 (1 2004), pp. 10–20.

[117] Ryan Prescott Adams and David J. C. MacKay. "Bayesian Online Changepoint Detection". In: (2007). arXiv: 0710.3742 [stat.ML]. URL: https://arxiv.org/abs/0710.3742.

[118] Daniel Barry and J. A. Hartigan. "Product Partition Models for Change Point Problems". In: *The Annals of Statistics* 20.1 (Mar. 1992), pp. 260–279. URL: https://doi.org/10.1214/aos/1176348521.

[119] N. M. Maizels. "The nucleotide sequence of the lactose messenger ribonucleic acid transcribed from the UV5 promoter mutant of Escherichia coli." eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 70 (12 1973), pp. 3585–9.

[120] Robert Landick. "Transcriptional Pausing as a Mediator of Bacterial Gene Regulation." eng. In: *Annual review of microbiology* 75 (2021), pp. 291–314.

[121] John P. Richardson. “Preventing the synthesis of unused transcripts by rho factor”. In: *Cell* 64.6 (1991), pp. 1047–1049. URL: https://www.sciencedirect.com/science/article/pii/009286749190257Y.

[122] Irina Artsimovitch and Robert Landick. “The transcriptional regulator RfaH stimulates RNA chain synthesis after recruitment to elongation complexes by the exposed nontemplate DNA strand.” eng. In: *Cell* 109 (2 2002), pp. 193–203.

[123] L F Lau, J W Roberts, and R Wu. “RNA polymerase pausing and transcript release at the lambda tR1 terminator in vitro.” In: *Journal of Biological Chemistry* 258.15 (1983), pp. 9391–9397. URL: https://www.sciencedirect.com/science/article/pii/S0021925817446801.

[124] Tao Pan et al. “Folding of a large ribozyme during transcription and the effect of the elongation factor NusA”. In: *Proceedings of the National Academy of Sciences* 96.17 (Aug. 1999), pp. 9545–9550. URL: https://doi.org/10.1073/pnas.96.17.9545.

[125] Achillefs N. Kapanidis et al. “Initial Transcription by RNA Polymerase Proceeds Through a DNA-Scrunching Mechanism”. In: *Science* 314.5802 (Nov. 2006), pp. 1144–1147. URL: https://doi.org/10.1126/science.1131399.

[126] Andrey Revyakin et al. “Abortive Initiation and Productive Initiation by RNA Polymerase Involve DNA Scrunching”. In: *Science* 314.5802 (2006), pp. 1139–1143. URL: https://www.science.org/doi/abs/10.1126/science.1131398.

[127] Lilian M. Hsu. “Monitoring abortive initiation”. In: *Methods* 47.1 (2009), pp. 25–36. URL: https://www.sciencedirect.com/science/article/pii/S1046202308001965.

[128] David Dulin et al. “Pausing controls branching between productive and non-productive pathways during initial transcription in bacteria”. In: *Nature Communications* 9.1 (2018), p. 1478. URL: https://doi.org/10.1038/s41467-018-03902-9.

[129] Diego Duchi et al. “RNA Polymerase Pausing during Initial Transcription.” eng. In: *Molecular cell* 63 (6 2016), pp. 939–50.

[130] Vasisht R. Tadigotla et al. “Thermodynamic and kinetic modeling of transcriptional pausing”. In: *Proceedings of the National Academy of Sciences* 103.12 (2006), pp. 4439–4444. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0600508103.

[131] E. Nudler et al. “The RNA-DNA hybrid maintains the register of transcription by preventing backtracking of RNA polymerase.” eng. In: *Cell* 89 (1 1997), pp. 33–41.

[132] T. Dertinger et al. “Fast, background-free, 3D super-resolution optical fluctuation imaging (SOFI)”. In: *Proceedings of the National Academy of Sciences* 106.52 (2009), pp. 22287–22292. URL: https://www.pnas.org/doi/abs/10.1073/pnas.0907866106.

[133] Susan Cox et al. “Bayesian localization microscopy reveals nanoscale podosome dynamics”. In: *Nature Methods* 9.2 (2012), pp. 195–200. URL: https://doi.org/10.1038/nmeth.1812.

[134] Łukasz Kidziński et al. "Deep neural networks enable quantitative movement analysis using single-camera videos". In: *Nature Communications* 11.1 (2020), p. 4054. URL: https://doi.org/10.1038/s41467-020-17807-z.

[135] Sahar Fazeli, Judith Sabetti, and Manuela Ferrari. "Performing Qualitative Content Analysis of Video Data in Social Sciences and Medicine: The Visual-Verbal Video Analysis Method". In: *International Journal of Qualitative Methods* 22 (2023), p. 16094069231185452. URL: https://doi.org/10.1177/16094069231185452.

[136] Boxiao Pan et al. "Spatio-Temporal Graph for Video Captioning With Knowledge Distillation". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10867–10876.

[137] R. Geethanjali and A. Valarmathi. "A novel hybrid deep learning IChOA-CNN-LSTM model for modality-enriched and multilingual emotion recognition in social media". In: *Scientific Reports* 14.1 (2024), p. 22270. URL: https://doi.org/10.1038/s41598-024-73452-2.

[138] Kristian Rechendorff et al. "Persistence length and scaling properties of single-stranded DNA adsorbed on modified graphite." eng. In: *The Journal of chemical physics* 131 (9 2009), p. 095103.

[139] M. C. Murphy et al. "Probing single-stranded DNA conformational flexibility using fluorescence spectroscopy." eng. In: *Biophysical journal* 86 (4 2004), pp. 2530–7.

[140] Terence Strick et al. "Twisting and stretching single DNA molecules". In: *Progress in Biophysics and Molecular Biology* 74.1 (2000). Single Molecule Biochemistry and Molecular Biology, pp. 115–140. URL: https://www.sciencedirect.com/science/article/pii/S0079610700000183.

[141] Anna Archetti et al. "Waveguide-PAINT offers an open platform for large field-of-view super-resolution imaging". In: *Nature Communications* 10.1 (2019), p. 1267. URL: https://doi.org/10.1038/s41467-019-09247-1.

[142] Anish Priyadarshi et al. "A transparent waveguide chip for versatile total internal reflection fluorescence-based microscopy and nanoscopy". In: *Communications Materials* 2.1 (2021), p. 85. URL: https://doi.org/10.1038/s43246-021-00192-5.

[143] Daniel Sage et al. "Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software". In: *Nature Methods* 16.5 (2019), pp. 387–395. URL: https://doi.org/10.1038/s41592-019-0364-4.

[144] Klarinda de Zwaan et al. "High-Throughput Single-Molecule Microscopy with Adaptable Spatial Resolution Using Exchangeable Oligonucleotide Labels". In: *ACS Nano* 19.13 (Apr. 2025), pp. 13149–13159. URL: https://doi.org/10.1021/acsnano.4c18502.

[145] Christof Hepp et al. "High-throughput single-virion DNA-PAINT reveals structural diversity, cooperativity, and flexibility during selective packaging in influenza". In: *Nucleic Acids Res* 53.19 (Oct. 2025), gkaf1020. URL: https://doi.org/10.1093/nar/gkaf1020.

[146] Mirjam Kümmerlin et al. “Tunable fluorogenic DNA probes drive fast and high-resolution single-molecule fluorescence imaging”. In: *bioRxiv* (Jan. 2025), p. 2025.01.21.634148. URL: http://biorxiv.org/content/early/2025/01/24/2025.01.21.634148.abstract.

[147] Nobuhiro Morone et al. “Improved unroofing protocols for cryo-electron microscopy, atomic force microscopy and freeze-etching electron microscopy and the associated mechanisms”. In: *Microscopy (Oxf)* 69.6 (Dec. 2020), pp. 350–359. URL: https://doi.org/10.1093/jmicro/dfaa028.

[148] Eiji Usukura et al. “An Unroofing Method to Observe the Cytoskeleton Directly at Molecular Resolution Using Atomic Force Microscopy”. In: *Scientific Reports* 6.1 (2016), p. 27472. URL: https://doi.org/10.1038/srep27472.

[149] Yuki Umeda et al. “Microfluidic cell unroofing for the in situ molecular analysis of organelles without membrane permeabilization”. In: *Lab Chip* 25 (9 2025), pp. 2222–2233. URL: http://dx.doi.org/10.1039/D5LC00102A.

[150] Shuqi Zhou et al. “Deep learning based local feature classification to automatically identify single molecule fluorescence events”. In: *Communications Biology* 7.1 (2024), p. 1404. URL: https://doi.org/10.1038/s42003-024-07122-4.

[151] Johannes Thomsen et al. “DeepFRET, a software for rapid and automated single-molecule FRET data classification using deep learning”. In: *eLife* 9 (2020). Ed. by Sebastian Deindl, Suzanne R Pfeffer, and Shixin Liu, e60404. URL: https://doi.org/10.7554/eLife.60404.

[152] Leyou Zhang, Jieming Li, and Nils G. Walter. “Pretrained Deep Neural Network Kin-SiM for Single-Molecule FRET Trace Idealization”. In: *J. Phys. Chem. B* 129.4 (Jan. 2025), pp. 1167–1175. URL: https://doi.org/10.1021/acs.jpcb.4c05276.

Appendices

A Sequence designs

A.1 Barcoded strands in IAV

Segment	Probe number	Hybridization Part (5' –>3')	Imager Target (5' –>3')
PB2	1	TTCGGATGGCCATCAATTAG	TTATCTACATA
PB2	2	CATACTTACTGACAGCCAGA	TTATCTACATA
PB2	3	AGACGTGGTGTTGGTAATGA	TTATCTACATA
PB2	4	GGAGAGAAGGCTAATGTGCT	TTATCTACATA
PB2	5	ATGAACTGAGCAACCTTGCG	TTATCTACATA
PB2	6	AACAAGAGATATGGGCCAGC	TTATCTACATA
PB2	7	ATTCCTCATTCTGGGCAAAG	TTATCTACATA
PB2	8	TGTGAGGGGATCAGGAATGA	TTATCTACATA
PB2	9	AGTTCTCCTCATTTACTGTG	TTATCTACATA
PB2	10	GCTCCACCAAAGCAAAGTAG	TTATCTACATA
PB2	11	ATAAAACTTCTTCCCTTCGC	TTATCTACATA
PB2	12	GACATTTGATACCGCACAGA	TTATCTACATA
PB2	13	TTAGAGGCCAATACAGTGGG	TTATCTACATA
PB2	14	CAGTCTTTAGTACCTAAGGC	TTATCTACATA
PB2	15	TCAGTGTTGGTCAATACCTA	TTATCTACATA
PB2	16	CTCATCGTCAATGATGTGGG	TTATCTACATA
PB2	17	AGGTCAGTGAAACACAGGGA	TTATCTACATA
PB2	18	GAAATGTACTACTGTCTCCC	TTATCTACATA
PB2	19	GACCGTTTTTTGAGAATCCG	TTATCTACATA
PB2	20	TAGATGAGTACTCCAGCACG	TTATCTACATA
PB2	21	TGACTCCAAGCATCGAGATG	TTATCTACATA
PB2	22	GAATGATTGGGATATTGCCC	TTATCTACATA
PB2	23	ATTGGGGAGTTGAACCTATC	TTATCTACATA

Table A.1 continued from previous page

PB2	24	ATAGGGCGAATCAGCGATTG	TTATCTACATA
PB2	25	AAGCAGTCAGAGGTGATCTG	TTATCTACATA
PB2	26	GCCATGGTATTTTCACAAGA	TTATCTACATA
PB2	27	AGTCGATTGCCGAAGCAATA	TTATCTACATA
PB2	28	GCTGATAGTGAGTGGGAGAG	TTATCTACATA
PB2	29	GAGTTCACAATGGTTGGGAG	TTATCTACATA
PB2	30	CTTCAGTTTTGGTGGATTCA	TTATCTACATA
PB2	31	GCTGCAATGGGACTGAGAAT	TTATCTACATA
PB2	32	CAACAGAAGAGCAAGCCGTG	TTATCTACATA
PB2	33	GTAGACATCCTTAGGCAGAA	TTATCTACATA
PB2	34	TTTATTGGAGATGTGCCACA	TTATCTACATA
PB2	35	AGTATCAGCAGATCCACTAG	TTATCTACATA
PB2	36	AGCTTGATTATTGCTGCTAG	TTATCTACATA
PB2	37	ACTCAAGGAACATGCTGGGA	TTATCTACATA
PB2	38	AGAGAGAACTGGTCCGCAAA	TTATCTACATA
PB2	39	CCTTTGATGGTTGCATACAT	TTATCTACATA
PB2	40	GAGCCAGGATACTAACATCG	TTATCTACATA
PB2	41	GTTGTTTTCCCTAACGAAGT	TTATCTACATA
PB2	42	GGCACAGGATGTAATCATGG	TTATCTACATA
PB2	43	AAATCCTGGTCATGCAGATC	TTATCTACATA
PB2	44	CCTGTCCATTTTAGAAACCA	TTATCTACATA
PB2	45	AGGCTAAAGCATGGAACCTT	TTATCTACATA
PB2	46	GCTGTGACATGGTGGAATAG	TTATCTACATA
PB2	47	ATCAGACCGAGTGATGGTAT	TTATCTACATA
PB2	48	CTCACAAAAACCACCGTGGA	TTATCTACATA
PB1	1	AGTGAATTTAGCTTGTCCTT	TTATACATCTA
PB1	2	TTGAAGAGCTCAGACGGCAA	TTATACATCTA
PB1	3	CCGAATTGATGCACGGATTG	TTATACATCTA
PB1	4	CAGAAGACCAGTCGGGATAT	TTATACATCTA
PB1	5	GGAGTATGATGCTGTTGCAA	TTATACATCTA
PB1	6	CACATGGTCCAGCCAAAAAC	TTATACATCTA
PB1	7	GTTTATGCAACCCACTGAAC	TTATACATCTA
PB1	8	ATGGATGAGGATTACCAGGG	TTATACATCTA
PB1	9	ATTCCTGAAGTCTGCCTAAA	TTATACATCTA
PB1	10	ACGGAGGCCCAAATTTATAC	TTATACATCTA
PB1	11	TACAGGTACACGTACCGATG	TTATACATCTA
PB1	12	CCCTTCAGTTGTTCATCAAA	TTATACATCTA
PB1	13	AATGATCTTGGTCCAGCAAC	TTATACATCTA
PB1	14	AACGAGTCAGCGGACATGAG	TTATACATCTA
PB1	15	TTGTTGCCAATTTCAGCATG	TTATACATCTA
PB1	16	CGGAGTCGACAGGTTTTATC	TTATACATCTA
PB1	17	CACCCAATCATGAAGGGATT	TTATACATCTA
PB1	18	CAATCCTCTGACGATTTTGC	TTATACATCTA

Table A.1 continued from previous page

PB1	19	CCAAGACTACTTACTGGTGG	TTATACATCTA
PB1	20	TAAGCACTGTATTAGGCGTC	TTATACATCTA
PB1	21	GGAATGATGATGGGCATGTT	TTATACATCTA
PB1	22	TCGCTCTTAATAGAGGGGAC	TTATACATCTA
PB1	23	ACCTGCAGAAATGCTAGCAA	TTATACATCTA
PB1	24	GCGAGACTGGGAAAAGGGTA	TTATACATCTA
PB1	25	ATGTTCTAAGTATTGCTCCA	TTATACATCTA
PB1	26	ATATGACCAGAAATCAGCCC	TTATACATCTA
PB1	27	GAATCCTCGGATGTTTTTGG	TTATACATCTA
PB1	28	ACTTTCTTTCACCATCACTG	TTATACATCTA
PB1	29	GATGACCAATTCTCAGGACA	TTATACATCTA
PB1	30	TGAACAATCAGGGTTGCCAG	TTATACATCTA
PB1	31	TGAGACACTGGCAAGGAGTA	TTATACATCTA
PB1	32	AGGGATGCAAATAAGGGGGT	TTATACATCTA
PB1	33	AACGGAGAGCAATTGCAACC	TTATACATCTA
PB1	34	AGATGCTGAGAGAGGGAAGC	TTATACATCTA
PB1	35	TTAGAGCATTGACCCTGAAC	TTATACATCTA
PB1	36	GGCTCATAGACTTCCTTAAG	TTATACATCTA
PB1	37	TGTTCAGATCAAATGGCCTC	TTATACATCTA
PB1	38	GCATTGGCCAACACAATAGA	TTATACATCTA
PB1	39	TAGAAACCAACCTGCTGCAA	TTATACATCTA
PB1	40	AGACCTATGACTGGACTCTA	TTATACATCTA
PB1	41	GGAGGTTGTTCAGCAAACAC	TTATACATCTA
PB1	42	GATTGTGTATTGGAGGCGAT	TTATACATCTA
PB1	43	ATGAACCAAGTGGTTATGCC	TTATACATCTA
PB1	44	CAACTCAACCCGATTGATGG	TTATACATCTA
PB1	45	AGCACAACTTTCCCTTATAC	TTATACATCTA
PB1	46	AAGTGCCAGCACAAAATGCT	TTATACATCTA
PB1	47	TCAATCCGACCTTACTTTTC	TTATACATCTA
PB1	48	GCGAAAGCAGGCAAACCATT	TTATACATCTA
PA	1	TGCTATCCATACTGTCCAAA	TTATGAATCTA
PA	2	ATTGAGTTAGTTGTGGCAGT	TTATGAATCTA
PA	3	GGTTCAACTCATTCCTTACA	TTATGAATCTA
PA	4	TGGGTTTTGCTTAATGCTTC	TTATGAATCTA
PA	5	GAGGAGTGCCTGATTAATGA	TTATGAATCTA
PA	6	ATCTTGGGGGGCTATATGAA	TTATGAATCTA
PA	7	CTTATCGTTCAGGCTCTTAG	TTATGAATCTA
PA	8	GCATCTCCACAACTAGAAGG	TTATGAATCTA
PA	9	AGTCGGTATTCAACAGCTTG	TTATGAATCTA
PA	10	GGTCTGCAGGACTTTATTAG	TTATGAATCTA
PA	11	GTGGAGGAAAGTTCCATTGG	TTATGAATCTA
PA	12	ATCAGAAACATGGCCCATTG	TTATGAATCTA
PA	13	CTGAGTCCTCTGTCAAAGAG	TTATGAATCTA

Table A.1 continued from previous page

PA	14	CCTCCAGTCACTTCAACAAA	TTATGAATCTA
PA	15	CCATGTTCTTGTATGTGAGA	TTATGAATCTA
PA	16	TATAAGAAGTGCCATAGGCC	TTATGAATCTA
PA	17	TGACCCAAGACTTGAACCAC	TTATGAATCTA
PA	18	AGGAAGATCCCACTTAAGGA	TTATGAATCTA
PA	19	GACCAACTTGTATGGTTTCA	TTATGAATCTA
PA	20	GAACTAAGGAGGGAAGGCGA	TTATGAATCTA
PA	21	TTGCTTAATGCATCTTGTGC	TTATGAATCTA
PA	22	GGGGGTGTACATCAATACTG	TTATGAATCTA
PA	23	TTTCACATCAGAGGTGTCTC	TTATGAATCTA
PA	24	TTCAAGCTGGATAGAGCTCG	TTATGAATCTA
PA	25	ACAAGGCATGCGAACTGACA	TTATGAATCTA
PA	26	GGCACCAGAAAAGGTAGACT	TTATGAATCTA
PA	27	AAGTGGGCACTTGGTGAGAA	TTATGAATCTA
PA	28	CTTCTGTCATGGAAGCAAGT	TTATGAATCTA
PA	29	ACCGCTATATGATGCAATCA	TTATGAATCTA
PA	30	GTCCAAATTCCTACTGATGG	TTATGAATCTA
PA	31	CACTTAGACTTCCGAATGGG	TTATGAATCTA
PA	32	CAAGCTGTCTCAAATGTCCA	TTATGAATCTA
PA	33	GAACCGAACGGCTACATTGA	TTATGAATCTA
PA	34	TAGAGCCTATGTGGATGGAT	TTATGAATCTA
PA	35	ACTTCTCCAGCCTTGAAAAT	TTATGAATCTA
PA	36	CACAGGAACAATGCGCAAGC	TTATGAATCTA
PA	37	GGGCTAGGATCAAAACCAGA	TTATGAATCTA
PA	38	CCACAAAGGCAGACTACACT	TTATGAATCTA
PA	39	TCTCGTTCACTGGGGAAGAA	TTATGAATCTA
PA	40	GGGGCTGAGAAACCAAAGTT	TTATGAATCTA
PA	41	TGGCCTGGACAGTAGTAAAC	TTATGAATCTA
PA	42	ATCGAGGGAAGAGATCGCAC	TTATGAATCTA
PA	43	GCAAGGCGAGTCAATAATCG	TTATGAATCTA
PA	44	TTGCAGCAATATGCACTCAC	TTATGAATCTA
PA	45	GAAAGAGTATGGGGAGGACC	TTATGAATCTA
PA	46	GATTGTCGAGCTTGCGGAAA	TTATGAATCTA
PA	47	TTTTGTGCGACAATGCTTCA	TTATGAATCTA
PA	48	CGAAAGCAGGTACTGATCCA	TTATGAATCTA
HA	1	TGCAGAATATGCATCTGAGA	TTTCAATGTAT
HA	2	GTGTTCTAATGGATCTTTGC	TTTCAATGTAT
HA	3	CTGGGGGCAATCAGTTTCTG	TTTCAATGTAT
HA	4	CAGTTCACTGGTGCTTTTGG	TTTCAATGTAT
HA	5	TGGCGATCTACTCAACTGTC	TTTCAATGTAT
HA	6	ATCAATGGGGATCTATCAGA	TTTCAATGTAT
HA	7	GTGTGACAATGAATGCATGG	TTTCAATGTAT
HA	8	GGATGTTTTGAGTTCTACCA	TTTCAATGTAT

Table A.1 continued from previous page

HA	9	TCTGGATTTCCATGACTCAA	TTTCAATGTAT
HA	10	GGATTTCTGGACATTTGGAC	TTTCAATGTAT
HA	11	GGTGAACACTGTTATCGAGA	TTTCAATGTAT
HA	12	TGCCATTAACGGGATTACAA	TTTCAATGTAT
HA	13	TTGCCGGTTTTATTGAAGGG	TTTCAATGTAT
HA	14	TCCAGAGGTCTATTTGGAGC	TTTCAATGTAT
HA	15	AAGGAACAATCCGTCCATTC	TTTCAATGTAT
HA	16	AATTGAGGATGGTTACAGGA	TTTCAATGTAT
HA	17	CCCAAAATACGTCAGGAGTG	TTTCAATGTAT
HA	18	ACCCAGTCACAATAGGAGAG	TTTCAATGTAT
HA	19	AGTCTCCCTTACCAGAATAT	TTTCAATGTAT
HA	20	ACCCCTGGGAGCTATAAACA	TTTCAATGTAT
HA	21	GCATGAGTGTAACACGAAGT	TTTCAATGTAT
HA	22	ATCATCACCTCAAACGCATC	TTTCAATGTAT
HA	23	CACTGAGTAGAGGCTTTGGG	TTTCAATGTAT
HA	24	ATAGCACCAATGTATGCTTT	TTTCAATGTAT
HA	25	ACTATTACTGGACCTTGCTA	TTTCAATGTAT
HA	26	AGAGATCAAGCTGGGAGGAT	TTTCAATGTAT
HA	27	CCCCGGAAATAGCAGAAAGA	TTTCAATGTAT
HA	28	GAAAATGCTTATGTCTCTGT	TTTCAATGTAT
HA	29	CGCCTAACAGTAAGGAACAA	TTTCAATGTAT
HA	30	CGGAGAAGGAGGGCTCATAC	TTTCAATGTAT
HA	31	TACAGAAATTTGCTATGGCT	TTTCAATGTAT
HA	32	CCATGAGGGGAAAAGCAGTT	TTTCAATGTAT
HA	33	ACAAACGGAGTAACGGCAGC	TTTCAATGTAT
HA	34	AAAGAAAGCTCATGGCCCAA	TTTCAATGTAT
HA	35	ATTGAGCTCAGTGTCATCAT	TTTCAATGTAT
HA	36	CATCGACTATGAGGAGCTGA	TTTCAATGTAT
HA	37	CCTACATTGTAGAAACACCA	TTTCAATGTAT
HA	38	CTGCTTCCAGTGAGATCATG	TTTCAATGTAT
HA	39	CTCTTGGGAAACCCAGAATG	TTTCAATGTAT
HA	40	GGAAATGTAACATCGCCGGA	TTTCAATGTAT
HA	41	AAGGAATAGCCCCACTACAA	TTTCAATGTAT
HA	42	GCCACAACGGAAAACTATGT	TTTCAATGTAT
HA	43	TCTGTTAACCTGCTCGAAGA	TTTCAATGTAT
HA	44	CTCGAGAAGAATGTGACAGT	TTTCAATGTAT
HA	45	TCAACCGACACTGTTGACAC	TTTCAATGTAT
HA	46	TATAGGCTACCATGCGAACA	TTTCAATGTAT
HA	47	GCTGCAGATGCAGACACAAT	TTTCAATGTAT
HA	48	TGAAGGCAAACCTACTGGTC	TTTCAATGTAT
NP	1	CGAGCTCTCGGACGAAAAGG	TTATGTTAATG
NP	2	AGAAGATGTGTCTTTCCAGG	TTATGTTAATG
NP	3	CATCTGACATGAGGACCGAA	TTATGTTAATG

Table A.1 continued from previous page

NP	4	TGGCAGCATTCAATGGGAAT	TTATGTTAATG
NP	5	ATCTCCCTTTTGACAGAACA	TTATGTTAATG
NP	6	TCAGCATACAACCTACGTTC	TTATGTTAATG
NP	7	AATCAACAGAGGGCATCTGC	TTATGTTAATG
NP	8	CCATAAGGACCAGAAGTGGA	TTATGTTAATG
NP	9	AAGAGGGAAGCTTTCCACTA	TTATGTTAATG
NP	10	TCAAAGGGACGAAGGTGCTC	TTATGTTAATG
NP	11	ATTCTGCCGCATTTGAAGAT	TTATGTTAATG
NP	12	TCCAGCACACAAGAGTCAAC	TTATGTTAATG
NP	13	GTACAGCCTAATCAGACCAA	TTATGTTAATG
NP	14	CAGACTGCTTCAAAACAGCC	TTATGTTAATG
NP	15	CTCTCTAGTCGGAATAGACC	TTATGTTAATG
NP	16	CTGCACTCATATTGAGAGGG	TTATGTTAATG
NP	17	GATCTCACTTTTCTAGCACG	TTATGTTAATG
NP	18	GATCAAGTGAGAGAGAGCCG	TTATGTTAATG
NP	19	AACTTCTGGAGGGGTGAGAA	TTATGTTAATG
NP	20	CTCTCTGATGCAAGGTTCAA	TTATGTTAATG
NP	21	GAGGACAAGAGCTCTTGTTC	TTATGTTAATG
NP	22	CTCACATGATGATCTGGCAT	TTATGTTAATG
NP	23	CTGGCGCCAAGCTAATAATG	TTATGTTAATG
NP	24	CTGGAGGACCTATATACAGG	TTATGTTAATG
NP	25	TGCGGGGAAAGATCCTAAGA	TTATGTTAATG
NP	26	AGAGAGAATGGTGCTCTCTG	TTATGTTAATG
NP	27	GGGACGGTTGATCCAAAACA	TTATGTTAATG
NP	28	TGTGCACCGAACTCAAACTC	TTATGTTAATG
NP	29	ATTGGACGATTCTACATCCA	TTATGTTAATG
NP	30	TGAAATCAGAGCATCCGTCG	TTATGTTAATG
NP	31	ACTGATGGAGAACGCCAGAA	TTATGTTAATG
NP	32	TCTCAAGGCACCAAACGATC	TTATGTTAATG
NP	33	TCACTCACTGAGTGACATCA	TTATGTTAATG
NA	1	GTCTGTTCAAAAAACTCCTT	TTTCTTCATTA
NA	2	TGCCATTCAGCATTGACAAG	TTTCTTCATTA
NA	3	TACTGTAGATTGGTCTTGGC	TTTCTTCATTA
NA	4	CTAGACTGTATGAGGCCGTG	TTTCTTCATTA
NA	5	TCAACATCCTGAGCTGACAG	TTTCTTCATTA
NA	6	TCAGGGTATAGCGGAAGTTT	TTTCTTCATTA
NA	7	GATGTTGTGGCAATGACTGA	TTTCTTCATTA
NA	8	TAGTAAGTTCTCTGTGAGGC	TTTCTTCATTA
NA	9	CCTAATGGATGGACAGAGAC	TTTCTTCATTA
NA	10	TGGGTTTGAGATGATTTGGG	TTTCTTCATTA
NA	11	GACCAAAAGTCACAGTTCCA	TTTCTTCATTA
NA	12	GGTATGGTAATGGTGTTTGG	TTTCTTCATTA
NA	13	GGAGCAAACGGAGTAAAGGG	TTTCTTCATTA

Table A.1 continued from previous page

NA	14	TAGGATACATCTGCAGTGGG	TTTCTTCATTA
NA	15	CATGGGTGTCTTTCGATCAA	TTTCTTCATTA
NA	16	AATTGGCATGGTTCGAACCG	TTTCTTCATTA
NA	17	TTACCCTGATACCGGCAAAG	TTTCTTCATTA
NA	18	CTCACTATGAGGAATGTTCC	TTTCTTCATTA
NA	19	CGATAGAGTTGAATGCACCT	TTTCTTCATTA
NA	20	TCGAAAAGGGGAAGGTTACT	TTTCTTCATTA
NA	21	GCTGGCCTCGTACAAAATTT	TTTCTTCATTA
NA	22	ACTATAATGACTGATGGCCC	TTTCTTCATTA
NA	23	GCCTGTGTAAATGGTTCATG	TTTCTTCATTA
NA	24	TTGAGGACACAAGAGTCTGA	TTTCTTCATTA
NA	25	TAATGGAGCAGTGGCTGTAT	TTTCTTCATTA
NA	26	CAATCGGAATTTCAGGTCCA	TTTCTTCATTA
NA	27	TTGGTCAGCAAGTGCATGTC	TTTCTTCATTA
NA	28	TATAGGGCCTTAATGAGCTG	TTTCTTCATTA
NA	29	GTGGGACTGTTAAGGACAGA	TTTCTTCATTA
NA	30	GCCTTACTGAATGACAAGCA	TTTCTTCATTA
NA	31	GAATGCAGGACCTTTTTTCT	TTTCTTCATTA
NA	32	TTTGTCATAAGAGAGCCCTT	TTTCTTCATTA
NA	33	AATTGGTTCCAAAGGAGACG	TTTCTTCATTA
NA	34	GGTGGGCTATATACAGCAAA	TTTCTTCATTA
NA	35	ATTCATCTCTTTGTCCCATC	TTTCTTCATTA
NA	36	ACTTCAGTGATATTAACCGG	TTTCTTCATTA
NA	37	ATAGCACCTGGGTAAAGGAC	TTTCTTCATTA
NA	38	GCAACCAAAACATCATTACC	TTTCTTCATTA
NA	39	GGAAGTCAAAACCATACTGG	TTTCTTCATTA
NA	40	TCTCAATATGGATTAGCCAT	TTTCTTCATTA
NA	41	AGTCGGACTAATTAGCCTAA	TTTCTTCATTA
NA	42	CCATTGGATCAATCTGTCTG	TTTCTTCATTA
M	1	GATGGTCATTTTGTCAGCAT	TTTTAGGTAAA
M	2	AAGGAACAGCAGAGTGCTGT	TTTTAGGTAAA
M	3	TACGGAAGGAGTGCCAAAGT	TTTTAGGTAAA
M	4	CCGTCGCTTTAAATACGGAC	TTTTAGGTAAA
M	5	GCACTTGACATTGTGGATTC	TTTTAGGTAAA
M	6	TTGCCGCAAATATCATTGGG	TTTTAGGTAAA
M	7	TCAGAAACGAATGGGGGTGC	TTTTAGGTAAA
M	8	TCCAGTGCTGGTCTGAAAAA	TTTTAGGTAAA
M	9	AACCATTGGAACTCATCCTA	TTTTAGGTAAA
M	10	TCAGGCTAGACAAATGGTGC	TTTTAGGTAAA
M	11	CAAATGGCTGGATCGAGTGA	TTTTAGGTAAA
M	12	CACTACAGCTAAGGCTATGG	TTTTAGGTAAA
M	13	GGTGACAACAACCAATCCAC	TTTTAGGTAAA
M	14	GTATGTGCAACCTGTGAACA	TTTTAGGTAAA

Table A.1 continued from previous page

M	15	GTATGGGCCTCATATACAAC	TTTTAGGTAAA
M	16	ATCTCACTCAGTTATTCTGC	TTTTAGGTAAA
M	17	TTAGGATTTGTGTTCACGCT	TTTTAGGTAAA
M	18	AGACAAGACCAATCCTGTCA	TTTTAGGTAAA
M	19	GATCTTGAGGTTCTCATGGA	TTTTAGGTAAA
M	20	TGAAGATGTCTTTGCAGGGA	TTTTAGGTAAA
M	21	GGTCGAAACGTACGTACTCT	TTTTAGGTAAA
NS	1	GCTTGAAGTGGAGCAAGAGA	TTAATTGAGTA
NS	2	TTTATGCAAGCCTTACATCT	TTAATTGAGTA
NS	3	TCCACTCACTCCAAAACAGA	TTAATTGAGTA
NS	4	TCTACAGAGATTCGCTTGGA	TTAATTGAGTA
NS	5	AACACAGTTCGAGTCTCTGA	TTAATTGAGTA
NS	6	AAAATGCAGTTGGAGTCCTC	TTAATTGAGTA
NS	7	CAGGACATACTGCTGAGGAT	TTAATTGAGTA
NS	8	AAATTTCACCATTGCCTTCT	TTAATTGAGTA
NS	9	CACCGAAGAGGGAGCAATTG	TTAATTGAGTA
NS	10	CTCTAATATTGCTAAGGGCT	TTAATTGAGTA
NS	11	TCAGTGTGATTTTTGACCGG	TTAATTGAGTA
NS	12	ACCAGGCGATCATGGATAAG	TTAATTGAGTA
NS	13	AAATGTCAAGGGACTGGTCC	TTAATTGAGTA
NS	14	GCGTTACCTAACTGACATGA	TTAATTGAGTA
NS	15	GAAGAATCCGATGAGGCACT	TTAATTGAGTA
NS	16	CGTGCTGGAAAGCAGATAGT	TTAATTGAGTA
NS	17	CTTCGCCGAGATCAGAAATC	TTAATTGAGTA
NS	18	GCAGACCAAGAACTAGGTGA	TTAATTGAGTA
NS	19	TGGATCCAAACACTGTGTCA	TTAATTGAGTA

Table A.1: Barcoded sequences used in IAV DNA-PAINT. All sequences are arranged from 5' to 3' end.

A.2 SPIN-seq sequences

Experiments/construct	Strand Name	Sequence
Formation of 8-nt gapped DNA	Gap-(1-65)-G^{30}-$B^{Bio,1}$	5' CAG TCA AGA GCC TGA GGC AGC AGA TTA TCG ACG GTG GAT GTA TGG TAA TGG GAC GAA GAA TGA GG–biotin 3'
	Gap-(1-27)-$Top^{Cy3B,18}$	5' CCT CAT TCT TCG TCC CAT(Cy3B) TAC CAT ACA 3'
	Gap-(36-65)-Top	5' CGA TAA TCT GCT GCC TCA GGC TCT TGA CTG 3'
Template for formation of 8-nt gapped DNA with T at position 3 of the gap	Gap-(1-65)-T^{30}-$B^{Bio,1}$	5' CAG TCA AGA GCC TGA GGC AGC AGA TTA TCG ACG GTT GAT GTA TGG TAA TGG GAC GAA GAA TGA GG–biotin 3'
Template for formation of 8-nt gapped DNA with A at position 3 of the gap	Gap-(1-65)-A^{30}-$B^{Bio,1}$	5' CAG TCA AGA GCC TGA GGC AGC AGA TTA TCG ACG GTA GAT GTA TGG TAA TGG GAC GAA GAA TGA GG–biotin 3'
Template for formation of 8-nt gapped DNA with C at position 3 of the gap	Gap-(1-65)-C^{30}-$B^{Bio,1}$	5' CAG TCA AGA GCC TGA GGC AGC AGA TTA TCG ACG GTC GAT GTA TGG TAA TGG GAC GAA GAA TGA GG–biotin 3'
8 nt fluorescent seals	S8-C^3	5'Atto647N-TCC ACC GT 3'
	S8-G^3	5' Atto647N-TCG ACC GT 3'
	S8-A^3	5' Atto647N-TCA ACC GT 3'
	S8-T^3	5' Atto647N-TCT ACC GT 3'
8-nt R-seal for one-base reading	R8-N^3	5' Atto647N- TCN ACC GT 3'
Set of 4 8-nt U-seals for one-base reading	U8-A^3	5' TCA ACC GT 3'
	U8-T^3	5' TCT ACC GT 3'
	U8-G^3	5' TCG ACC GT 3'
	U8-C^3	5' TCC ACC GT 3'
Formation of 13-nt gap gapped DNA	Gap-(1-65)-$B^{Bio,1}$	5' CAG TCA AGA GCC TGA GGC AGC AGA TTA TCG ACG GTG GAT GTA TGG TAA TGG GAC GAA GAA TGA GG–biotin 3'
	Gap-(1-27)-$Top^{Cy3B,25}$	5' CCT CAT TCT TCG TCC CAT(Cy3B) TAC CAT ACA 3'
	Gap-(41-65)-Top	5' A TCT GCT GCC TCA GGC TCT TGA CTG 3'

Table A.2 continued from previous page

13-nt R-seal for 3-base reading	R13-$N^5N^5N^7$	5' - Atto647N -TCC ANN NTC GAT A- BHQ1 - 3'
Set of 12 13-nt U-seals for 3-base reading	U13-$A^5N^6N^7$	5' TCC AAN NTC GAT A 3'
	U13-$T^5N^5N^7$	5' TCC ATN NTC GAT A 3'
	U13-$G^5N^6N^7$	5' TCC AGN NTC GAT A 3'
	U13-$C^5N^6N^7$	5' TCC ACN NTC GAT A 3'
	U13-$N^5A^6N^7$	5' TCC ANA NTC GAT A 3'
	U13-$N^5T^6N^7$	5' TCC ANT NTC GAT A 3'
	U13-$N^5G^6N^7$	5' TCC ANG NTC GAT A 3'
	U13-$N^5C^6N^7$	5' TCC ANC NTC GAT A 3'
	U13-$N^5N^6A^7$	5' TCC ANN ATC GAT A 3'
	U13-$N^5N^6T^7$	5' TCC ANN TTC GAT A 3'
	U13-$N^5N^6G^7$	5' TCC ANN GTC GAT A 3'
	U13-$N^5N^6C^7$	5' TCC ANN CTC GAT A 3'
13-nt R-seal for 5-base reading	R13-$N^5N^6N^7N^8N^9$	5'- Atto647N -TCC ANN NNN GAT A- BHQ1 - 3'
Set of 20 13-nt U-seals for 5-base reading	U13-$A^5N^6N^7N^8N^9$	5' TCC AAN NNN GAT A 3'
	U13-$T^5N^6N^7N^8N^9$	5' TCC ATN NNN GAT A 3'
	U13-$G^5N^6N^7N^8N^9$	5' TCC AGN NNN GAT A 3'
	U13-$C^5N^6N^7N^8N^9$	5' TCC ACN NNN GAT A 3'
	U13-$N^5A^6N^7N^8N^9$	5' TCC ANA NNN GAT A 3'
	U13-$N^5T^6N^7N^8N^9$	5' TCC ANT NNN GAT A 3'
	U13-$N^5G^6N^7N^8N^9$	5' TCC ANG NNN GAT A 3'
	U13-$N^5C^6N^7N^8N^9$	5' TCC ANC NNN GAT A 3'
	U13-$N^5N^6A^7N^8N^9$	5' TCC ANN ANN GAT A 3'
	U13-$N^5N^6T^7N^8N^9$	5' TCC ANN TNN GAT A 3'
	U13-$N^5N^6G^7N^8N^9$	5' TCC ANN GNN GAT A 3'
	U13-$N^5N^6C^7N^8N^9$	5' TCC ANN CNN GAT A 3'
	U13-$N^5N^6N^7A^8N^9$	5' TCC ANN NAN GAT A 3'
	U13-$N^5N^6N^7T^8N^9$	5' TCC ANN NTN GAT A 3'
	U13-$N^5N^6N^7N^8G^9$	5' TCC ANN NNG GAT A 3'
	U13-$N^5N^6N^7N^8C^9$	5' TCC ANN NNC GAT A 3'
DNAs for annealing of CAPcons dsDNA	CAPcons-(1-49)-C^{25}-$B^{Bio,49}$	5' biotin-GTG CCT AAA ATG TGA TCT AGA TCA CAT TTA TTG CGT AGA GCT CAC TGC C 3'
	CAPcons-(1-49)-G^{25}-$Top^{Cy3B,13}$	5' GGC AGT GAG CTC T(Cy3B) AC GCA ATA AAT GTG ATC TAG ATC ACA TTT TAG GCA C 3'

Table A.2 continued from previous page		
DNAs for annealing of CAPcons dsDNA G•C	CAPcons-(1-49)-G^{25}-$B^{Bio,49}$	5' biotin-GTG CCT AAA ATG TGA TCT AGA TCA GAT TTA TTG CGT AGA GCT CAC TGC C 3'
	CAPcons-(1-49)-C^{25}-$Top^{Cy3B,13}$	5' GGC AGT GAG CTC T(Cy3B) AC GCA ATA AAT CTG ATC TAG ATC ACA TTT TAG GCA C 3'
DNA for preparation of CAP variant library at position 5 of the CAP consensus	CAPcons-(1-49)-N^{25}-$B^{Bio,49}$	5' biotin-GTG CCT AAA ATG TGA TCT AGA TCA NAT TTA TTG CGT AGA GCT CAC TGC C 3'
	CAPcons-(1-20)-$Top^{Cy3B,11}$	5' GGC AGT GAG CT(Cy3B)C TAC GCA AT 3'
Flanking strands for in situ gap formation on CAPcons library	CAPcons-(30-49)-Top	5' CTA GAT CAC ATT TTA GGC AC 3'
	CAPcons-(1-20)-$Top^{Cy3B,11}$	5' GGC AGT GAG CT(Cy3B)C TAC GCA AT 3'
9-nt CAPcons R-Seal	R9-CAP-N^5	5' Cy5 - AAA TNT GAT 3'
Set of 4 9-nt CAPcons U-seals	U9-CAP-A^5	5' AAA TAT GAT 3'
	U9-CAP-T^5	5' AAA TTT GAT 3'
	U9-CAP-G^5	5' AAA TGT GAT 3'
	U9-CAP-C^5	5' AAA TCT GAT 3'
DNAs for generation of lacCONS variant promoter	lacCONS(-39/-10)-$Top^{Cy3B,-15;Bio,-39}$	5' biotin- AGG CTT GAC ACT TTA TGC TTC GGC T(Cy3B) CG TAT 3'
	lacCONS(-39/+10)-$N^{+6}N^{+7}$-B	5' TCT NNC AAT TCC AAC ACG GCT ACG AGC CGA AGC ATA AAG TG 3'
	lacCONS(+11/+49)-$B^{Atto647N,+20}$	5' GTC ATC CGA AGT CTC AGG TCA CTG GAA ATT(Atto647N) GTT ATC CGC 3'
	lacCONS(+11/+49)-Top	5' GCG GAT AAC AAT TTC CAG TGA CCT GAG ACT TCG GAT GAC 3'
Flanking strands for lacCONS gap formation around positions +6/+7	lacCONS(+11/+49)-B	5' GTC ATC CGA AGT CTC AGG TCA CTG GAA ATT GTT ATC CGC 3'
	lacCONS(-39/+2)-B	5' TTC CAC ACA TTA TAC GAG CCG AAG CAT AAA GTG TCA AGC CT 3'

Table A.2 continued from previous page		
8-nt lacCONS R-seal	R8-lacCONS(+3/+9)-$N^{+7}N^{+6}$-B	5' Atto647N-TCT NNC AA 3'
Set of 8 8-nt lacCONS U-seals	U8-lacCONS(+3/+9) -A^{+7}-N^{+6}-B	5' TCT ANC AA 3'
	U8-lacCONS(+3/+9) -T^{+7}-N^{+6}-B	5' TCT TNC AA 3'
	U8-lacCONS(+3/+9) -G^{+7}-N^{+6}-B	5' TCT GNC AA 3'
	U8-lacCONS(+3/+9) -C^{+7}-N^{+6}-B	5' TCT CNC AA 3'
	U8-lacCONS(+3/+9) -N^{+7}-A^{+6}-B	5' TCT NAC AA 3'
	U8-lacCONS(+3/+9) -N^{+7}-T^{+6}-B	5' TCT NTC AA 3'
	U8-lacCONS(+3/+9) -N^{+7}-G^{+6}-B	5' TCT NGC AA 3'
	U8-lacCONS(+3/+9) -N^{+7}-C^{+6}-B	5' TCT NCC AA 3'
DNAs for preparation of lacCONS $A^{+6}T^{+7}$	lacCONS(-39/+10)-$A^{+6}T^{+7}$-$B^{Cy3B,-15;Bio,-39}$	5' Biotin-AGG CTT GAC ACT TTA TGC TTC GGC T(Cy3B)CG TAT AAT GTG TGG AAT TGA TAG A 3'
	lacCONS(-39/+10)-$A^{+6}T^{+7}$-Top	5' TCT ATC AAT TCC ACA CAT TAT ACG AGC CGA AGC ATA AAG TGT CAA GCC T 3'
DNAs for preparation of lacCONS $C^{+6}G^{+7}$ variant	lacCONS(-39/+10)-$C^{+6}G^{+7}$-$B^{Cy3B,-15;Bio,-39}$	5' Biotin - AGG CTT GAC ACT TTA TGC TTC GGC T(Cy3B) CG TAT AAT GTG TGG AAT TGC GAG A 3'
	lacCONS(-39/+10)-$C^{+6}G^{+7}$-Top	5' TCT CGC AAT TCC ACA CAT TAT ACG AGC CGA AGC ATA AAG TGT CAA GCC T 3'

Table A.2: Sequences of DNA constructs for SPIN-Seq. Our DNA nomenclature includes experiment type, sequence length, functional group with its position and nucleotide with its position. For example, in a strand named Gap(1-27)-$Top^{Cy3B,18}$, Gap refers to Gap sequencing experiments, (1-27) refers to the nucleotide length and position of the strand with respect to the 5' end, Top denotes the top strand, and the superscript (Cy3B,18) specifies a Cy3B fluorophore attached to position 18. The type of seals is denoted at the beginning by letter R (report-seal) or U (unlabelled-seal).

B
Python scripts

```
import cv2
import numpy as np
from scipy.stats import multivariate_normal
from scipy.ndimage import binary_hit_or_miss,
   distance_transform_edt, binary_fill_holes, binary_dilation
from fil_finder import FilFinder2D
import astropy.units as u
from skimage.filters import gaussian, threshold_isodata,
   butterworth
from skimage.measure import label
from skimage.morphology import convex_hull_image
from skimage.registration import phase_cross_correlation
from diplib import EuclideanSkeleton
import pandas as pd
from picasso.io import save_locs, load_locs, load_movie
from scipy.spatial import KDTree
import re
import os
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
import picasso.render as _render
from tifffile import imwrite, imread
from concurrent.futures import ThreadPoolExecutor
import multiprocessing
from pystackreg import StackReg
from pystackreg.util import to_uint16
import ruptures as rpt
from scipy.signal import savgol_filter
from sklearn.cluster import AgglomerativeClustering
from multiprocessing import shared_memory
from dbscan import DBSCAN
from sklearn.model_selection import train_test_split
from scipy.cluster.hierarchy import fcluster, linkage
from sklearn.covariance import EllipticEnvelope
```

```
from sklearn.preprocessing import StandardScaler
import pandas as pd
from picasso.gausslq import locs_from_fits_gpufit,
   fit_spots_parallel, locs_from_fits, fits_from_futures
from picasso.localize import identify_async,
   identifications_from_futures, get_spots
from picasso.aim import aim
from picasso.postprocess import link
from picasso.lib import ensure_sanity, n_futures_done
from scipy.ndimage import shift
from numpy.lib import recfunctions as rfn
import time
from mpl_toolkits.axes_grid1.inset_locator import inset_axes

try:
    import cupy as cp
    from cupyx.scipy.ndimage import shift as cupy_shift
    from picasso.gausslq import fit_spots_gpufit
except:
    pass
```

B.1 2D super-resolution image construction and skeletonisation

```
def get_probability_distribution(lpx, lpy, psize):
    width = np.ceil(0.5 * lpy / psize)
    height = np.ceil(0.5 * lpx / psize)
    x = np.arange(-1 * width, width + 1)
    y = np.arange(-1 * height, height + 1)
    xv, yv = np.meshgrid(x, y, indexing='ij')
    pos = np.dstack((yv, xv))

    cov = [[np.square((lpx / 3) / psize), 0], [0, np.square((lpy /
    3) / psize)]]
    try:
        proba = multivariate_normal.pdf(pos, cov=cov)
    except:
        proba = multivariate_normal.pdf(pos, cov=np.array(cov) +
   np.array([[10 ** (-6), 0], [0, 10 ** (-6)]]))

    return proba, int(width), int(height)


# this function checkes whether the points is a split point
# in 2d by using the hit-or-miss method
def get_split_point(skeleton):
    # slelems is the list that contains all the possible 3*3
   structuring elements
    # which are split points
    selems = list()
    selems.append(np.array([[0, 1, 0], [1, 1, 1], [0, 0, 0]]))
    selems.append(np.array([[1, 0, 1], [0, 1, 0], [1, 0, 0]]))
    selems.append(np.array([[1, 0, 1], [0, 1, 0], [0, 1, 0]]))
```

```
    selems.append(np.array([[0, 1, 0], [1, 1, 0], [0, 0, 1]]))
    selems.append(np.array([[0, 0, 1], [1, 1, 1], [0, 1, 0]]))
    selems = [np.rot90(selems[i], k=j) for i in range(5) for j in
range(4)]

    selems.append(np.array([[0, 1, 0], [1, 1, 1], [0, 1, 0]]))
    selems.append(np.array([[1, 0, 1], [0, 1, 0], [1, 0, 1]]))

    branches = np.zeros_like(skeleton, dtype=bool)
    for selem in selems:
        branches |= binary_hit_or_miss(skeleton, selem)  # |= is
in-place add
    # print(branches)
    # plt.imshow(branches)
    # plt.show()
    return branches


def construct_super_resolution_image(locs, oversampling=500, lpx
=0.1, lpy=0.1,
                                     xrange=None, yrange=None, xshift=
None, yshift=None):
    # move the localizations to the center of the image and leave
a 0.125 pixel border
    if xrange != None and yrange != None and xshift != None and
yshift != None:
        xrange, yrange = xrange + 0.25, yrange + 0.25
        locs['x'] = locs['x'] - xshift + 0.125
        locs['y'] = locs['y'] - yshift + 0.125

    else:
        xrange, yrange = np.ptp(locs[['x', 'y']], axis=0) + 0.25
        locs['x'] = locs['x'] - min(locs['x']) + 0.125
        locs['y'] = locs['y'] - min(locs['y']) + 0.125

    mat = np.zeros((int(np.ceil(yrange * oversampling)), int(np.
ceil(xrange * oversampling))))

    # get the image called mat with the localizations as Gaussian
    pos_proba, width, height = get_probability_distribution(lpx,
lpy, 1 / oversampling)
    # Compute the centers
    centers_x = (locs['x'] * oversampling).astype(int)
    centers_y = (locs['y'] * oversampling).astype(int)

    # Create a meshgrid for the positions
    x = np.arange(-width, width + 1)
    y = np.arange(-height, height + 1)
    xx, yy = np.meshgrid(x, y)

    # Iterate over all centers and add the probability
distribution
    for cx, cy in zip(centers_x, centers_y):
        mat[cy + yy, cx + xx] += pos_proba
```

```
    # apply low-pass filter and gaussian filter
    image = butterworth(mat, cutoff_frequency_ratio=0.02,
   high_pass=False, order=1)
    image = gaussian(image, oversampling * 0.01 * 5)  # sigma =
   5.85mn

    return image


def get_longest_path(locs, oversampling=500, lpx=0.1, lpy=0.1,
                     xrange=None, yrange=None, xshift=None, yshift
   =None):
    '''this function create a super-resolution image by
   considering each localization
        as a Gaussian whose sigma is lpx and lpy
        and then apply low-pass filter and gaussian filter to this
    image
        and then apply thresholding to get a binary image. If
   there are multiple objects
        in the binary image, then apply convex hull to connect
   them
        Then get the skeleton of the binary image
        If there are branches exist in the skeleton (split points
   exist),
        apply FilFinder2D to get the longest path
        oversampling is the ratio of the pixel size of the super-
   resolution
        mage to the original image (117nm)'''

    locs = locs.reset_index()  # copying the locs dataframe

    mat = construct_super_resolution_image(locs, oversampling=
   oversampling, lpx=lpx, lpy=lpy,
                                           xrange=range, yrange=
   yrange, xshift=xshift, yshift=yshift)

    # apply low-pass filter and gaussian filter
    image = butterworth(mat, cutoff_frequency_ratio=0.02,
   high_pass=False, order=1)
    image = gaussian(image, oversampling * 0.01 * 5) # sigma =
   5.85mn

    threshold = threshold_isodata(image)
    binary = image > threshold
    binary = binary_fill_holes(binary)

    _, num = label(binary, connectivity=2, return_num=True)

    if num > 1:
        binary = convex_hull_image(binary)
    else:
        pass

    axis = np.array(EuclideanSkeleton(binary, 'one neighbor'))
    distance_transform = distance_transform_edt(binary)
```

```
    branches = get_split_point(axis)
    if np.all(branches == False):
        pass
    else:
        fil = FilFinder2D(image, beamwidth=0 * u.pix)  # distance=
   u.pc,
        fil.skeleton = axis
        fil.analyze_skeletons(skel_thresh=1 * u.pix)
        axis = fil.skeleton_longpath


    points = np.transpose(np.nonzero(axis))
    if len(points) < 2:
        angle = np.nan
    else:
        vx, vy, _, _ = cv2.fitLine(np.flip(points, axis=1), cv2.
   DIST_L2, 0, 0.01, 0.01)
        angle = float(np.arctan(- vy / vx)) / np.pi * 180

    widths = distance_transform[np.nonzero(axis)]

    return axis, angle, np.median(widths)
```

B.2 Co-localisation analysis of vRNPs

```
def get_skeleton_based_params_one_complex(complex, key,
   oversampling_factor=500, original_ps=117):
    params_list = []
    centered_skeleton = []
    contained_segs = complex['segment'].unique()
    z_exist = 'z' in complex.columns

    pixel_size = original_ps / oversampling_factor


    scaler = StandardScaler(with_std=False)
    xshift, yshift = min(complex['x']), min(complex['y'])
    xrange, yrange = np.ptp(complex[['x', 'y']], axis=0)

    for seg in contained_segs:
        locs_one_seg = complex[complex['segment'] == seg]
        gp = locs_one_seg['group'].unique()[0]

        # this skeletons of all segments are drawn on the same
   canvas, thus
        # The size of canvas will affect the threshold found by
   iso_data algorithm
        # and affect the binary image, spine length etc.
        skeleton, angle, width = get_longest_path(locs_one_seg,
   xshift=xshift, oversampling=oversampling_factor,
                                                  yshift=yshift,
   xrange=xrange, yrange=yrange)
        sk = np.flip(np.transpose(np.nonzero(skeleton)), axis=1)
```

```
    # all centralized skeletons of different segments will
later be pooled together
    # to get the direction of the virus. Putting the skeleton
in the middle is because the
    # general direction is the direction of the maximum
variance.
    centered_skeleton.append(scaler.fit_transform(sk))

    params = [len(locs_one_seg), locs_one_seg['x'].mean(),
locs_one_seg['y'].mean(),
              np.sum(skeleton) * pixel_size, angle, width *
pixel_size, seg, gp]
    # convert the unit to nm (the oversampling factor is 100)


    if z_exist == True:
        params.append(np.percentile(locs_one_seg['z'], 84.3) -
 np.percentile(locs_one_seg['z'], 15.7))
        params.append(locs_one_seg['z'].mean())

    params_list.append(params)


column_names = ['locs_num', 'x', 'y', 'spine_length', 'angle',
'width', 'segment', 'group']
if z_exist == True:
    column_names.append('z_range')
    column_names.append('z')

params = pd.DataFrame(params_list, columns=column_names)
params.set_index(['segment'], inplace=True)

if len(contained_segs) > 1:
    points = np.concatenate(centered_skeleton, axis=0)

    if len(points) < 2:
        params['angle_deviation'] = np.nan

    else:
        eigvals, eigvecs = np.linalg.eigh(np.cov(points.T))
        order = eigvals.argsort()[::-1]
        eigvals, eigvecs = eigvals[order], eigvecs[:, order]
        vec = eigvecs[:, 0]  # the mainVec is the direction of
 the spine
        mainAngle = float(np.arctan(- vec[1] / (vec[0] + 10 **
(-6)))) / np.pi * 180
        # add negative sign because the y axis is flipped in
the image
        params['angle_deviation'] = np.nan
        for seg in contained_segs:
            angle_dev = params.loc[seg]['angle'] - mainAngle
            if 0 < angle_dev < 90:
                pass
            elif 90 < angle_dev < 180:
                angle_dev = 180 - angle_dev
```

```
                elif -90 < angle_dev < 0:
                    angle_dev = - angle_dev
                elif -180 < angle_dev < -90:
                    angle_dev = 180 + angle_dev

                params.loc[seg, 'angle_deviation'] = angle_dev

    elif len(contained_segs) == 1:
        params['angle_deviation'] = 0


    params['complex'] = key
    return params


# this function put all locs in one complex together
# with one extract column called 'segment'
# note the position of each segment's centre here is calculated by
    the mean of all the localizations
# and in the final csv file the centre position is the mean of
   fitted spine
# do the noise filtering and get rid of complex that have multiple
    same segments here
# Only one segments is allowed in each complex, if there are
   multiple segments in one complex
# the complex will be discarded
def combine_locs_in_one_complex(segGp_locs, center, j, noise_level
   ):
    com = center[center['complex'] == j]
    contained_segs = com['segment'].unique()
    seg_counts = [len(com[com['segment'] == seg]['group'].unique()
   ) for seg in contained_segs]

    com = com.set_index(['segment'])
    if 0 < noise_level < 0.5:
        covEnv = EllipticEnvelope(contamination=noise_level)
    elif noise_level == 0:
        pass
    else:
        raise ValueError('noise_level should be between 0 and 0.5'
   )

    if np.all(np.array(seg_counts) == 1):
        locs_list = []

        for s in contained_segs:
            g = com.at[s, 'group']
            locs = segGp_locs[s][g]

            # noise filtering
            if noise_level > 0:
                X = locs[['x', 'y']].to_numpy()
                labels = covEnv.fit_predict(X)
                locs = locs[labels == 1].copy()
```

```
            locs['segment'] = s
            locs_list.append(locs)

        one_complex = pd.concat(locs_list, ignore_index=True)

        return one_complex, j

    else:
        return None, None



def concate_save_params(complexes, csv_file_path):
    with multiprocessing.Pool() as pool:
            paramsList = pool.starmap(
   get_skeleton_based_params_one_complex,
                                      [(complexes[key], key) for
   key in complexes.keys()])

    parameters = pd.concat(paramsList)
    parameters.reset_index(inplace=True)
    parameters.rename(columns={'index': 'segment'}, inplace=True)
    parameters.sort_values(by=['complex', 'segment'], inplace=True
   )

    dtype_dict = {'locs_num': int, 'x': np.float32, 'y': np.
   float32, 'z': np.float32,
                  'spine_length': np.float32, 'segment': str, '
   group': int,
                  'angle_deviation': np.float32, 'angle': np.
   float32}

    if 'z' in list(complexes.values())[0].columns:
        dtype_dict['z_range'] = np.float32
    else:
        del dtype_dict['z']

    parameters.astype(dtype_dict)

    parameters.to_csv(csv_file_path, index=False)

    return



# the minimum_split_locs_num only matters when cluster_split is
   True only keep segments
# whose localization number is larger than minimum_split_locs_num
   in the complex
def colocalization_analysis(dir, maxDis=2, size=(400, 515),
   noise_level=0.2, pattern = '_P(.+?)_', z_to_xy_pixel=117,# get
   the segment number
                            shifts=None, cluster_split=False,
   minimum_split_locs_num=80, skip_segs=[]):
```

```
if isinstance(shifts, pd.DataFrame):
    x_min = -min(shifts.loc['shift_x', :])
    x_max = size[0] - max(shifts.loc['shift_x', :])
    y_min = -min(shifts.loc['shift_y', :])
    y_max = size[1] - max(shifts.loc['shift_y', :])

elif shifts == None:
    x_min = 0
    x_max = size[0]
    y_min = 0
    y_max = size[1]

else:
    raise ValueError('shifts should be either None or a
dataframe')


files = [dir + '/' + o for o in os.listdir(dir) if o.endswith(
'.hdf5')]


# get the centre position of each segment
segGp_locs = {} # nested dictionary {segment: {group: locs}}
center_list = []

for f in files:
    seg = re.search(pattern, os.path.basename(f).split('/')
[-1])
    seg = 'P' + seg.group(1)
    if seg not in skip_segs:
        segGp_locs[seg] = {}
        locs = pd.read_hdf(f, 'locs')

        if 'z' in locs.columns:
            locs['z'] = locs['z'] / z_to_xy_pixel  # convert z
 to pixel

        for i in locs['group'].unique():
            gp = locs[locs['group'] == i]
            segGp_locs[seg][i] = gp

            cx, cy = gp['x'].mean(), gp['y'].mean()
            if x_min < cx < x_max and y_min < cy < y_max:
                center_list.append([cx, cy, seg, i])
            else:
                continue

centers = pd.DataFrame(center_list, columns=['x', 'y', '
segment', 'group'])


# colocalization analysis
hierarchy = linkage(centers[['x', 'y']].to_numpy(), method='
single', metric='euclidean')
```

```
labels = fcluster(hierarchy, maxDis, criterion='distance')
centers['complex'] = labels

if cluster_split == False:
    complexes = {}
else:
    subset_locs_complexes_1 = {}
    subset_locs_complexes_2 = {}


with multiprocessing.Pool() as pool:
    for one_complex, id in pool.starmap(
combine_locs_in_one_complex,
                                        [(segGp_locs, centers,
 j, noise_level) for j in centers['complex'].unique()]):
        if cluster_split == False:
            if one_complex is not None:
                complexes[id] = one_complex

        else:
            if one_complex is not None:
                kept_segs = [seg if np.sum(one_complex['
segment'] == seg) > minimum_split_locs_num else None
                             for seg in one_complex['segment'
].unique()]
                one_complex = one_complex.loc[one_complex['
segment'].isin(kept_segs)]
                if len(one_complex) > 0:
                    locs_subset_1, locs_subset_2 =
train_test_split(one_complex,
                                                    test_size
=0.5, stratify=one_complex['segment'])
                    subset_locs_complexes_1[id] =
locs_subset_1
                    subset_locs_complexes_2[id] =
locs_subset_2



if cluster_split == False:
    # note the x, y and z are all in pixel
    np.save(dir + '/' + 'complexes_0.1lp.npy', complexes)
else:
    np.save(dir + '/' + 'subset_locs_complexes_1.npy',
subset_locs_complexes_1)
    np.save(dir + '/' + 'subset_locs_complexes_2.npy',
subset_locs_complexes_2)


if cluster_split == False:
    concate_save_params(complexes,  dir + '/
complexes_parameters_0.1lp.csv')
else:
    concate_save_params(subset_locs_complexes_1, dir + '/
subset_locs_complexes_1_parameters.csv')
```

```
        concate_save_params(subset_locs_complexes_2, dir + '/
    subset_locs_complexes_2_parameters.csv')


    return
```

B.3 Change point detection

```
def create_circular_masks(image_shape, yx, inner_radius, gap_width
    , outer_radius):
    yy, xx = yx
    yi, xi = np.indices(image_shape)
    dist_sq = (yy - yi) ** 2 + (xx - xi) ** 2

    # Create masks
    center_mask = dist_sq <= inner_radius ** 2
    inner_plus_gap = inner_radius + gap_width
    bg_outer_mask = dist_sq <= outer_radius ** 2
    bg_inner_mask = dist_sq <= inner_plus_gap ** 2

    # Background is the ring between outer radius and inner+gap
    radius
    bg_mask = np.logical_and(bg_outer_mask, ~bg_inner_mask)

    return center_mask, bg_mask


def calculate_intensities(movie_path, hdf5_path, inner_radius=2,
    gap_width=1,
                          roi=(0, 428, 684, 856), outer_radius=4):
    """
    the signal is calculated from the median value in the inner
    circle minus
    the median value between the outer circle and (inner circle
    plus a gap)
    """
    #locs = pd.read_hdf(hdf5_path, key='locs')
    locs = pd.read_csv(hdf5_path)
    locs = locs[locs['frame'] == 0]
    yx_coords = locs[['y', 'x']].to_numpy()
    #groups = np.unique(locs['group'])
    #yx_coords = locs.groupby(by='group').mean()[['y', 'x']].
    to_numpy()

    image_stack = imread(movie_path)
    image_stack = image_stack[:, roi[0]:roi[2], roi[1]:roi[3]]

    n_frames = image_stack.shape[0]
    n_localizations = len(yx_coords)

    signals = np.zeros((n_frames, n_localizations))
    backgrounds = np.zeros((n_frames, n_localizations))

    for i, (y, x) in enumerate(yx_coords):
```

```
        # Create masks for this localization
        center_mask, bg_mask = create_circular_masks(
            image_shape=image_stack.shape[1:],
            yx=(y, x),
            inner_radius=inner_radius,
            gap_width=gap_width,
            outer_radius=outer_radius
        )

        # Calculate intensities
        roi_pixels = image_stack[:, center_mask]
        roi_median = np.median(roi_pixels, axis=1)

        bg_values = image_stack[:, bg_mask]
        bg_median = np.median(bg_values, axis=1)
        bg_sum = bg_median

        signals[:, i] = roi_median - bg_sum
        backgrounds[:, i] = 0


    signal_df = pd.DataFrame(signals,  columns=locs.index) #
   columns=locs.index

    # # Filter : Remove constant signals (low standard deviation)
    signal_stds = signal_df.std(axis=0)
    signal_df_filtered = signal_df.loc[:, signal_stds >= 1]

    # # Save the filtered results
    save_path = movie_path.replace('.tif', '_intensity.csv')
    signal_df_filtered.to_csv(save_path, index=False)
    #
    # # Also filter the background array to match
    valid_columns = signal_df_filtered.columns
    backgrounds_filtered = backgrounds[:, [i for i, col in
   enumerate(locs.index) if col in valid_columns]]

    return signal_df_filtered, backgrounds_filtered

class time_series_bocd():
    def __init__(self, data):
        self.trace = data
        self.T = len(data)

        self.stage_parameter = None
        self.R = np.zeros((self.T + 1, self.T + 1))
        self.pmean = np.zeros(self.T)
        self.pvar = np.zeros(self.T)
        self.cps = []

    def bocd(self):
        hazard = 1 / self.T
        # More robust prior estimation
        varx = np.var(self.trace)
        mean0 = np.median(self.trace)  # Using median for
```

```
robustness

    # Estimate var0 using interquartile range (more robust
than variance)
    iqr = np.percentile(self.trace, 75) - np.percentile(self.
trace, 25)
    var0 = (iqr / 1.34) ** 2  # Convert IQR to variance
estimate

    model = GaussianUnknownMean(mean0, var0, varx)

    log_R = -np.inf * np.ones((self.T + 1, self.T + 1))
    log_R[0, 0] = 0
    pmean = np.empty(self.T)
    pvar = np.empty(self.T)
    log_message = np.array([0])
    log_H = np.log(hazard)
    log_1mH = np.log(1 - hazard)

    for t in np.arange(1, self.T + 1):
        x = self.trace[t - 1]

        # Model predictions with smoothing
        weights = np.exp(log_R[t - 1, :t])
        weights = weights / weights.sum()  # Normalize
        pmean[t - 1] = np.sum(weights * model.mean_params[:t])
        pvar[t - 1] = np.sum(weights * model.var_params[:t])

        log_pis = model.log_pred_prob(t, x)
        log_growth_probs = log_pis + log_message + log_1mH
        log_cp_prob = logsumexp(log_pis + log_message + log_H)

        new_log_joint = np.append(log_cp_prob,
log_growth_probs)
        log_R[t, :t + 1] = new_log_joint - logsumexp(
new_log_joint)

        model.update_params(t, x)
        log_message = new_log_joint

    self.R = np.exp(log_R)
    self.pmean = pmean
    self.pvar = pvar
    return


def PELT_linear_analysis(self, penalty=50, mini_size=1):

    data = self.pmean

    # PELT change point detection
    algo = rpt.KernelCPD(kernel='linear', min_size=mini_size).
fit(data)
    bkps = algo.predict(pen=penalty)
    bkps.insert(0, 0)
```

```
        starts = bkps[:-1]
        ends = bkps[1:]
        self.cps = bkps[1:-1]

        intensities = [np.median(data[start: end]) for start, end
    in zip(starts, ends)]

        stage_params = np.column_stack([intensities, starts, ends,
     range(len(starts))])
        stage_params = pd.DataFrame(stage_params, columns=['
    intensity', 'start', 'end', 'stage']
                                    ).astype({'start': int, 'end':
     int, 'stage': int})

        stage_params['class'] = 0
        #print(stage_params)
        if len(starts) > 1:
            threshold = 2 * np.median(np.sqrt(self.pvar)) + np.
    percentile(self.pmean, 15)
            stage_params['class'] = (stage_params['intensity'] >
    threshold).astype(int)
            # ------------------- merge consecutive stages in the
    same class-------------------------
            # merge consecutive stages with belong to the same
    class
            stage_params['group'] = (
                (stage_params['class'] != stage_params['class'].
    shift())
            ).cumsum()

            # Perform the aggregation to merge stages
            stage_params = stage_params.groupby('group').agg({
                'intensity': 'first',
                'start': 'min',
                'end': 'max',
                'class': 'first',
                # 'merge': 'first',
            }).reset_index(drop=True)

            # Recalculate the real mean intensity using the
    original signal
            stage_params['intensity'] = stage_params.apply(
                lambda row: np.median(data[int(row['start']):int(
    row['end'])]), axis=1)

            # stage_params.sort_values(by='start', inplace=True)
            stage_params.reset_index(drop=True, inplace=True)

        self.stage_parameter = stage_params
        #print(stage_params)
        self.based_corrected_pmean = data

        return
```

```
 def plot_posterior(self, title):
     # --- Figure (leave extra right margin so the external
colorbar isn't clipped)
     fig, (ax1, ax2) = plt.subplots(
         2, 1, figsize=(20, 10),
         gridspec_kw={'height_ratios': [3, 1]}
     )
     fig.subplots_adjust(right=0.92, hspace=0.18)

     T = int(self.T)
     t = np.arange(T)

     # ============================
     # Top plot: data + predictions
     # ============================
     ax1.plot(t, self.trace, 'b-', alpha=0.5, label='Raw data')
     ax1.set_xlim(0, T)
     ax1.margins(0)
     ax1.set_ylabel('Value')
     ax1.grid(True, linestyle='--', alpha=0.7)

     ax1.plot(t, self.pmean, 'k-', linewidth=2, label='
Predicted mean')
     _std = np.sqrt(self.pvar)
     ax1.fill_between(t, self.pmean - _std, self.pmean + _std,
                      color='gray', alpha=0.2, label='± std')

     class_colors = {0: 'green', 1: 'orange'}
     for _, row in self.stage_parameter.iterrows():
         start = int(row['start'])
         end = int(row['end'])
         class_id = int(row['class'])
         ax1.axvspan(start, end, facecolor=class_colors.get(
class_id, 'gray'), alpha=0.2)

     for cp in self.cps:
         ax1.axvline(cp, c='red', ls='--', alpha=0.7, linewidth
=1)

     ax1.legend(loc='upper right')
     ax1.set_title(title)

     # ====================================
     # Bottom plot: run-length posterior R(t)
     # ====================================
     R_max = self.R.shape[0] - 1  # max run length index
     R_rot = np.rot90(self.R)  # keep the rotation step

     im = ax2.imshow(
         R_rot,
         aspect='auto',
         cmap='gray_r',
         norm=LogNorm(vmin=1e-4, vmax=1),
         extent=[0, T, 0, R_max]  # x=time, y=run length scale
     )
```

```
        ax2.set_xlim(0, T)

        ax2.margins(0)
        ax2.set_xlabel('Time')
        ax2.set_ylabel('Run length')

        # ==============================
        # Colorbar OUTSIDE the canvas
        # (does not shrink ax2)
        # ==============================
        cax = inset_axes(
            ax2,
            width="2.2%", height="100%",
            loc='lower left',
            bbox_to_anchor=(1.01, 0., 1, 1),  # just outside the
   right edge
            bbox_transform=ax2.transAxes,
            borderpad=0
        )
        cbar = fig.colorbar(im, cax=cax)
        cbar.set_label('P(run)', labelpad=1)

        plt.show()


class GaussianUnknownMean:

    def __init__(self, mean0, var0, varx):
        """Initialize model.

        meanx is unknown; varx is known
        p(meanx) = N(mean0, var0)
        p(x) = N(meanx, varx)
        """
        self.mean0 = mean0
        self.var0 = var0
        self.varx = varx
        self.mean_params = np.array([mean0])
        self.prec_params = np.array([1 / var0])

    def log_pred_prob(self, t, x):
        """Compute predictive probabilities pi, i.e. the posterior
    predictive
        for each run length hypothesis.
        """
        # Posterior predictive: see eq. 40 in (Murphy 2007).
        post_means = self.mean_params[:t]
        post_stds = np.sqrt(self.var_params[:t])
        return norm(post_means, post_stds).logpdf(x)

    def update_params(self, t, x):
        """Upon observing a new datum x at time t, update all run
   length
        hypotheses.
        """
```

```
        # See eq. 19 in (Murphy 2007).
        new_prec_params = self.prec_params + (1 / self.varx)
        self.prec_params = np.append([1 / self.var0],
    new_prec_params)
        # See eq. 24 in (Murphy 2007).
        new_mean_params = (self.mean_params * self.prec_params
    [:-1] + \
                           (x / self.varx)) / new_prec_params
        self.mean_params = np.append([self.mean0], new_mean_params
    )

    @property
    def var_params(self):
        """Helper function for computing the posterior variance.
        """
        return 1. / self.prec_params + self.varx
```

B.4 Movie correction

```
class one_channel_movie(object):
    def __init__(self, mov, roi=None, frame_range=np.inf):
        if isinstance(mov, str):
            self.movie_path = mov
            self.movie = None
            self.info = None

        elif isinstance(mov, np.ndarray):
            self.movie = mov
            self.movie_path = None
            initial_dict = {'Byte Order': '<',
                            'Data Type': 'uint16',
                            'File': '',
                            'Frames': mov.shape[0],
                            'Height': mov.shape[1],
                            'Micro-Manager Acquisiton Comments': '
    ',
                            'Width': mov.shape[2],}
            self.info = [initial_dict]
        else:
            raise ValueError("movie must be a string (path to the
    movie file) or a numpy array")

        self.roi = roi
        self.frame_range = frame_range
        self.camera_info = {}

        self.locs = None
        #self.cluster_param = None

    def __getitem__(self, index):
        return self.movie[index]

    # this function is modified from the localize function in
    picasso.localize
```

```
 # and it exports the fitting quality as well
 def movie_format(self, baseline=400):
     # print(self.movie_path)
     # print(self.movie)
     if self.movie_path is not None and self.movie is None:
         movie, info = load_movie(self.movie_path)
         # handle the roi before feeding it into the picasso
functions
         if self.roi is not None:
             movie = movie[:, self.roi[0]: self.roi[2], self.
roi[1]: self.roi[3]]

             info[0]['Width'] = self.roi[3] - self.roi[1]
             info[0]['Height'] = self.roi[2] - self.roi[0]


         # change the frame range if it is not None
         if self.frame_range is not np.inf:
             if isinstance(self.frame_range, (list, tuple)):
                 if self.frame_range[1] is np.inf:
                     movie = movie[self.frame_range[0]:, :, :]
                 else:
                     movie = movie[self.frame_range[0]: self.
frame_range[1], :, :]
                     info[0]['Frames'] = self.frame_range[1] -
self.frame_range[0]

             elif isinstance(self.frame_range, int):
                 movie = movie[self.frame_range, :, :]
                 movie = movie[np.newaxis, :, :]
                 info[0]['Frames'] = 1


             else:
                 raise ValueError("frame_range must be a list/
tuple for a range or"
                                  " an integer for a single
frame")

     elif self.movie is not None and self.movie_path is None:
         movie = self.movie
         info = self.info
     else:
         raise ValueError("Please provide either a movie path
or a movie array")

     self.camera_info["Baseline"] = baseline
     self.camera_info["Sensitivity"] = 2.5
     self.camera_info["Gain"] = 1
     self.camera_info["qe"] = 0.82

     self.movie = movie
     self.info = info

     return
```

```
    def lq_fitting(self, GPU, gradient=400, box=5):
        if self.movie is None or self.info is None:
            self.movie_format()

        curr, futures = identify_async(self.movie, gradient, box,
roi=None)
        N = len(self.movie)
        while curr[0] < N:
            time.sleep(0.2)
        ids = identifications_from_futures(futures)
        spots = get_spots(self.movie, ids, box, self.camera_info)
        em = self.camera_info["Gain"] > 1

        if GPU:
            theta = fit_spots_gpufit(spots)
            locs = locs_from_fits_gpufit(ids, theta, box, em)
        else:
            fs = fit_spots_parallel(spots, asynch=True)
            n_tasks = len(fs)
            while n_futures_done(fs) < n_tasks:
                time.sleep(0.2)
            theta = fits_from_futures(fs)
            locs = locs_from_fits(ids, theta, box, em)

        localize_info = {
            "Generated by": "Picasso Localize",
            "ROI": None,
            "Box Size": box,
            "Min. Net Gradient": gradient,
            "Convergence Criterion": 0,
            "Max. Iterations": 0,
            "Pixelsize": 117,
            "Fit method": 'lq'}

        self.info.append(localize_info)
        self.locs = ensure_sanity(locs, self.info)

        return


    def drift_correction(self, GPU=True, drift=None, segmentation
=100,
                         intersect_d=20 / 117, roi_r=60 / 117):
        # self.lq_gpu_fitting(min_net_gradient=min_net_gradient,
box=box)
        # the movie will be set during the lq_gpu_fitting
        if drift is None and max(self.locs.frame) >= 3 *
segmentation:
            corrected_locs, new_info, drift = aim(self.locs, self.
info,
                        segmentation=segmentation, intersect_d=
intersect_d, roi_r=roi_r)
```

```
            drift = drift.view(np.recarray)
            self.locs = corrected_locs
            self.info = new_info
            # else:
            #     drift = np.zeros((len(self.locs), 2))

        if drift is not None:
            if GPU:
                # apply negative drift frame by frame using GPU
                movie_gpu = cp.asarray(self.movie)
                # print(movie_gpu.shape[0])
                corrected_movie_gpu = cp.empty_like(movie_gpu)
                for i in np.arange(movie_gpu.shape[0]):
                    aim_shift = (-drift[i][1], -drift[i][0])
                    corrected_movie_gpu[i] = cupy_shift(movie_gpu[
i], aim_shift, mode='constant', cval=0, order=1)

                corrected_movie = cp.asnumpy(corrected_movie_gpu)

            else:
                corrected_movie = np.empty_like(self.movie)
                for i in range(self.movie.shape[0]):
                    aim_shift = (-drift[i][1], -drift[i][0])
                    corrected_movie[i] = shift(self.movie[i],
aim_shift, mode='constant', cval=0, order=1)

            self.movie = corrected_movie

        return drift



    def dbscan(self, eps=0.5, min_samples=20):
        if self.locs is None:
            raise ValueError('Please run localization function
first')

        clusters = dbscan(self.locs, radius=eps, min_samples=
min_samples)

        self.locs = clusters

        self.cluster_param = pd.DataFrame(self.locs).groupby('
group').mean()[['x', 'y']]
        self.cluster_param['count'] = pd.DataFrame(self.locs).
groupby('group').size()

        return


    def link(self, r_max=1, max_dark_time=1):
        if self.locs is None:
            raise ValueError('Please run localization function
first')
```

```
        linked_locs = link(self.locs, self.info, r_max=r_max,
    max_dark_time=max_dark_time)

        self.locs = linked_locs


        return


    def find_no_neighbour_mask(self, locs, box_radius):
        points = np.column_stack((locs['x'], locs['y']))

        # Build KDTree for fast neighbor lookup
        tree = KDTree(points)

        # Find neighbors within the given radius
        indices = tree.query_ball_point(points, box_radius)

        # Keep only points with no close neighbors (excluding self
    )
        keep_mask = np.array([len(neigh) == 1 for neigh in indices
    ])

        return keep_mask


    def overlap_prevent(self, box_radius=2):
        # ----------------- remove overlapping locs frame by frame
     -----------------

        unique_frames = np.unique(self.locs['frame'])  # Get all
    unique frame IDs
        filtered_locs = []  # Store results for all frames

        for frame in unique_frames:
            # Select points in the current frame
            mask = self.locs['frame'] == frame
            frame_locs = self.locs[mask]

            if len(frame_locs) == 0:
                continue  # Skip empty frames

            keep_mask = self.find_no_neighbour_mask(frame_locs,
    box_radius)
            filtered_locs.append(frame_locs[keep_mask])

        concatenated_locs = rfn.stack_arrays(filtered_locs,
    asrecarray=True)
        self.locs = concatenated_locs.view(np.recarray)

        return

def channel_separate(movie_path):
    movie = imread(movie_path)
    w = movie.shape[2]
```

```
    channel_1 = one_channel_movie(movie[:, :, :w//2])
    channel_2 = one_channel_movie(movie[:, :, w//2:])

    return channel_1, channel_2


def prepare_two_channel_movie(movie_path, gradient_1=1000,
   drift_correction=True,
                              gradient_2=1000, box_1=5, box_2=5,
   gpu=True):

    channel_1, channel_2 = channel_separate(movie_path)
    channel_1.movie_format()
    channel_2.movie_format()

    if drift_correction:
        channel_1.lq_fitting(GPU=gpu, gradient=gradient_1, box=
   box_1)
        channel_2.lq_fitting(GPU=gpu, gradient=gradient_2, box=
   box_2)

        if len(channel_1.locs) > 0:
            channel_1.drift_correction(gpu)
        if len(channel_2.locs) > 0:
            channel_2.drift_correction(gpu)


    return channel_1, channel_2


def process_frame(frame, transform_mat, sr):
    """
    Process a single frame (CPU-bound task).
    """

    return sr.transform(frame, tmat=transform_mat)


def process_frame_chunk(frame_chunk, transform_mat, sr):
    """
    Process a chunk of frames using multithreading.
    """
    with ThreadPoolExecutor() as thread_pool:
        # Process all frames in the chunk in parallel using
   threads
        results = list(thread_pool.map(
            lambda frame: process_frame(frame, transform_mat, sr),
            frame_chunk
        ))
    return results


def stackreg_channel_alignment(mov, transform_matrix,
   num_processes=4):
```

```
    """
    Align all frames in `mov` using hybrid parallelism.
    """

    sr = StackReg(StackReg.RIGID_BODY)

    # Split the frames into chunks for multiprocessing
    num_frames = mov.shape[0]
    num_processes = num_processes or multiprocessing.cpu_count()

    # Use multiprocessing to process chunks in parallel
    if num_frames < num_processes:
        # Process all frames in a single chunk without
   multiprocessing
        aligned_mov = np.array(process_frame_chunk(mov,
   transform_matrix, sr))
    else:
        # Calculate chunk_size only when num_frames >=
   num_processes
        chunk_size = max(1, num_frames // num_processes)  # Ensure
    chunk_size is at least 1
        frame_chunks = [mov[i:i + chunk_size] for i in range(0,
   num_frames, chunk_size)]

        # Use multiprocessing as before
        chunk_results = []
        with multiprocessing.Pool(processes=num_processes) as pool
   :
            for r in pool.starmap(process_frame_chunk, [(chunk,
   transform_matrix, sr) for chunk in frame_chunks]):
                chunk_results.append(r)
        aligned_mov = np.concatenate(chunk_results, axis=0)


    return to_uint16(aligned_mov)


def img_rescale(img, gamma_high=0.5, gamma_low=2, percentile=90,
                    gamma_correction=True):
    lo = img.min()
    hi = img.max()
    if hi <= lo: hi = lo + 1e-6
    z = (img - lo) / (hi - lo)
    # gamma < 1 brightens highlights
    if gamma_correction:
        gamma_z = np.power(z, gamma_high, where=z > np.percentile(
   z, percentile),
                    out=np.power(z, gamma_low))
        z = gamma_z

    z = np.clip(z, 0, 1)
    z = z * 255

    return z.astype(np.uint8)
```

```
def contrast_enhance_fiducial(img, threshold=2000, box=7,
                              out_range=(0, 255), percentiles=(1, 99)
    , # robust min/max; set to (0,100) for true min/max
                              show=True, dtype=np.uint8, connectivity
    =8):

    I = np.asarray(img)
    if I.ndim != 2:
        raise ValueError("img must be a 2D array")

    # Ensure odd box
    if box < 1:
        raise ValueError("box must be >= 1")
    if box % 2 == 0:
        box += 1

    # Picked pixels via threshold + dilation
    seeds = I >= float(threshold)
    footprint = np.ones((box, box), dtype=bool)
    keep_mask = binary_dilation(seeds, structure=footprint)

    vals = I.astype(np.float32, copy=False)
    out_lo, out_hi = map(float, out_range)
    out = np.zeros_like(vals, dtype=np.float32)

    if not np.any(keep_mask):
        # nothing picked
        if show:
            plt.imshow(out, cmap='gray'); plt.axis('off'); plt.
    grid(False); plt.show()
        return out.astype(dtype) if dtype is not None else out


    # label components (boxes)
    structure = np.ones((3,3), bool) if connectivity == 8 else np.
    array([[0,1,0],[1,1,1],[0,1,0]], bool)
    labels, n = label(keep_mask, structure=structure)

    # Compute robust per-label lows/highs via bincount on ranks
    # Simpler: loop (fast enough for typical #boxes)
    for k in range(1, n+1):
        m = labels == k
        pv = vals[m]
        lo, hi = np.percentile(pv, percentiles)
        den = max(hi - lo, 1e-12)
        s = (vals[m] - lo) / den
        s = np.clip(s, 0.0, 1.0)
        out[m] = s * (out_hi - out_lo) + out_lo

    # Cast if requested
    if dtype is not None:
        if np.issubdtype(np.dtype(dtype), np.integer):
            out = np.clip(out, out_lo, out_hi)
```

```
        out = out.astype(dtype)

    if show:
        plt.imshow(out, cmap='gray'); plt.axis('off'); plt.grid(
    False); plt.show()

    return out


def align_red_green(movie_path,  gpu):
    channel_1, channel_2 = prepare_two_channel_movie(movie_path,
    gpu=gpu, drift_correction=True)

    image_1 = contrast_enhance_fiducial(channel_1.movie[0, :, :],
    threshold=50000)
    image_2 = contrast_enhance_fiducial(channel_2.movie[0, :, :],
    threshold=1000)

    sr = StackReg(StackReg.RIGID_BODY)

    rg_transform_mat = sr.register(image_2, image_1)
    aligned_channel_1_movie = stackreg_channel_alignment(mov=
    channel_1.movie, transform_matrix=rg_transform_mat)
    aligned_ref = np.concatenate((aligned_channel_1_movie,
    channel_2.movie), axis=2)

    return aligned_ref, rg_transform_mat, image_2


def two_step_channel_align(movie_path, ref_image, rg_transform_mat
    , gpu, alignment_source, method):
    # align green to green_ref
    green, red = prepare_two_channel_movie(movie_path, gpu=gpu)

    if alignment_source == 'super-resolution':
        red_cluster = dbscan(red.locs, radius=1, min_samples=50)
        _, red_image = _render.render(red_cluster, red.info)
        red_image = img_rescale(red_image)
        if method == 'phase':
            shift, _, _ = phase_cross_correlation(ref_image,
    red_image,
                                                  overlap_ratio
    =0.95, upsample_factor=50)
            red_to_red_ref = np.array([[1, 0, -shift[1]], [0, 1, -
    shift[0]], [0, 0, 1]]).astype(np.float32)

        elif method == 'real':
            sr = StackReg(StackReg.RIGID_BODY)
            red_to_red_ref = sr.register(ref_image, red_image)
        else:
            raise ValueError("method must be 'phase' or 'real'")

    elif alignment_source == 'fiducial':
        red_image = red.movie[0, :, :]
```

```
            red_image = contrast_enhance_fiducial(red_image)
            sr = StackReg(StackReg.RIGID_BODY)
            red_to_red_ref = sr.register(ref_image, red_image)

        else:
            raise ValueError('alignment_source must be either "super-
    resolution" or "fiducial"')


        print(red_to_red_ref)
        red_aligned = stackreg_channel_alignment(mov=red.movie,
    transform_matrix= red_to_red_ref)
        green_aligned = stackreg_channel_alignment(mov=green.movie,

    transform_matrix=np.dot(red_to_red_ref, rg_transform_mat))
        aligned_movie = np.concatenate((green_aligned, red_aligned),
    axis=2)

        return aligned_movie



def position_correction(movie_path_list, locs_movie_path, gpu=True
    , method='phase'):
    ''' This function do locs_based_analysis between different
    green channels using fiducial markers.
    And the fiducial markers can be detected easily in green
    channel but not red channel.
    The locs in green and red channels of transcription movie
    should be co-localized. Thus,
    the transformation matrix between green and red channel are
    acquired from there. '''

    # align localization movie using fiducial markers
    aligned_locs_movie, rg_transform_mat, red_locs_ref =
    align_red_green(locs_movie_path, gpu=gpu)
    print(rg_transform_mat)
    imwrite(locs_movie_path.replace('.tif', '_corrected.tif'),
    aligned_locs_movie)

    # find the first movie after localization movie and align
    using fiducial markers
    # because the localization movie is only 10 frame can't create
    super-resolution image
    # from it
    fist_movie_path = min(movie_path_list, key=os.path.getmtime)
    aligned_first_mov = two_step_channel_align(fist_movie_path,
    red_locs_ref, rg_transform_mat,
                                               alignment_source='
    fiducial', gpu=gpu, method='real')
    imwrite(fist_movie_path.replace('.tif', '_corrected.tif'),
    aligned_first_mov)

    first_red = one_channel_movie(aligned_first_mov[:, :, 428:])
    first_red.movie_format()
```

```
    first_red.lq_fitting(GPU=gpu, gradient=1000)
    first_red_cluster = dbscan(first_red.locs, radius=1,
   min_samples=50)

    _, red_ref = _render.render(first_red_cluster, first_red.info)
    red_ref = img_rescale(red_ref)

    # for the rest of movies creating super-resolution images in
   red channel for registration
    movie_path_list.remove(fist_movie_path)
    for movie_path in movie_path_list:
        aligned_movie = two_step_channel_align(movie_path, red_ref
   , rg_transform_mat, method=method,
                                               alignment_source='
   super-resolution', gpu=gpu)
        imwrite(movie_path.replace('.tif', '_corrected.tif'),
   aligned_movie)

    return


def process_correction(dir_path, localization_key='localization',
   gpu=True, method='phase'):
    files = [x for x in os.listdir(dir_path) if x.endswith('.tif')
    or x.endswith('.raw')]
    ref_list = [x for x in files if localization_key in x]

    if len(ref_list) == 1:
        ref_path =os.path.join(dir_path, ref_list[0])
        files.remove(ref_list[0])
        mov_path = [os.path.join(dir_path, x) for x in files]

    elif len(ref_list) > 1:
        # when multiple reference movies detected use the first
   one recorded
        mov_path = [x for x in files if x not in ref_list]
        mov_path = [os.path.join(dir_path, x) for x in mov_path]

        ref_path_list = [os.path.join(dir_path, x) for x in
   ref_list]
        ref_path = min(ref_path_list, key=os.path.getmtime)

        ref_path_list.remove(ref_path)
        mov_path.extend(ref_path_list)

    else:
        raise ValueError('no reference file is found')

    position_correction(mov_path, ref_path, gpu=gpu, method=method
   )

    return
```

B.5 Base calling

```
def neighbour_counting(ref_points, mov_points, nuc, search_radius
    =2):
     # Extract x, y coordinates
     ref_coords = np.column_stack((ref_points['x'], ref_points['y'
    ']))
     mov_coords = np.column_stack((mov_points['x'], mov_points['y'
    ']))

     # Build KDTree for fast neighbor lookup
     tree = KDTree(mov_coords)

     # Query neighbors within the given radius
     indices = tree.query_ball_point(ref_coords, search_radius)

     # Count neighbors for each reference point
     neighbor_counts = [len(neigh) for neigh in indices]


     # Convert to DataFrame
     params = pd.DataFrame({nuc: neighbor_counts})
     params.index.name = 'ref_index'

     return params



def locs_based_analysis_preAligned(ref_path, mov_list, pattern,
    search_radius=2,
                                    mov_gradient=1000, max_frame=np
    .inf, mov_baseline=400,
                                    gpu=True, ref_roi=None,
    ref_gradient=400, roi=None, save_hdf5=False):
     if ref_path.endswith('.hdf5'):
         ref_locs, _ = load_locs(ref_path)

     elif ref_path.endswith('.tif'):
         ref = one_channel_movie(ref_path, roi=ref_roi, frame_range
    =0)
         ref.movie_format()
         ref.lq_fitting(GPU=gpu, gradient=ref_gradient, box=5)
         ref.overlap_prevent(box_radius=search_radius * 2)

         ref_locs = ref.locs
         save_locs(ref_path.replace('.tif', '.hdf5'), ref_locs, ref
    .info)

     else:
         raise ValueError('please provide the address of .hdf5 or .
    tif file')

     nuc_locs = {}
     for movie_path in mov_list:
```

```
        try:
            nuc = re.search(pattern, os.path.basename(movie_path))
    .group(1)
        except:
            warnings.warn('cannot find nucleotide for {}'.format(
    movie_path))
            continue

        if movie_path.endswith('.hdf5'):
            locs, info = load_locs(movie_path)
            locs = locs[locs.frame < max_frame]
            nuc_locs[nuc] = locs

        elif movie_path.endswith('.tif'):
            mov = one_channel_movie(movie_path, roi=roi,
    frame_range=(0, max_frame))
            mov.movie_format(baseline=mov_baseline)
            mov.lq_fitting(gpu, gradient=mov_gradient, box=5)
            nuc_locs[nuc] = mov.locs

            if save_hdf5:
                save_locs(movie_path.replace('.tif', '.hdf5'), mov
    .locs, mov.info)

        else:
            raise NotImplementedError

    # ----------------------- neighbour counting
    ----------------------
    total_params = []
    for nuc in nuc_locs.keys():
        param = neighbour_counting(ref_locs, nuc_locs[nuc], nuc,
    search_radius=search_radius)
        total_params.append(param)

    counting_params = pd.concat(total_params, axis=1)

    return counting_params



def process_analysis_Localization(dir_path, pattern, ref_path=None
    , target_format='.tif',
                                  localization_keyword='
    localization', max_frame=np.inf,
                                   search_radius=2, gradient=1000,
    gpu=True, save_hdf5=False):
    files = [x for x in os.listdir(dir_path) if x.endswith(
    target_format)]

    if ref_path is None:
        ref = [x for x in files if localization_keyword in x]
        if len(ref) != 1:
            raise ValueError("There should be one and only one
    reference file in the directory")
```

```
        else:
            files.remove(ref[0])
            ref_keyword = ref[0].replace(target_format, '')
        ref = os.path.join(dir_path, ref[0])
        mov_list = [os.path.join(dir_path, x) for x in files if
    localization_keyword not in x]

    else:
        ref = ref_path
        ref_keyword = os.path.basename(ref_path).split('.')[0]
        mov_list = [os.path.join(dir_path, x) for x in files if
    localization_keyword.lower() not in x.lower()]

    counts = locs_based_analysis_preAligned(ref, mov_list, pattern
    =pattern, search_radius=search_radius, gpu=gpu,
                                            roi=[0, 428, 684,
    856], ref_roi=[0, 0, 684, 428], max_frame=max_frame,
                                            ref_gradient=400,
    mov_gradient=gradient, save_hdf5=save_hdf5)
    counts.to_csv(dir_path + '/{}_neighbour_counting_radius{}_{}.
    csv'.format(ref_keyword, search_radius, max_frame))


    return



def model_func(t, A, K):
    return A * np.exp(K * t)

def fit_exp_nonlinear(t, y):
    A_guess = y[0]
    K_guess = -0.01  # Initial decay rate guess (negative since we
    expect decay)

    opt_parms, parm_cov = curve_fit(model_func, t, y, maxfev=1000,
    p0=(A_guess, K_guess))
    A, K = opt_parms
    return A, K


def threshold_selection(data, bin_size=10, n_decay_length=3):
    data = data[data > 0]
    counts, bin_edges = np.histogram(data, bins=np.arange(0, data.
    max() + bin_size, bin_size))
    positions = (bin_edges[1:] + bin_edges[:-1]) / 2

    # Calculate first derivative
    first_deriv = np.gradient(counts, positions)
    global_min_idx = np.argmin(first_deriv)
    global_min_x = positions[global_min_idx]

    mask = positions >= global_min_x
    # Linear Fit (Note that we have to provide the y-offset ("C")
    value!!
```

```
    A, K = fit_exp_nonlinear(positions[mask], first_deriv[mask])

    fit_y = model_func(positions[mask], A, K)

    fig, ax = plt.subplots(2, 1)
    ax[0].bar(positions, counts, width=bin_size)

    ax[1].plot(positions[mask], fit_y, '--', label='fitted
   exponential decay')
    ax[1].plot(positions, first_deriv, label='first derivative')
    ax[1].legend(loc='best')
    plt.show()

    transition_x = np.round(n_decay_length* np.abs(1/K) +
   global_min_x)
    print('decay length is {}'.format(np.abs(1/K)))
    print('the threshold is {}'.format(transition_x))

    return transition_x


def non_competitive_selection(param: pd.DataFrame, threshold:
   float):
    """
    Pick the column with the *largest* value per row (non-
   competitive),
    and compute a confidence from the gap between the best and
   runner-up.
    Rows are first filtered to keep those with at least one entry
   > threshold.
    Confidence is robustly scaled using [q10, q90] of the gap
   distribution.
    """
    # 1) Row filter: keep rows with alst least 1 entry > threshold
    mask = (param.to_numpy() > threshold).sum(axis=1) >= 1
    param_f = param.loc[mask]

    # 2) Choice: max per row
    choice = param_f.idxmax(axis=1)

    # 3) Confidence: top-two gap (best - runner_up)
    #    Use partition trick on negative values to get largest
   elements efficiently.
    vals = param_f.to_numpy()

    # For maximums, operate on -vals so the "smallest two" of -
   vals are the two largest of vals
    neg_vals = -vals
    part = np.partition(neg_vals, kth=[0, 1], axis=1)
    best = -part[:, 0]
    runner_up = -part[:, 1]
    margin = best - runner_up  # non-negative top-two gap

    # 4) Robust scaling to [0, 1] using q10-q95 of margins
    q10, q90 = np.percentile(margin, [0, 95])
```

```
    scale = max(q90 - q10, 1e-12)
    conf_arr = np.clip((margin - q10) / scale, 0, 1)

    conf = pd.Series(conf_arr, index=param_f.index)

    return choice, conf


def competitive_selection(param, threshold):
    b = np.sum(param.to_numpy() > threshold, axis=1)
    b = b >= 3
    param = param.loc[b]
    choice = param.idxmin(axis=1)
    sorted = param.to_numpy()
    sorted.sort(axis=1) #diff = sorted[:, 1] - sorted[:, 0] # diff
    = diff / np.percentile(diff, 95) # conf = np.clip(diff, 0, 1)
    vals = param.to_numpy()
    part = np.partition(vals, 1, axis=1)
    best = part[:, 0]
    runner_up = part[:, 1]
    margin = runner_up - best # top-two gap

    q5, q95 = np.percentile(margin, [0, 95])
    scale = max(q95 - q5, 1e-12)
    conf = np.clip((margin - q5) / scale, 0, 1) # # choice = param
   .idxmin(axis=1)

    return choice, conf

def base_calling(path, maximum_length, exp_type, correct_pick=None
   , threshold=None,
                 bin_width=5, display=False, save_results=False):
    param = pd.read_csv(path, index_col=0)
    #remove fiducial markers
    param = param.loc[~(param.max(axis=1) > maximum_length)]

    # ----------------- threshold selection
   ------------------------------
    if type(threshold) == int or type(threshold) == float:
        transition_point = threshold
    else:
        locs_counts = param.to_numpy().flatten()
        transition_point = threshold_selection(locs_counts,
   bin_size=bin_width)


    # ----------------- confidence VS accuracy rate plot
   -------------------
    if exp_type == 'competitive':
        choice, confidence = competitive_selection(param,
   transition_point)
    elif exp_type == 'non-competitive':
        choice, confidence = non_competitive_selection(param,
   transition_point)
    else:
```

```
        raise ValueError

    results = param.loc[choice.index].copy()
    results['calling'] = choice
    results['confidence'] = confidence
    results['calling'].replace({'A': 'T', 'C': 'G', 'T': 'A', 'G':
    'C'}, inplace=True)
    if save_results:
        results.to_csv(path.replace('.csv', '_base_calling_result.
    csv'), index=True)


    if display and isinstance(correct_pick, str):
        fig, ax = plt.subplots(2, 1)

        thresholds = []
        accuracy_rate = []
        molecule_number = []
        for t in np.arange(0, 1, 0.05):
            selected_choice = results.loc[results['confidence'] >
    t]
            if len(selected_choice) == 0:
                break
            else:
                summary = selected_choice['calling'].value_counts
    ()
                if correct_pick in summary.index:
                    rate = summary.loc[correct_pick] / summary.sum
    ()
                    accuracy_rate.append(rate)
                else:
                    accuracy_rate.append(0)

                thresholds.append(t)
                molecule_number.append(len(selected_choice))

        ax[0].plot(thresholds, accuracy_rate, '-o', label='
    accuracy rate')
        ax[1].plot(thresholds, molecule_number, '-o', label='
    molecular number')
        plt.legend(loc='best')
        plt.show()

        df = pd.DataFrame({'threshold': thresholds, 'accuracy_rate
    ': accuracy_rate,
                           'molecule_number': molecule_number})
        print(df)

    return results


def time_VS_accuracy(dir_path, correct_pick, minimum_confidence,
    exp_type,
                     display=True, threshold=None):
```

```
    files = [x for x in os.listdir(dir_path) if x.endswith('.csv')
]

    frame_num = []
    accuracy_rate = []
    molecule_num = []
    for f in files:
        num = f.split('.')[0]
        num = num.split('_')[-1]
        if num.isdigit():
            result = base_calling(os.path.join(dir_path, f), int(
num) * 0.95,
                                  exp_type=exp_type, display=
display,
                                  correct_pick=correct_pick,
threshold=threshold)
            frame_num.append(int(num))

            result = result[result['confidence'] >
minimum_confidence]
            summary = result['choice'].value_counts()

            number = summary.sum()
            molecule_num.append(number)
            accuracy_rate.append(summary.loc[correct_pick] /
number)
        else:
            warnings.warn("detected other types of csv file")
            continue

    fig, ax = plt.subplots(2, 1, )
    ax[0].plot(frame_num, accuracy_rate, 'o')
    ax[0].title.set_text('accuracy rate')
    ax[0].set_xlabel('frame')
    ax[0].set_ylabel('accuracy')

    ax[1].plot(frame_num, molecule_num, 'o')
    ax[1].title.set_text('molecular number')
    ax[1].set_xlabel('frame')
    ax[1].set_ylabel('molecular number')

    fig.tight_layout()
    plt.show()

    df = pd.DataFrame({'frame': frame_num, 'accuracy':
accuracy_rate, 'molecule_number': molecule_num})
    df.sort_values('frame', inplace=True)
    df.to_csv(dir_path + '/frame_vs_accuracy_confidence{}.csv'.
format(minimum_confidence), index=False)
    print(df)
    return
```

B.6 Fourier ring correlation analysis

```
def frc_sigmoid(q, k, q0, A, B):
    """Sigmoid function for FRC fitting"""
    return A + (B - A) / (1 + np.exp(k * (q - q0)))

def fitted_decay_resolution(spfre, mag, threshold):
    popt, pcov = curve_fit(frc_sigmoid, mag, spfre)
    fitted_mag = frc_sigmoid(spfre, *popt)
    resolution = 1 / spfre[fitted_mag < threshold][0]  # in the
   unit of pixel

    return fitted_mag, resolution

# Function to pad a single image
def pad_image(img, target_size, nx, ny):
    pad_x = (target_size - nx) // 2
    pad_y = (target_size - ny) // 2
    pad_x_end = pad_x + (target_size - nx) % 2
    pad_y_end = pad_y + (target_size - ny) % 2
    return np.pad(img, ((pad_y, pad_y_end), (pad_x, pad_x_end)),
   mode='constant', constant_values=0)

def pad_to_square_even_equal(img1, img2):
    """
    Pads the input images to make them square and even-sized
   symmetrically.
    Both images will have the same size after padding.
    """
    # Get the original dimensions
    ny1, nx1 = img1.shape
    ny2, nx2 = img2.shape

    # Determine the target size (max dimension, rounded up to the
   nearest even number)
    target_size = max(nx1, ny1, nx2, ny2)
    if target_size % 2 != 0:
        target_size += 1  # Ensure the target size is even

    # Pad both images
    padded_image1 = pad_image(img1, target_size, nx1, ny1)
    padded_image2 = pad_image(img2, target_size, nx2, ny2)

    return padded_image1, padded_image2


def compute_frc(image_1, image_2, bin_width):
    ''' Computes the Fourier Ring/Shell Correlation of two 2-D
   images '''
    # image_1 = image_1 / np.sum(image_1)
    # image_2 = image_2 / np.sum(image_2)
    f1, f2 = np.fft.fftshift(np.fft.fft2(image_1)), np.fft.
   fftshift(np.fft.fft2(image_2))


    af1f2 = np.real(f1 * np.conj(f2))
    af1_2, af2_2 = np.abs(f1) ** 2, np.abs(f2) ** 2
```

```
    # Compute distances in Fourier space
    nx, ny = af1f2.shape
    x = np.arange(-np.floor(nx / 2.0), np.ceil(nx / 2.0))
    y = np.arange(-np.floor(ny / 2.0), np.ceil(ny / 2.0))
    xx, yy = np.meshgrid(x, y)
    distances = np.sqrt(xx ** 2 + yy ** 2)

    # Bin distances and compute FRC
    max_distance = np.sqrt((nx // 2) ** 2 + (ny // 2) ** 2)
    bins = np.arange(0, max_distance + bin_width, bin_width)
    indices = np.digitize(distances.ravel(), bins)

    # Compute binned sums
    length = len(bins)
    f1f2_r = np.bincount(indices, weights=af1f2.ravel(), minlength
   =length)
    f12_r = np.bincount(indices, weights=af1_2.ravel(), minlength=
   length)
    f22_r = np.bincount(indices, weights=af2_2.ravel(), minlength=
   length)

    # Normalize by bin counts
    counts = np.bincount(indices, minlength=length)
    epsilon = 1e-10  # Avoid division by zero
    f1f2_r = f1f2_r / (counts + epsilon)
    f12_r = f12_r / (counts + epsilon)
    f22_r = f22_r / (counts + epsilon)

    # Compute FRC
    density = f1f2_r / np.sqrt(f12_r * f22_r + epsilon)


    return density[1:], (bins[:-1] + bins[1:]) / 2



def get_resolution(locs_array, columns, id, binsize=1.0, threshold
   =1/7, display=False):
    pixel_size = 117/500  # nm

    locs = pd.DataFrame(locs_array, columns=columns)
    cluster = locs[locs['group'] == id].copy()

    core_sample_ratio = cluster['core_sample'].sum() / len(cluster
   )
    x_range, y_range = np.ptp(cluster[['x', 'y']], axis=0)
    x_shift, y_shift = min(cluster['x']), min(cluster['y'])

    total_image = construct_super_resolution_image(cluster.copy(),
    xrange=x_range, yrange=y_range,
                                                   xshift=x_shift, yshift
   =y_shift)
    binary = total_image > threshold_isodata(total_image)
    binary = binary_fill_holes(binary)
```

```
    _, num = label(binary, connectivity=2, return_num=True)

    if num > 1:
        binary = convex_hull_image(binary)
    else:
        pass

    total_image_area = np.sum(binary) * pixel_size * pixel_size

    # --------- Split the localizations into two sets and create
super-resolution images
    sub1, sub2 = train_test_split(cluster, test_size=0.5)


    im1 = construct_super_res_image(sub1, xrange=x_range, yrange=
y_range,
                                    xshift=x_shift, yshift=y_shift
)

    im2 = construct_super_res_image(sub2, xrange=x_range, yrange=
y_range,
                                    xshift=x_shift, yshift=y_shift
)

    im1, im2 = pad_to_square_even_equal(im1, im2)
    image_size = im1.shape[0]


    frc, frc_bins = compute_frc(im1, im2, bin_width=binsize)
    spatial_frequency = frc_bins / (image_size / 2)  # Spatial
frequency in units of 1/pixel

    mask = spatial_frequency < 0.1  # it's not possible that the
resolution better than 2nm
    spatial_frequency = spatial_frequency[mask]
    frc = frc[mask]

    try:
        fitted_frc, resolution = fitted_decay_resolution(
spatial_frequency, frc, threshold=threshold)
        resolution = resolution * pixel_size
    except:
        resolution = np.nan
        fitted_frc = frc


    if display == True:
        fig, ax = plt.subplots()
        ax.axhline(y=threshold, color='g', linestyle='--',
                   label=f'threshold 1/7')
        ax.plot(spatial_frequency, fitted_frc, '-', label='Fitted
FRC')
        ax.plot(spatial_frequency, frc, 'o', alpha=0.5, label='
Original FRC')  # Optional: plot
```

```
        plt.xlabel('spacial frequency') # in the unite of 1/
    super_resolution_pixel
        plt.ylabel('FRC')
        plt.title('cluster:{} resolution: {}'.format(id, np.round(
    resolution)))
        plt.legend()
        plt.show()
        #plt.savefig('E:/Thesis/chapter2_super-resolution/figures/
    FRC/cluster_{}.png'.format(id), dpi=600)

    return id, resolution, len(cluster), core_sample_ratio,
    total_image_area



def locs_selection(locs, eps=0.5, mini_binding_rate=0.01):
    min_samples = int(max(locs['frame']) * mini_binding_rate)
    labels, core_sample_mask = DBSCAN(locs[['x', 'y']].to_numpy(),
    eps=eps, min_samples=min_samples)
    locs['group'] = labels
    locs['core_sample'] = core_sample_mask
    locs = locs[locs['group'] != -1]

    return locs


def get_resolution_distribution(locs, frame, minimum_split_size
    =20, display=False):
    temp_locs = locs[locs['frame'] < frame]
    # # Convert DataFrame to NumPy array for sharing
    locs_array = temp_locs.to_numpy()
    columns = temp_locs.columns

    if display == False:
        # Create shared memory and copy the array
        shm = shared_memory.SharedMemory(create=True, size=
    locs_array.nbytes)
        shared_array = np.ndarray(locs_array.shape, dtype=
    locs_array.dtype, buffer=shm.buf)
        np.copyto(shared_array, locs_array)

        # Use multiprocessing to compute resolutions
        res_params = []
        with multiprocessing.Pool() as pool:
            for results in pool.starmap(get_resolution, [(
    shared_array, columns, id) for id in
                temp_locs['group'].unique() if np.sum(temp_locs['
    group'] == id) >= minimum_split_size]):
                    res_params.append([*results])

        # Clean up shared memory
        shm.close()
        shm.unlink()
```

```
        resolution_df = pd.DataFrame(np.array(res_params).reshape
    (-1, 5),
                                     columns=['group', 'resolution
    ', 'locs_num', 'core_sample_ratio', 'total_image_area'])

        return resolution_df

    elif display == True:
        ids = [i for i in temp_locs['group'].unique() if np.sum([
    temp_locs['group'] == i]) >= minimum_split_size]
        np.random.shuffle(ids)
        for i in ids:
            get_resolution(locs_array, columns, i, display=True)


    else:
        raise ValueError('display should be either True or False')

    return
```