Electronic supplementary information Multi-resolution simulation of DNA transport through large synthetic nanostructures

Adnan Choudhary[†], Christopher Maffeo[†,‡] and Aleksei Aksimentiev[*,†,‡]

[†]*Department of Physics, University of Illinois at Urbana–Champaign, Urbana, IL 61801, USA*

[‡]*Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

[*]Email: aksiment@illinois.edu

Contents

Supplementary Methods

Multi-resolution models of DNA constructs

All CG models of double-stranded DNA (dsDNA) systems were produced using the simulation package mrDNA [1]. The package was originally developed to enable fast structural relaxation of DNA origami constructs but, more generally, it allows modeling and simulation of DNA systems at a user-defined level of coarse graining. To equilibrate our DNA constructs, we simulated them in steps of increasing resolution, beginning with a very coarse (100 or 50 basepairs per bead) model and ultimately arriving at a two-beads-per-basepair representation, which was used for translocation simulations. To model single-stranded DNA (ssDNA), we used a custom two-beads-per-nucleotide model that was developed and calibrated to reproduce results of single molecule experiments [2]. Further details are provided in the system-specific methods section below.

Coarse-grained simulations of nanopore transport

The majority of DNA translocation simulations described in this work were performed using a combination of coarse-grained (CG) molecular dynamics (MD) and continuum modeling. All of the CG models used [1, 2] were simulated using Atomic-Resolution Brownian Dynamics (ARBD), a GPU-accelerated simulation engine developed in-house which employed a Brownian dynamics integrator [3]. The timestep and diffusion constant of each CG particle used in our two-beads-per-basepair simulations of dsDNA were 40 fs and 79.06 Å^2/ns, respectively.

To incorporate the effects of externally applied electrostatic potentials, continuum modeling was performed using COMSOL. The material properties of the system were set to those of water, *i.e.*, 100 kg/m^3 density, 0.00089 Pa·s dynamic viscosity, and relative permittivity of 80. The equations solved by the *Electrostatics* module were

$$\boldsymbol{\nabla} \cdot \boldsymbol{D} = \rho_v$$

$$\boldsymbol{E} = -\boldsymbol{\nabla}V,$$

where $\boldsymbol{E}$, $\boldsymbol{D}$, and V are the electric field, displacement, and potential, and ρ_v is the space charge density. The applied potentials were introduced into the calculation as boundary conditions

enforced at the appropriate surfaces. We have found that beginning with a somewhat coarse mesh and allowing COMSOL to adaptively mesh for 2-3 iterations efficiently generates a mesh over which the solution for the potential is smoothly varying and free of mesh-induced defects.

For systems lacking axial symmetry, this is the level of continuum modeling at which we have observed COMSOL to converge to a solution. For systems possessing axial symmetry, the *Transport of Diluted Species* and *Creeping Flow* modules can also be included which solve for the concentration of ions and the velocity they impart to the fluid, respectively. This introduces screening charges that allow the accurate modeling of charged surfaces.

Results from COMSOL were used in ARBD simulations by producing a DX-format file that specifies the electrostatic potential on a grid of points representing the simulation volume. The resolution of the outputted grid can be specified in COMSOL independently in the x, y, and z directions. It should be noted that exporting a grid on a finer resolution results in proportionally larger output DX files. We find that using a resolution of 1 or 0.5 nanometer works well for the systems described here, and that increasing beyond this resolution results in DX files that may be too large to be loaded onto the GPU for simulation.

Note that for complicated geometries, exporting on a simple grid in the manner described here may result in some grid points being outside of the actual simulation volumes. At these points, COMSOL writes NaN values. By deliberately exporting 5 steps in each direction beyond the simulation volume, we generate several layers of NaN values throughout the system in the shape of the solid walls. These layers are then identified using the `find_boundaries` routine of the image processing Python module, scikit-image[4] to produce a repulsive harmonic steric potential representing the solid walls of the system. The value of the steric potential is set to zero in the simulation volume and is increased with each boundary layer n as kn^2, where $k = 100$ kcal/mol, see also Ref. [5] for a more detailed description. The resulting steric potential is then written to a DX file. Before exporting the electrostatic potential to a DX file, the NaN values are replaced with zeros.

By identifying which voxel each simulation bead is in and numerically computing the negative gradient in the potential, ARBD exerts forces on the beads throughout the simulation, which is applied with a charge reduction factor, typically 0.25 for 1M KCl [6]. While this charge reduction factor of 0.25 for 1 M KCl has been observed in both all-atom simulations [6] and experiment [7], it has also been observed that the degree of screening is dependent on the salt type, concentration

and temperature [8, 9] as well as on the pore geometry [10]. In our experience, screening values of 0.15-0.4 are representative for translocation studies, but the precise value has to be calibrated separately for each system, often by comparison to experimental measurements.

The process for generating DX files from axially symmetric COMSOL models is only slightly different. COMSOL exports the electrostatic potential solution in a 2D symmetry plane which must be rotated to specify the potential over a 3D grid. We find that exporting from COMSOL with a radial resolution of 0.25 nm and then interpolating the solution over a 3D grid with 0.5 nm resolution works well. By propagating NaN values through these calculations the steric potential can be constructed as described above.

Steric exclusion model calculations of ionic current blockades

The simulation approach described above does not explicitly account for the presence of water and ions. While this precludes computing ionic currents by tracking ion passage, Wilson *et al.* demonstrated that by taking many frames throughout a simulation and performing a continuum finite-element analysis on each one, surprisingly accurate ionic currents can be produced that are in quantitative agreement with currents predicted by all-atom MD simulation [11]. This method creates a mesh representing the simulation volume and assigns to it the appropriate conductivity values for the electrolyte solution in the presence of an analyte. With the conductivity $\sigma\left(\boldsymbol{r}\right)$ specified, the continuity equation is then solved to find the electrostatic potential $V(\boldsymbol{r})$ under the boundary conditions that correspond to the applied potential,

$$\boldsymbol{\nabla}\cdot\boldsymbol{J}(\boldsymbol{r})=0 \qquad \boldsymbol{J}=-\sigma(\boldsymbol{r})\boldsymbol{\nabla}V(\boldsymbol{r})$$
$$\implies \boldsymbol{\nabla}\cdot\left(\sigma(\boldsymbol{r})\boldsymbol{\nabla}V(\boldsymbol{r})\right)=\ 0. \tag{1}$$

For this equation, the relevant boundary conditions are Dirichlet conditions where the external potentials are applied, and Neumann conditions on all other surfaces. Finally, using the solution for the electric potential obtained from Eq. 1, we can plug $V(\boldsymbol{r})$ into the following integral which is performed over one of the Dirichlet surfaces,

$$i=\oiint \hat{\boldsymbol{n}}\cdot\boldsymbol{J}(\boldsymbol{r})\ d\boldsymbol{S}=-\oiint \hat{\boldsymbol{n}}\cdot\left(\sigma(\boldsymbol{r})\boldsymbol{\nabla}V(\boldsymbol{r})\right)\ d\boldsymbol{S}, \tag{2}$$

to yield the current flowing through the surface, i. By repeating the procedure for multiple frames of a simulation trajectory, an ionic current trace is generated.

3

To numerically solve Eq. 1, we use the finite element solver FEniCS [12, 13]. With the FEniCS `BoxMesh` command, we generate a regular tetrahedral mesh spanning the simulation volume. Vertices of the mesh located within the volume occupied by the electrolyte are initially assigned the experimental bulk conductivity value, whereas vertices located inside solid walls (such as the interior of a solid-state membrane) are set to have a conductivity of zero. To account for the presence of DNA in a particular electrolyte solution, we use functions that describe how the local conductivity changes with the distance from the DNA fragment. For dsDNA, these functions were calculated by tracking the motion of explicit ions in all-atom molecular dynamics simulations where the DNA was held linearly while an electrical bias was applied [9, 14]. For ssDNA, we use inter-bead interaction potential to arrive with a good guess of the local conductivity dependence, Supplementary Fig. 2. In our experience, an inefficient algorithm for modifying the conductivity near DNA can result in the local conductivity map calculation taking orders of magnitude longer than necessary. One effective approach is to iterate through each DNA bead and modify the local conductivity within a cutoff radius, 2.5 nm for dsDNA. At each of the affected grid points, the effect of the closest DNA bead is kept. For the simulations discussed here, a 0.5 nm mesh resolution is used, after which the appropriate surfaces are marked and assigned the above boundary conditions. Finally, we use the Generalized Minimal Residual method [15] with an algebraic multigrid preconditioner [16] to solve Eq. 1.

Because several of the systems considered here are large, performing the finite-element calculations just described can take prohibitively long times. Fortunately, translocation studies involve tight constrictions where almost all of the change in electrostatic potential occurs. This allows us to justify truncating systems to just the constrictions, and performing the analysis as described. When this is done, the Dirichlet boundary conditions must be applied to the appropriate surfaces of the truncated system.

Truncating in this manner introduces a numerical instability for the trajectory frames where DNA punctures one of the Dirichlet surfaces where the current is calculated. This instability is present if the currents computed at both Dirichlet surfaces show significant random discrepancies over time. To address this issue, we artificially enlarge the truncated system by shifting one of the Dirichlet surfaces by the cutoff radius used in modifying the local conductivity. In this way, the DNA is never allowed to puncture the integration surface used to compute the currents. In the work presented here, this slightly increases the resistance of the constriction by at most 0.8%,

and therefore proportionally decreases the predicted current. This truncation was performed in the slit system where we kept only the 400 nm long slit and the capillary systems where we kept only the 300 nm long capillaries. Each of these truncated systems was also artificially extended by 2.5 nm to eliminate instabilities.

Finally, we note that FEniCS allows computation to be parallelized across many cpu cores through the use of the `mpirun` command. This can drastically reduce the computation time from several minutes per frame to seconds. Finding the optimum number of cores to use is a matter of trial and error for each system. The currents calculated here were performed on local workstations and used between 5 and 20 cores, resulting in analysis rates between 6 and 60 seconds per frame.

System-specific methods and protocols

Here, we elaborate on additional details regarding the simulations performed in the main text.

To-scale simulation of DNA translocation through a solid-state nanopore

MrDNA equilibration. The DNA fragment was initially built in a linear conformation with an artificially shortened bond length to fit it inside the reservoir and equilibrated with only the steric confinement potential applied. The equilibration was run for 6×10^7 steps at 200 fs/step and a resolution of 25 bp/bead. We then moved on to 6×10^7 steps at 100 fs/step with 5 bp/bead, and finally to 6×10^7 steps at 40 fs/step with the two-beads-per-base-pair model shown in Fig. 1b of the main text.

DNA translocation through a micron-scale 2D slit

COMSOL modeling. When exporting the electrostatic potential from COMSOL, the grid resolution used was $1 \times 1 \times 0.5$ nm. The finer resolution was used only in the z-direction to ensure the ultrathin 3.5 nm slit was accurately represented without unnecessarily increasing the size of the DX files describing the electrostatic and steric potentials. Additionally, the geometry of the slit was artificially extended at the exit reservoir end by 2.5 nm to prevent the issues of numerical instability, as described in Methods above.

MrDNA equilibration. MrDNA was used to equilibrate the system in five steps of increasing resolution. The 5,000 bp DNA fragment was built to have a linear configuration and placed inside the entry reservoir. The molecule was equilibrated under the action of the steric potential for 10^8 steps with a timestep of 200 fs/step and a resolution of 100 bp/bead. The equilibration was continued for 10^7 steps at 200 fs/step with 50 bp/bead, 10^7 steps at 200 fs/step with 25 bp/bead, 10^7 steps at 100 fs/step with 5 bp/bead, and finally 2×10^6 steps at 40 fs/step with the two-beads-per-base-pair model represented in Fig. 1b of the main text.

SEM calculations. The geometry of the slit was artificially extended at the exit reservoir end by 2.5 nm to prevent the issues of numerical instability, as described in the Methods above.

Capture and directed motion of ssDNA in a double nanopore system

Local conductivity near ssDNA. To determine how the local conductivity is impacted in the vicinity of the P and B beads representing ssDNA, we assumed the functional form to be a simple linear ramp, from zero to bulk 4 M LiCl conductivity. To determine the beginning and end points of the linear ramp, we used as guidance the interaction potentials between same-type beads, shown in Supplementary Fig. 2a. We estimated the effective steric radius of each bead type from the CG interaction potentials by halving the distance corresponding to 5 $k_{\mathrm{B}}T$. We then added the ion radius, 0.05 nm, to obtain the beginning of the linear ramp, Supplementary Fig. 2b. Next, we identified the distances at which each interaction potential ceases to be repulsive. We halved these values and again added 0.05 nm to obtain the ending values of the linear ramp.

SEM calculations. Boundary conditions are assigned in FEniCS by using inequalities in a *CompiledSubDomain* command to label surface elements of the mesh. Once labeled, these surfaces can be assigned the desired electrostatic potential using a *DirichletBC* command. In this manner, the entire top surface in Fig. 3a of the main text was given a potential of 0, while the left and the right halves of the bottom surface were assigned individual target potentials, permitting us to describe an asymmetric bias condition, such as the one shown in Fig. 3d of the main text. Finally, FEniCS also allows surface integrals to be computed over labeled surfaces, which we used to evaluate the ionic currents flowing through each nanopore, Eq. 2.

DNA data storage readout using a nanocapillary

COMSOL modeling. As also discussed in the Methods section of the ESI, in addition to the *electrostatics* module we also used the *Transport of Diluted Species* and *Creeping Flow* modules in COMSOL to solve for the concentration of ions and the velocity they impart to the fluid. Diffusion constants appropriate for LiCl were assigned, and the nominal 4 M concentration for each ion species was enforced through *Inflow* conditions in each reservoir. This represents an important improvement to the continuum simulation because, in addition to the 400 mV bias applied across the capillary, the capillary surfaces are also given a constant surface charge of -0.01 C/m^2 as in experiment [17, 18].

MrDNA equilibration. With only the steric potential applied to the DNA construct, we used MrDNA to equilibrate the DNA construct for 3×10^8 steps with 200 fs/step and 100 bp/bead resolution. We then simulated 3×10^8 steps at 200 fs/step with 25 bp/bead, followed by 6×10^7 steps at 200 fs/step with 5 bp/bead, and finally 6×10^7 steps at 40 fs/step with the two-beads-per-base-pair model depicted in Fig. 1b of the main text. The DNA side arms were attached at the final equilibration step. To ensure our subsequent translocation simulations complete in a reasonable time, one end of the DNA construct was restrained to remain inside the tip of the capillary during the equilibration simulation. More precisely, during the four equilibration steps, one end of the DNA was held along the axis of the capillary 25 nm outside of the capillary, then 12.5 nm outside, then at 0 nm, and finally 12.5 nm inside the capillary tip such that the translocation simulation begins with DNA already captured, as seen in the first snapshot of Fig. 4d in the main text. The shift of the restraint position in the above equilibration protocol was useful because, at coarse resolutions, the bonds between the DNA beads are longer than the tip of the capillary, which may result in DNA beads being stuck inside the capillary in a physically impossible configuration under the influence of the steric confinement potential.

SEM calculations. To calculate the currents using SEM, the system's reservoirs were truncated. The conical shape of the capillary required that the points in the rectangular conductivity grid outside of the capillary were assigned zero conductivity.

Supplementary Figures
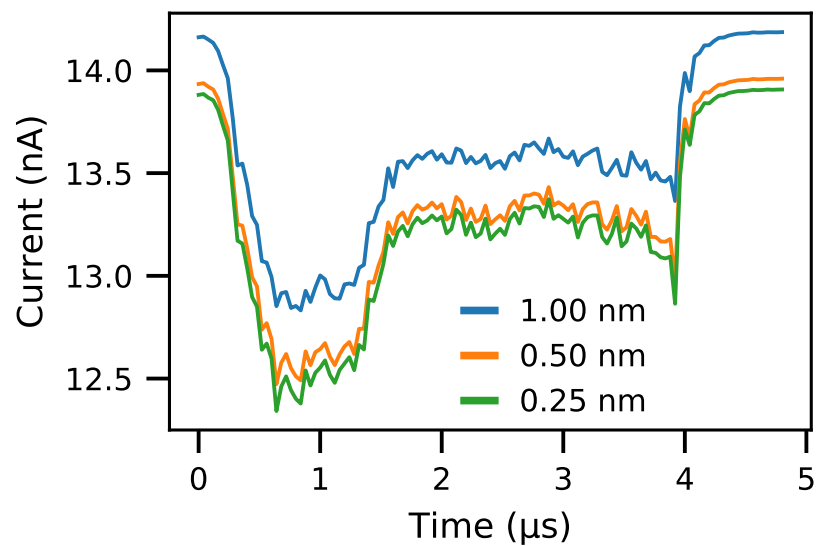
Supplementary Figure 1: Effect of grid resolution on ionic currents. The same trajectory shown in Fig. 1 of the main text is reanalyzed to produce ionic current estimates using local conductivity maps of 1, 0,5, and 0.25 nm resolution.
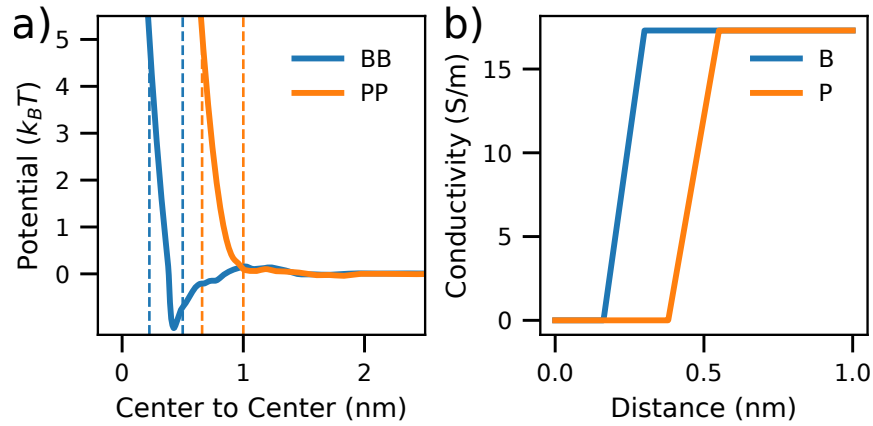
Supplementary Figure 2: Steric exclusion model of current blockades from ssDNA. (a) Interaction potentials between same-type beads for the two-beads-per-nucleotide model of ssDNA [2]. Dashed lines indicate the region over which the interaction potential ranges from strong to no repulsion. Strong repulsion is defined as 5 $k_B T$ of potential energy. (b) Local electrolyte conductivity versus distance from the center of a P or B bead used for SEM calculations of nanopore blockade currents in Fig. 3 of the main text. The bulk conductivity was set to that of 4 M LiCl. Radii designating the beginning and end of the linear ramp were obtained by halving the diameters indicated by dashed lines in panel a and adding 0.05 nm to account for the radius of an ion.

Hands-on guide to SEM analysis of simulation trajectories

The following are instructions for using the files supplied in the Supplementary file `guide.tar.gz` to perform the translocation simulations and calculate the currents shown in Figure 1 of the main document. Similar instructions are given for Figures 2-5. The files needed to perform the translocation simulations in Figures 2-5 as well as updated versions of this guide are available at https://gitlab.engr.illinois.edu/tbgl/tutorials/multiresolution-modeling-of-nanopore-transport

The software listed below is required to follow the guide:

1. COMSOL Multiphysics ($>=$ 5.3.1.275)

2. Python 3.x with following modules:

    (a) mrdna

    (b) fenics

    (c) mshr

    (d) scikit-image

    (e) scipy

    (f) numpy

    (g) gridData (bundled with MDAnalysis [19])

    (h) dill

3. ARBD

4. VMD

5. FEniCS

This protocol was developed and tested using a Linux operating system; the following may work with other operating systems, but some steps may require adaptation.

Figure 1: Solid-State Nanopore

1. Open the COMSOL model of the system, `fig1.mph`. The electrostatic potential has already been solved for you in this file, so you only need to output the solution. Do this by navigating in the "Model Builder" on the left to `Results->Export ->Data 1`. Right-click "Data" and select "Settings." In the "Filename" field, enter the path to your working directory and the filename `fig1_COMSOL.txt`. Press "Export" at the top of this window to export the electrostatic potential. Close COMSOL.

2. Run `python makeGrids.py`, which uses `fig1_COMSOL.txt` to generate the electrostatic and confinement potentials in the form of DX files. These files are outputted with the names `electric.dx` and `confine.dx`.

3. Run `python IC.py` to create, equilibrate, and start simulating the 500 bp dsDNA model. Several simulations will automatically run, one after the other. You will need ARBD and MrDNA installations for this step.

4. Descend into the output directory, `cd output00`.

5. After the translocation simulation starts, you should not allow it to run to completion. Instead, visualize the simulation using `vmd run.psf output/run.dcd -f ../confine.dx` to monitor the translocation. When it is complete, terminate the simulation and move on to the next steps. You will need a VMD installation for this step.

6. Descend into the output directory, `cd output`.

7. Once the translocation is complete, you should load the simulation as before to select the section of the trajectory you want to perform the current analysis for. Set the `first` and `last` variables in this script to . Set the value of `step` in this script to 1, 10, or 100 to reproduce the sampling frequencies in Figure 1f. We recommend beginning with a value of 1000 first.

8. Run `vmd -dispdev text < dnaloc.tcl` to output the DNA bead locations to file.

9. After installing the FEniCS package, run `python sem_analysis.py` to perform the current analysis. Match the value of step in this script to the value used in `dnaloc.tcl`. If you

want to use multiple cores for the current analysis, say 15, run `mpirun -n 15 python sem_analysis.py`

Figure 2: Ultrathin Slit

1. Open the COMSOL model of the system, `fig2.mph`. The electrostatic potential has already been solved for you in this file, so you only need to output the solution. Do this by navigating in the "Model Builder" on the left to `Results->Export ->Data 1`. Right-click "Data" and select "Settings." In the "Filename" field, enter the path to your working directory and the filename `fig2_COMSOL.txt`. Press "Export" at the top of this window to export the electrostatic potential. Close COMSOL.

2. Run `python makeGrids.py`, which uses `fig2_COMSOL.txt` to generate the electrostatic and confinement potentials in the form of DX files. These files are outputted with the names `electric.dx` and `confine.dx`.

3. Run `python IC.py` to create, equilibrate, and start simulating the 5000 bp dsDNA model. Several simulations will automatically run, one after the other. You will need ARBD and MrDNA installations for this step.

4. Descend into the output directory, `cd output00`.

5. After the translocation simulation starts, you should not allow it to run to completion. Instead, visualize the simulation using `vmd run.psf output/run.dcd -f ../confine.dx` to monitor the translocation. When it is complete, terminate the simulation and move on to the next steps. You will need a VMD installation for this step.

6. Descend into the output directory, `cd output`.

7. Run `vmd -dispdev text < dnaloc.tcl` to output the DNA bead locations to file.

8. After installing the FEniCS package, run `python sem_analysis.py` to perform the current analysis. If you want to use multiple cores for the current analysis, say 15, run `mpirun -n 15 python sem_analysis.py`

Figure 3: Double Nanopore

1. Using an installation of VMD, run `vmd -dispdev text < dnaloc.tcl` to output the DNA bead locations to file. Repeat with `dnaloc.miss.tcl` and `dnaloc.asy.tcl`.

2. After installing the FEniCS package, run `python sem_analysis.py` to perform the current analysis. If you want to use multiple cores for the current analysis, say 15, run `mpirun -n 15 python sem_analysis.py`. Repeat with `sem_analysis.miss.py` and `sem_analysis.asy.py`.

Figure 4: DNA Data Nanostructures

1. Open the COMSOL model of the system, `fig4.mph`. The electrostatic potential has already been solved for you in this file, so you only need to output the solution. Do this by navigating in the "Model Builder" on the left to `Results->Export ->Data 1`. Right-click "Data" and select "Settings." In the "Filename" field, enter the path to your working directory and the filename `fig4_COMSOL.txt`. Press "Export" at the top of this window to export the electrostatic potential used to translocate the DNA nanostructure into the capillary.

2. For the electrostatic potential used to translocate the DNA nanostructure out of the capillary, repeat the same process with the COMSOL model `fig4.r.mph`, and export the potential with the filename `fig4.r_COMSOL.txt`. Close COMSOL.

3. Run `python makeGrids.py`, which uses `fig4_COMSOL.txt` and `fig4.r_COMSOL.txt` to generate the electrostatic and confinement potentials in the form of DX files. These files are outputted with the names `electric.dx`, `confine.dx` and `electric.r.dx`, `confine.r.dx`.

4. Run `python IC.py` to create and equilibrate the DNA nanostructure. You will need an MrDNA installation for this step.

5. Descend into the output directory with `cd output00`.

6. Using an installation of ARBD, run `arbd run_in.bd output/run_in &> out_in` to perform the translocation simulation. The `out_in` file contains the arbd log.

7. Do not allow the simulation to run to completion. Instead, load the simulation using `vmd run_in.psf output/run_in.dcd -f ../confine.dx` to monitor the translocation. When the entire DNA nanostructure has translocated into the capillary, terminate the simulation. You will need a VMD installation for this step.

8. Run `vmd -dispdev text < writeRestart.out.tcl` to create the initial condition for the simulation that translocates the DNA nanostructure out of the capillary.

9. Run `arbd run_out.bd output/run_out &> out_out` to perform the translocation simulation. The `out_out` file contains the arbd log.

10. Do not allow the simulation to run to completion. Instead, load the simulation using `vmd run_out.psf output/run_out.dcd -f ../confine.r.dx` to monitor the translocation. When the entire DNA nanostructure has translocated out of the capillary, terminate the simulation.

11. Descend into the output directory, `cd output`.

12. Run `vmd -dispdev text < dnaloc.in.tcl` and `vmd -dispdev text < dnaloc.out.tcl` to output the DNA bead locations to file.

13. After installing the FEniCS package, run `python sem_analysis.in.py` to perform the current analysis. If you want to use multiple cores for the current analysis, say 15, run `mpirun -n 15 python sem_analysis.in.py`. Then do the same for `sem_analysis.out.py`.

Figure 5: DNA Origami Plate

1. Open the COMSOL model of the system, `fig5.mph`. The electrostatic potential has already been solved for you in this file, so you only need to output the solution. Do this by navigating in the "Model Builder" on the left to `Results->Export ->Data 1`. Right-click "Data" and select "Settings." In the "Filename" field, enter the path to your working directory and the filename `fig5_COMSOL.txt`. Press "Export" at the top of this window to export the electrostatic potential. Close COMSOL.

2. Run `python makeGrids.py`, which uses `fig5_COMSOL.txt` to generate the electrostatic and confinement potential in the form of a DX file. This file is outputted with the name `conf_elec.dx`.

3. Run `python IC.py` to create and equilibrate the DNA origami plate. You will need an MrDNA installation for this step.

4. Descend into the output directory with `cd output00`.

5. Run `vmd -dispdev text < writeIC.tcl`, which uses the last frame of the equilibration simulation to generate a random configuration for the DNA plate. The file corresponding to this configuration is outputted with the name `start.txt`.

6. Using an installation of ARBD, run `arbd run.bd output/run &> out_run` to perform the translocation simulation. The `out_run` file contains the arbd log.

7. Do not allow the simulation to run to completion. Instead, load the simulation using `vmd run.psf output/run.dcd -f ../conf_elec.dx` to monitor the translocation. When the entire DNA origami plate has translocated out of the capillary, terminate the simulation. You will need a VMD installation for this step.

8. Descend into the output directory, `cd output`.

9. Run `vmd -dispdev text < dnaloc.tcl` to output the DNA bead locations to file.

10. After installing the FEniCS package, run `python sem_analysis.py` to perform the current analysis corresponding to both high and low salt conditions. If you want to use multiple cores for the current analysis, say 15, run `mpirun -n 15 python sem_analysis.py`

References

[1] Christopher Maffeo and Aleksei Aksimentiev. MrDNA: A multi-resolution model for predicting the structure and dynamics of DNA systems. *Nucleic Acids Res.*, 48(9):5135–5146, 2020.

[2] Christopher Maffeo, Thuy T. M. Ngo, Taekjip Ha, and Aleksei Aksimentiev. A coarse-grained model of unstructured single-stranded DNA derived from atomistic simulation and single-molecule experiment. *J. Chem. Theory Comput.*, 10:2891–2896, 2014.

[3] Jeffrey Comer and Aleksei Aksimentiev. Predicting the DNA sequence dependence of nanopore ion current using atomic-resolution Brownian dynamics. *J. Phys. Chem. C*, 116(5):3376–3393, 2012.

[4] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: Image processing in Python. *PeerJ*, 2:e453, 6 2014.

[5] Adnan Choudhary, Himanshu Joshi, Han-Yi Chou, Kumar Sarthak, James Wilson, Christopher Maffeo, and Aleksei Aksimentiev. High-fidelity capture, threading, and infinite-depth sequencing of single DNA molecules with a double-nanopore system. *ACS Nano*, 14(11):15566–15576, 2020.

[6] Binquan Luan and Aleksei Aksimentiev. Electro-osmotic screening of the DNA charge in a nanopore. *Phys. Rev. E*, 78:021912, 2008.

[7] Ulrich F. Keyser, Bernard N. Koeleman, Stijn van Dorp, Diego Krapf, Ralph M. M. Smeets, Serge G. Lemay, Nynke H. Dekker, and Cees Dekker. Direct force measurements on DNA in a solid-state nanopore. *Nat. Phys.*, 2:473–477, 2006.

[8] Stefan W. Kowalczyk, David B. Wells, Aleksei Aksimentiev, and Cees Dekker. Slowing down DNA translocation through a nanopore in lithium chloride. *Nano Lett.*, 12(2):1038–1044, 2012.

[9] Maxim Belkin and Aleksei Aksimentiev. Molecular dynamics simulation of DNA capture and transport in heated nanopores. *ACS Appl. Mater. Interfaces*, 8(20):12599–12608, 2016.

[10] S. van Dorp, Ulrich F. Keyser, Nynke H. Dekker, Cees Dekker, and S. G. Lemay. Origin of the electrophoretic force on DNA in solid-state nanopores. *Nat. Phys.*, 5:347–351, 2009.

[11] James Wilson, Kumar Sarthak, Wei Si, Luyu Gao, and Aleksei Aksimentiev. Rapid and accurate determination of nanopore ionic current using a steric exclusion model. *ACS Sens.*, 4(3):634–644, 2019.

[12] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.

[13] Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E Rognes, and Garth N Wells. The FEniCS project version 1.5. *Arch. Numer. Soft.*, 3(100):9–23, 2015.

[14] Christopher Maffeo, Robert Schöpflin, Hergen Brutzer, René Stehr, Aleksei Aksimentiev, Gero Wedemann, and Ralf Seidel. DNA–DNA interactions in tight supercoils are described by a small effective charge density. *Phys. Rev. Lett.*, 105(15):158101, 2010.

[15] Youcef Saad and Martin H Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[16] John W Ruge and Klaus Stüben. Algebraic multigrid. In *Multigrid methods*, pages 73–130. SIAM, 1987.

[17] Kaikai Chen, Jinglin Kong, Jinbo Zhu, Niklas Ermann, Paul Predki, and Ulrich F. Keyser. Digital data storage using dna nanostructures and solid-state nanopores. *Nano Lett.*, 19(2):1210–1215, 2019.

[18] Kaikai Chen, Jinbo Zhu, Filip Bošković, and Ulrich F Keyser. Nanopore-based DNA hard drives for rewritable and secure data storage. *Nano Lett.*, 20(5):3754–3760, 2020.

[19] Naveen Michaud-Agrawal, Elizabeth J Denning, Thomas B Woolf, and Oliver Beckstein. Mdanalysis: a toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.*, 32(10):2319–2327, 2011.