

Supplementary Information

**3D Convolution neural network autoencoder for
predicting solvent configuration changes in
condensed phase biomass reactions**

Anjana Puliyananda, Arul Mozhi Devan Padmanathan, Samir H. Mushrif, and
Vinay Prasad*

*Department of Chemical and Materials Engineering, 9211 116 Street NW Edmonton,
Alberta T6G 1H9, Canada.*

E-mail: vprasad@ualberta.ca

Computational Methods: Molecular Trajectory Data

S1 ReSolv Method

The entropy gained by cellulose chains stabilize high-temperature structures and alter decomposition kinetics. This influence of condensed phase effects on the reaction chemistry can only be probed by accounting for explicit molecules and their corresponding entropic contributions. Accounting for these entropic contributions is crucial for developing a mechanistic understanding explaining cellulose chemistry. Therefore, a combination of quantum mechanics (QM) and molecular mechanics (MM) approaches are used. First the reaction chemistry in the gas phase is calculated using DFT methods. Further, thermodynamic integration (TI) method implementation on GROMACS 2018.7 is used to capture the correct finite temperature interaction energies. The gas phase DFT computed free energy barriers are corrected to correspond to condensed phase free energy barrier at four temperatures (100 K, 500 K, 900 K, 1200 K) for 60 residue cellobiose system.

S1.1 Gas Phase Transition State (TS) Calculations

Gas phase barriers are calculated using an isolated cellobiose molecule. All-electron Density Functional Theory (DFT) calculations were conducted using the Gaussian 09 code. The calculations were carried out employing the hybrid functional M06-2X76 in conjunction with the 6-31+G(2d,p) basis set, a recommended choice for carbohydrate chemistry. Both geometry optimization and Transition State (TS) search calculations were performed without any constraints. The Berny algorithm was employed to determine the TS. Subsequently, frequency calculations were executed to differentiate between saddle points and local minima. To verify the correct positioning of reactants and products on the potential energy surface, Intrinsic Reaction Coordinates (IRCs) were followed in both directions.

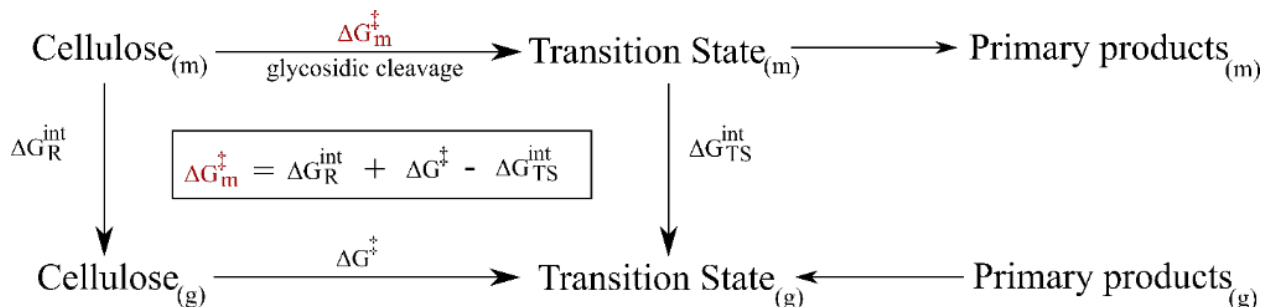


Figure S1: Schematic showing the calculation of condensed phase corrections to gas phase free energy barriers.

S1.2 Thermodynamic Integration Method

Gas phase reactant and transition state (TS) coordinates serve as the initial configurations. Langevin dynamics and OPLS/AA force-field parameters which have been benchmarked in published literature for carbohydrate molecules are utilized. The system configurations are optimized and equilibrated. Pressure equilibration is performed under NPT conditions at 1 atm for 2 ns, establishing the appropriate system volume/phase. Subsequently, TI calculations are conducted under NVT conditions for 8 ns at the relevant temperatures, with 30 λ -windows optimized for intermediate states. In all the simulation temperature control was maintained by using the Nosé-Hoover thermostat. Periodic boundary conditions were applied in all directions. Potential derivatives obtained from MD simulations for each λ -window are integrated to determine ΔG^{int} . Combining gas phase calculations and thermodynamic integration results, the free energy barrier for cellulose decomposition in the melt-phase (ΔG_m^{\ddagger}) is estimated as illustrated in Figure S1.

S2 Relative Stability of Hydronium Ion in Solvent

To investigate the effects of DMSO on the interaction of fructose and HMF with the hydronium ion, forcefield based MD and WT-MTD methods were used. These simulations were carried out in a canonical NVT ensemble at 298 K, employing GROMACS Version 4.6.7. Systems with 1% and 10% wt. of fructose and HMF, as previous experimental studies have

demonstrated that fructose conversion and HMF selectivity vary with fructose concentration, with DMSO concentrations ranging from 0 to 80% wt. For all systems, 3 molecules of fructose, 4 hydronium ions and 4 Cl^- are present. To assess the relative stability of hydronium ions in the immediate vicinity of fructose and HMF compared to the bulk solvent, WT-MTD simulations were conducted.

S2.1 Molecular Dynamics Simulations

Specifically, TIP3P model for water and the OPLS-AA force field for modeling DMSO, fructose, and HMF molecules is used. The OPLS-AA force field has been extensively validated for liquid phase simulations and has successfully captured the strong hydrogen bonding interactions between the hydroxyl groups of carbohydrates and DMSO’s sulfonyl group. To maintain a constant temperature during the simulations, the Nosé-Hoover thermostat with coupling time constants of 1 ps. Each simulation consisted of 0.1 ns of steepest descent energy minimization to reduce the energy gradient below 10 kJ/mol nm, followed by 0.4 ns of equilibration runs, 0.4 ns of isothermal-isobaric runs, and finally, 12 ns of production runs with a single MD time step of 1 fs. All simulations were conducted under periodic boundary conditions in all directions, with a cutoff of 0.7 nm using Particle-Mesh Ewald (PME) electrostatics. The MD trajectories were analyzed using Visual Molecular Dynamics (VMD), version 1.9.3.

S2.2 Well-tempered Metadynamics (WT-MTD)

WT-MTD simulations were coupled with the 12 ns MD simulations and performed in the NVT ensemble at 298 K using GROMACS-PLUMED/2016.3. Before initiating the WT-MTD runs, the system was equilibrated using MD. The WT-MTD production runs were carried out in the NVT ensemble for 300 ns, using a time step of 1 fs. Bias potentials were deposited after every 500 steps with an initial height of 1.2 kJ/mol and a bias factor used to reduce the hill height of 25. The sigma values, representing the width of the deposited

potentials, were set to 0.05 for both collective variables. Importantly, as the simulation time increased, the hill height decreased, providing adaptive adjustments to the Gaussian size. WT-MTD involves the introduction of bias potentials as a sum of Gaussians deposited along the system trajectory as a function of chosen collective variables (CVs). In this study, the coordination number between fructose/HMF molecules and hydronium ions serves as the first collective variable (CV1), and the number of water and DMSO molecules in this solvation shell, making it the second collective variable (CV2). The potential deposited during the WT-MTD simulation can be retraced, allowing for the reconstruction of the free energy surface (FES) as a function of collective variables. The first minimum in the FES corresponds to the hydronium ion in the bulk solvent, while the second minimum corresponds to the hydronium ion present in the first solvation shell of fructose. To quantify the relative stability of the hydronium ion in the first solvation shell of fructose/HMF, the difference in free energies between these two minima are calculated. A lower value of this ΔG (the free energy of the hydronium ion in the vicinity of fructose subtracted by the free energy of the hydronium ion in the bulk solvent) indicates a higher relative stability of the hydronium ion near fructose/HMF. In other words, a more negative ΔG suggests that the hydronium ion is more stable and favorable to be found in the vicinity of fructose/HMF compared to the bulk solvent.

Architecture and Parameters: 3d CNN Autoencoder

The hyperparameters of the 3d CNN autoencoder neural network primarily include specifications of its architecture by way of types/number of layers, their corresponding kernel, stride and padding dimensions, choice of weight initialization, optimizer, learning rate, training strategies, data splitting etc. The hyperparameter specifications used in this work are provided below.

- Activation function: Rectified Linear Unit (ReLU).

- Initialization method for weights: Uniform Xavier distribution.
- Optimizer: ADAM with a learning rate of 1e-3.
- Training validation split: 80:20.
- Epochwise training: 500 epochs
- Early-stopping mindelta is 1e-5 with a patience/tolerance of 10 epochs.

Further, the hyperparameters with respect to the architecture as an output to the `model.summary()` of the PyTorch neural network has been printed herein. The dimensions of a data sample corresponding to this architecture has been uploaded as an image on the GitHub repository (`'network architecture.jpg'`).

```

Net(
(conv1): Conv3d(100, 64, kernelsize=(3, 3, 3), stride=(1, 1, 1))
(bn1): BatchNorm3d(64, eps=1e-05, momentum=0.1, affine=True, trackrunningstats=True)
(conv2): Conv3d(64, 32, kernelsize=(3, 3, 3), stride=(1, 1, 1))
(bn2): BatchNorm3d(32, eps=1e-05, momentum=0.1, affine=True, trackrunningstats=True)
(pool1): MaxPool3d(kernelsize=(2, 2, 2), stride=(2, 2, 2), padding=(0, 0, 0),
dilation=(1, 1, 1), ceilmode=False)
(conv3): Conv3d(32, 16, kernelsize=(3, 3, 3), stride=(1, 1, 1))
(bn3): BatchNorm3d(16, eps=1e-05, momentum=0.1, affine=True, trackrunningstats=True)
(conv4): Conv3d(16, 8, kernelsize=(3, 3, 3), stride=(1, 1, 1))
(bn4): BatchNorm3d(8, eps=1e-05, momentum=0.1, affine=True, trackrunningstats=True)
(pool2): MaxPool3d(kernelsize=(2, 2, 2), stride=(2, 2, 2), padding=(0, 0, 0),
dilation=(1, 1, 1), ceilmode=False)
(fc1): Linear(infeatures=64, outfeatures=32, bias=True)
(fc2): Linear(infeatures=32, outfeatures=16, bias=True)
(fc3): Linear(infeatures=16, outfeatures=32, bias=True)

```

```
(fc4): Linear(infeatures=32, outfeatures=64, bias=True)
(unpool1): MaxUnpool3d(kernelsize=(2, 2, 2), stride=(2, 2, 2), padding=(0, 0,
0))
(convT1): ConvTranspose3d(8, 16, kernelsize=(3, 3, 3), stride=(1, 1, 1))
(bn5): BatchNorm3d(16, eps=1e-05, momentum=0.1, affine=True, trackrunningstats=True)
(convT2): ConvTranspose3d(16, 32, kernelsize=(3, 3, 3), stride=(1, 1, 1))
(bn6): BatchNorm3d(32, eps=1e-05, momentum=0.1, affine=True, trackrunningstats=True)
(unpool2): MaxUnpool3d(kernelsize=(2, 2, 2), stride=(2, 2, 2), padding=(0, 0,
0))
(convT3): ConvTranspose3d(32, 64, kernelsize=(3, 3, 3), stride=(1, 1, 1))
(bn7): BatchNorm3d(64, eps=1e-05, momentum=0.1, affine=True, trackrunningstats=True)
(convT4): ConvTranspose3d(64, 100, kernelsize=(3, 3, 3), stride=(1, 1, 1))
(bn8): BatchNorm3d(100, eps=1e-05, momentum=0.1, affine=True,
trackrunningstats=True)
)
```