Database of DNA electron density training structures

Alex J. Lee, Joshua R. Rackers, and William P. Bricker

September 12, 2022

This database contains the data for the DNA structures used to train the e3nn machine learning electron density model in the publication, "Predicting accurate *ab initio* DNA electron densities with equivariant neural networks". The data is contained in python pickle files containing 400 samples.

File name	**Base sequence**
`01-at-400-train.pkl`	AT
`02-ta-400-train.pkl`	TA
`03-aa-400-train.pkl`	AA
`04-ca-400-train.pkl`	CA
`05-gt-400-train.pkl`	GT
`06-ct-400-train.pkl`	CT
`07-ga-400-train.pkl`	GA
`08-cg-400-train.pkl`	CG
`09-gc-400-train.pkl`	GC
`10-gg-400-train.pkl`	GG

The electron densities are represented in the def2-universal-JFIT atom-centered auxiliary basis set. In the machine learning model, data is pre-processed by subtracting out isolated atom densities using the function `get_iso_permuted_dataset` in `utils.py`. The auxiliary basis set, the isolated atom densities, and example scripts for running the model can be found in the public Github repository: JoshRackers/**equivariant_electron_density** .

Contents of the pickle files are Python dictionaries for each sample in the following format:

Key	**Description**	**Size**	**Units**
'type'	Atomic numbers	$[n_{atom}]$	N/A
'pos'	Nuclear positions	$[n_{atom}, 3]$	Å
'onehot'	One-hot encoding for 5 atom types	$[n_{atom}, 5]$	N/A
'coefficients'	Coefficients of the electron density in the auxiliary basis	$[n_{atom}, 77]$	a.u.
'exponents'	Exponents of the auxiliary basis set	$[n_{atom}, 77]$	a.u.
'norms'	Normalization constants of the auxiliary basis set	$[n_{atom}, 77]$	a.u.
'rs_max'	Maximum number of basis functions for each order, ℓ, in the format (multiplicity, ℓ)	[(14, 0), (5, 1), (5, 2), (2, 3), (1, 4)]	N/A
'energy'	DFT total energy	$[n_{atom}]$	Hartrees
'forces'	DFT energy gradient force	$[n_{atom}, 3]$	Hartrees/Bohr

Also included is an example of a trained model in `'example-trained-model.pkl'` that can be loaded using PyTorch's `torch.load` and `load_state_dict` functions. This is the trained model that was used to obtain data in the paper. The hyperparameters for this model are reported in the Table S1 in the Supplementary Material of the paper.

Funding Statement

We thank LDRD ACORN under OSP number A21-0245 and the Sandia Academic Alliance for supporting this work. We thank the UNM Center for Advanced Research Computing, supported in part by the National Science Foundation, for providing the high performance computing and large-scale storage resources used in this work. Additional high performance computing resources, including the GPU-based machine for training and running the machine learning model, were provided by Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.