

advances.sciencemag.org/cgi/content/full/6/43/eabc6216/DC1

Supplementary Materials for

Targeted sequence design within the coarse-grained polymer genome

Michael A. Webb, Nicholas E. Jackson, Phwey S. Gil, Juan J. de Pablo*

*Corresponding author. Email: depablo@uchicago.edu

Published 21 October 2020, *Sci. Adv.* **6**, eabc6216 (2020)
DOI: 10.1126/sciadv.abc6216

The PDF file includes:

Sections S1 to S3
Tables S1 to S4
Fig. S1

Other Supplementary Material for this manuscript includes the following:

(available at advances.sciencemag.org/cgi/content/full/6/43/eabc6216/DC1)

Data files S1 to S5

S1 Force Field Parameters

This section provides supplementary information to the main text regarding the specific force-field parameters used to run the coarse-grained polymer simulations.

Table S1: Non-bonded potential parameters.

i	m_i/m	σ_{ii}/σ	$\varepsilon_{ii}/\varepsilon$
α	1.0	1.0	0.5
β	1.0	1.0	0.2
γ	1.0	1.0	0.1
δ	1.0	1.0	0.4

Table S2: Bond-stretching parameters.

$i - j$	$K_{ij} \frac{\sigma^2}{\varepsilon}$	$R_{ij}^{(0)}/\sigma$
$\beta - X$	30.0	1.8
else	30.0	1.5

Table S3: Angle-bending parameters.

$i - j - k$	$K_{ijk} \frac{\text{rad}^2}{\varepsilon}$	$\theta_{ijk}^{(0)}$
$\alpha - \alpha - (\alpha, \beta)$	30.0	165°
$\alpha - \beta - \alpha$	30.0	165°
$\beta - \beta - (\alpha, \beta)$	90.0	170°
$\beta - \alpha - \beta$	90.0	170°
$(\alpha, \beta) - (\alpha, \beta) - (\gamma, \delta)$	90.0	110°
else	30.0	165°

Table S4: Dihedral torsion parameters.

$i - j - k - l$	K_{ijkl}/ε
$(\alpha, \beta, \gamma, \delta) - \beta - \beta - (\alpha, \beta, \gamma, \delta)$	2.0
else	1.0

S2 Comparison of OHE versus Property-coloring for Class I Polymers

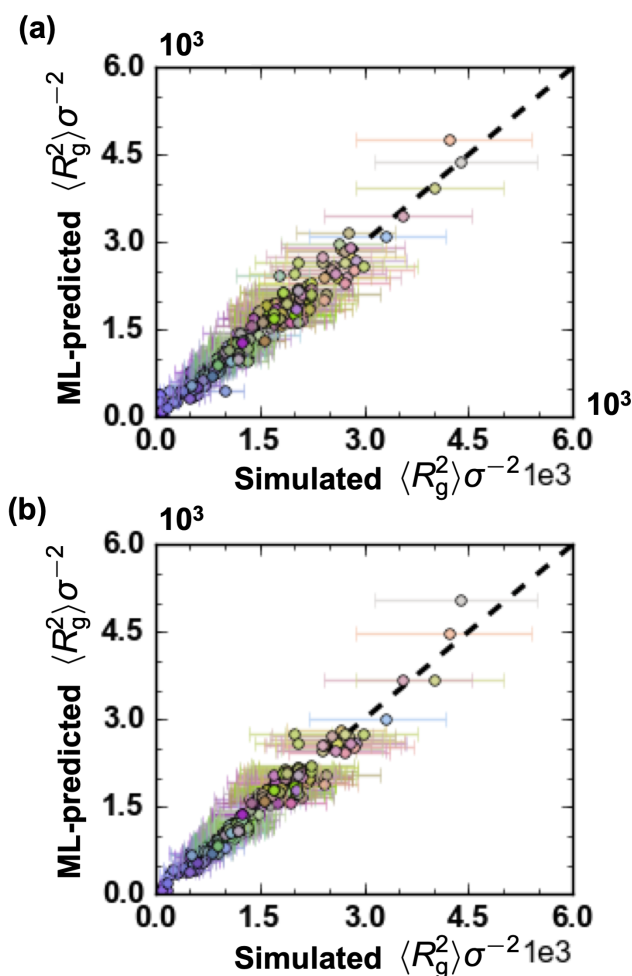


Figure S1: Comparison of machine learning regression model performance on 308 held-out Class I polymers for (a) OHE input featurization and (b) property-coloring featurization. For (a), the coefficient of determination (r^2) is 0.947, the mean absolute error (MAE) is $119.6\sigma^2$, and the standard deviation of absolute errors is $115.0\sigma^2$. For (b), $r^2 = 0.958$, MAE = $105.9\sigma^2$, and SDAE = $102.8\sigma^2$.

S3 Description of Python scripts and data files

This section provides a brief description of the data files included as supporting information.

S3.1 Class I Polymer Data

Raw data for Class I polymers is provided in the directory `ClassI_data`. Three files are included: `seqs.dat`, `avg.dat`, and `ps.dat`. `seqs.dat` provides a list of all the Class I polymers along with the corresponding 4-constitutional unit (4-CU) repeat unit, followed by all unique permutations and inversions. The number coding within `seqs.dat` matches that in Figure 1a of the main text. `avg.dat` provides all the simulated $\langle R_g^2 \rangle$ for the Class I polymers, which are matched by row to sequences in `seqs.dat`. Similarly, `ps.dat` provides the 25th (first column) and 75th (second column) percentile values for the distributions underlying $\langle R_g^2 \rangle$ for each sequence, again matched by row to those in `seqs.dat`.

S3.2 Class II Polymer Data

Raw data for Class II polymers is provided in the directory `ClassII_data`. Three files are included: `ran.seqs.dat`, `ran.avg.dat`, and `ran.ps.dat`. The files `ran.avg.dat` and `ran.ps.dat` are analogous to the files described for Class I polymers. Meanwhile, `ran.seqs.dat` is similar to `seqs.dat`, except that there are no permutations given, and the full sequence is shown using the number scheme in Figure 1a of the main text.

S3.3 Training Scripts

To exemplify the approach to training the machine-learning models, three Python scripts are included: `train_NN_OHE.py`, `train_LSTM_OHE.py`, and `train_CNN_mask.py`. All three scripts are written and tested for Python 3.7.4 and require scikit-learn and Keras, along with various modules that are distributed along with the Anaconda package. Respectively, these scripts train regression models using one-hot encoding of a 4-CU repeat unit, one-hot

encoding of an entire polymer sequence in tandem with a long short-term memory recurrent neural network, and property-coloring of the entire polymer sequence in tandem with a convolutional neural network. These featurization approaches and the machine learning architectures are described in the Materials and Methods Section of the main text. All three of these scripts also make use of a data manipulation module, `data_mod.py`, which reads and manipulates the formatted input/output files described in Sections S3.1 and S3.2. The script `train_CNN_mask.py` additionally makes use of the file `eps.mask`, which provides the property-coloring scheme for each of the CUs in Figure 1a of the main text.