Supporting Information

Predicting materials properties with little data using shotgun transfer learning

Hironao Yamada,†⊥ Chang Liu,†‡⊥ Stephen Wu,†⊥ Yukinori Koyama,‡
Shenghong Ju,§ Junichiro Shiomi,†§ Junko Morikawa,‡∥ and Ryo Yoshida*,†‡¶⊥

†The Institute of Statistical Mathematics, Research Organization of Information and
Systems, Tachikawa, Tokyo 190-8562, Japan
‡National Institute for Materials Science, Tsukuba, Ibaraki 305-0047, Japan
¶The Graduate University for Advanced Studies, Tachikawa, Tokyo 190-8562, Japan
§The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan
∥Tokyo Institute of Technology, Meguro-ku, Tokyo 152-8550, Japan
⊥Contributed equally to this work

E-mail: yoshidar@ism.ac.jp
XenonPy is a Python library that implements a comprehensive set of machine learning tools for materials informatics. The current release (v0.3.7: 2019/8/7) is a prototype version, which provides some limited modules. For details, see https://xenonpy.readthedocs.io.

XenonPy has the following features:

- An interface with public materials databases
- A library of materials descriptors (compositional/structural/molecular descriptors)
- The XenonPy.MDL pre-trained model library (v0.1.0b, 2019/7/31: more than 140,000 models with 35 properties of small molecules, polymers, and inorganic compounds, as listed in Table 1 in the main text)
- Machine learning tools
- A transfer learning feature using pre-trained models in XenonPy.MDL

Users can interact with the search API in Python using any given query strings to obtain a specific subset of pre-trained models. Furthermore, XenonPy offers a simple-to-use tool chain for seamless performance of transfer learning using a selected pre-trained model. The full list of currently available models and sample codes (for API querying, transfer learning, and so on) is provided at https://xenonpy.readthedocs.io/en/latest/features.html#xenonpy-mdl-and-transfer-learning. The library is ever-growing. Examples of the prediction performance exhibited by the current best-performing models are shown in Figure S1.
Figure S1: Prediction–observation plots for current best-performing models in XenonPy.MDL. Properties of (a) small molecules, (b) polymers, and (c) inorganic compounds are ordered from left to right.
Table S1: List of fingerprint descriptors in the rcdk and RDKit libraries that were used in building the shotgun model library.

| rcdk          | length  | RDKit                                      | length  |
|---------------|---------|--------------------------------------------|---------|
| standard      | 1,024   | basic fingerprints                         | 2,048   |
| extended      | 1,024   | atom pairs                                 | 2,048   |
| graph         | 1,024   | topological torsions                       | 2,048   |
| hybridization | 1,024   | Morgan fingerprints (without feature-based) | 2,048   |
| maccs         | 166     | Morgan fingerprints (with feature-based)    | 2,048   |
| estate        | 79      |                                            |         |
| pubchem       | 881     |                                            |         |
| kr            | 4,860   |                                            |         |
| circular      | 1,024   |                                            |         |
Figure S2: Chemical structures of the 52 polymers used in the task of predicting $C_P$. The training polymers were divided into six subgroups as numbered in the figure using the $K$-means clustering. Expert chemists annotated the identified clusters according to their compositional and structural features as (1) hydrocarbon mainchain polymers, (2) aliphatic esters, (3) phenols ethers, (4) aromatic esters, (5) N containing aromatics, and (6) diphenyl substituted metals. With this grouping, we performed the stratified group 6-fold CV to evaluate the generalization capability of transferred models.