Machine learning-assisted design of molecular structure of p-phenylenediamine antioxidants

Zongya Wu[a], Shuai Sun[a], Chaokun Huang[a], Li Zhou[a], Yanlong Luo[b], and Xiujuan Wang[a] *

a Key Laboratory of Advanced Rubber Material, Ministry of Education/Key Laboratory of Rubber-Plastics, Ministry of Education, Qingdao University of Science and Technology, Qingdao 266042, P.R. China.

b College of Science, Nanjing Forestry University, Nanjing 210037, PR. China

* Correspondence: wangxj@qust.edu.cn (X.W.); Tel.: 86-0532-84023536 (X.W.)

I. Choice for chemical space

To manage the size of the chemical space and ensure computational feasibility, the design was restricted to symmetric side chains. This restriction was applied to reduce the combinatorial complexity and to avoid an exponential increase in the number of molecular variants, which would otherwise exceed the practical limits of simulation and dataset generation. Meanwhile, this design strategy preserved the essential structure–property relationships relevant to antioxidant performance. In addition, symmetric substitution ensures that both aniline groups in p-phenylenediamine derivatives are located in identical chemical environments, resulting in consistent N–H bond dissociation energies (BDEs) at both reactive sites. This simplification also reflects real-world relevance, as many widely used aniline-based antioxidants are symmetric in structure, as reported in previous studies.[1]

To define a chemically relevant and practically feasible design space, the maximum length of the alkyl side chains was set to 20 carbon atoms. This upper limit was determined based on a review of existing antioxidant molecular structures. Molecular weights of most p-phenylenediamine-based antioxidants are typically found in the range of 100–300 g/mol, while those of other classes such as aromatic amines and hindered amine light stabilizers (HALS) have been reported to range up to 500 g/mol.[1] A few rare commercial examples, such as Tinuvin 123 (Decanedioic acid, bis(2,2,6,6-tetramethyl-1-(octyloxy)-4-piperidinyl) ester), have been documented with molecular weights exceeding 700 g/mol. In this study, the largest designed molecule—with two alkyl side chains each with 20 carbon atoms and a polar group—was estimated to reach approximately 700 g/mol. Therefore, side chains longer than 20 carbon atoms were excluded.

When the carbon chain length is 20, the presence of a methyl side group on the 20th carbon results in a molecule with a 21-carbon linear chain and no side group, which exceeds the defined chemical space. Therefore, to avoid this, $n = 20$ is not allowed, and the maximum value of n is limited to the range $[1, m - 1]$.

II. Molecular dynamics simulation procedure

For the force field selection, the COMPASSIII force field was employed for all molecular dynamics simulations involving both antioxidant molecules and triglyceride systems. This choice was based on the proven accuracy of COMPASS-based force fields in modeling complex organic materials, including lipids and nitrogen-containing functional groups.

Previous studies have demonstrated the reliability of this force field for similar systems. For instance, Li et al. applied the COMPASS force field to investigate the interactions between palmitic acid and supercritical CO_2, as well as the influence of temperature and pressure on its solubility behavior.[2] Zhu et al. further utilized the COMPASSIII force field to study the adsorption behavior and mechanisms of triglycerides on activated bamboo charcoal using molecular dynamics simulations.[3] These applications confirm the suitability of COMPASSIII for accurately representing both triglyceride dynamics and interactions with organic antioxidants.

In light of these validations and the force field's broad applicability to hydrocarbons and heteroatom-containing organics, we adopted COMPASSIII to simulate the antioxidant–triglyceride composite systems in this work. The detailed MD simulation setup is provided below.

MD simulations included the following parts: geometry optimization, anneal, NPT and NVT. MD simulations of the antioxidants and triglyceride models were performed under the COMPASSIII force field. All atomic charges were assigned according to the force field parameters.[4–7] The models were first subjected to multiple structural relaxations using a combination of the Smart minimization algorithm and the conjugate gradient method to minimize the system energy. The convergence criteria for the geometry optimization were set to 1.0×10^{-4} kcal/mol for energy, 0.005 kcal/mol/Å for force, and 5.0×10^{-5} Å for displacement, with a maximum of 2,000,000 iterations. During this stage, electrostatic and van der Waals interactions were treated with atom-based summation methods, using a cubic spline truncation and a cutoff distance of 18.5 Å.

After geometry optimization, the cells were annealed for 100 cycles between 300 K and 500 K to eliminate unreasonable local conformations and promote relaxation. Each annealing cycle consisted of 5 heating ramps, with 100 dynamic steps per ramp, totaling 100,000 steps. During the annealing process, atom-based summation methods were also used for non-bonded interactions with a cutoff distance of 12.5 Å.

Following annealing, a 500 ps dynamic simulation was performed under the canonical (NVT) ensemble at 298 K to allow sufficient structural relaxation. The time step was set to 1.0 fs, resulting in 500,000 integration steps. Initial atomic velocities were randomly assigned according to the Maxwell–Boltzmann distribution. The Andersen thermostat was employed to maintain a constant temperature during this process.[8]

Subsequently, the equilibrated structures underwent further relaxation under the isothermal-isobaric (NPT) ensemble at 298 K and 1 atm pressure. The NPT dynamics were conducted for 1000 ps (1.0 ns) with a time step of 1.0 fs, corresponding to 1,000,000 steps in total. The Andersen thermostat and the Berendsen barostat were used to control the system's temperature and pressure, respectively. The Ewald summation method was applied for the accurate treatment of long-range electrostatic interactions throughout the simulation.

III. Genetic algorithm procedure

During the GA iterations, candidate models with randomly initialized hyperparameters were generated and evaluated based on their coefficient of determination (R^2) and residual predictive deviation (*RPD*) values. These two metrics were jointly used as the fitness criteria to ensure a balance between model accuracy and generalization ability. Over successive generations, better-performing configurations were selected and modified through crossover and mutation operations, suitable hyperparameter combinations were obtained that improved the predictive performance and robustness of the final model. The hyperparameter search space was defined as follows: Number of hidden layers: 1 to 10; Number of neurons per layer: 16 to 512; Dropout rate: 0.0 to 0.5; Learning rate: 0.00001 to 0.01; Batch size: 16 to 256.

IV. Relative Percent Difference and R-squared

The formula for *RPD* is as follows:

$$RPD = \frac{SD_{Actual\ Values}}{SD_{Errors}} \qquad \text{Eq. (S1)}$$

where $SD_{Actual\ Values}$ is the standard deviation of actual values, SD_{Errors} is the standard deviation of errors, $RPD > 2.0$ indicates that the model has good predictive ability; $1.4 < RPD < 2.0$ indicates that the model has moderate predictive ability; $1 < RPD < 1.4$ indicates that the model has poor predictive ability; $RPD < 1$ indicates that the model has no predictive ability.[9]

The R^2 is given by

$$R^2 = \frac{\sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \qquad \text{Eq. (S2)}$$

where y_i is the empirical value for the i^{th} record, $\hat{y}_i$ is the predicted value for the i^{th} record, and $\bar{y}$ is the empirical mean value. A high R^2 value means a better prediction.

V. Normalization Method

The Z-Score is given by

$$x' = \frac{x - \mu}{\sigma} \qquad \text{Eq. (S3)}$$

where μ and σ is the mean value and standard deviation, respectively.

$\Delta\delta$ and BDE are normalized using the Min-Max method to obtain values between 0 and 1, eliminating dimensional differences, as follows:

$$Normalized\ Value = \frac{Value - Min(Value)}{Max(Value) - Min(Value)} \quad \text{Eq. (S4)}$$

The weighted average of the normalized values of $\Delta\delta$ and BDE is calculated as κ. When both parameters are given equal weight, κ is calculated as follows:

$$\kappa = \frac{Normalized(\Delta\delta) + Normalized(BDE)}{2} \quad \text{Eq. (S5)}$$

VI. Other Figures and Tables

Table S1 Software environment for obtaining descriptors in RDkit

Package Name	Version	Build String
boost	1.78.0	py311h12feb9d_4
boost-cpp	1.78.0	h9f4b32c_1
brotli	1.0.9	hcfcfb64_8
brotli-bin	1.0.9	hcfcfb64_8
bzip2	1.0.8	h8ffe710_4
ca-certificates	2023.01.10	haa95532_0
cairo	1.16.0	hd694305_1014
certifi	2022.12.7	py311haa95532_0
contourpy	1.0.7	py311h005e61a_0
cycler	0.11.0	pyhd8ed1ab_0
et_xmlfile	1.1.0	py311haa95532_0
expat	2.5.0	h1537add_0
font-ttf-dejavu-sans-mono	2.37	hab24e00_0
font-ttf-inconsolata	3	h77eed37_0
font-ttf-source-code-pro	2.038	h77eed37_0
font-ttf-ubuntu	0.83	hab24e00_0
fontconfig	2.14.2	hbde0cde_0
fonts-conda-ecosystem	1	0
fonts-conda-forge	1	0
fonttools	4.39.0	py311ha68e1ae_0
freetype	2.12.1	h546665d_1
gettext	0.21.1	h5728263_0
greenlet	2.0.2	py311h12c1d0e_0
icu	70.1	h0e60522_0
intel-openmp	2023.0.0	h57928b3_25922
kiwisolver	1.4.4	py311h005e61a_1
lcms2	2.15	h3e3b177_1
lerc	4.0.0	h63175ca_0
libblas	3.9.0	16_win64_mkl
libbrotlicommon	1.0.9	hcfcfb64_8
libbrotlidec	1.0.9	hcfcfb64_8
libbrotlienc	1.0.9	hcfcfb64_8
libcblas	3.9.0	16_win64_mkl
libdeflate	1.17	hcfcfb64_0
libffi	3.4.2	h8ffe710_5

libglib	2.74.1	he8f3873_1
libhwloc	2.9.0	h51c2c0f_0
libiconv	1.17	h8ffe710_0
libjpeg-turbo	2.1.5.1	hcfcfb64_0
liblapack	3.9.0	16_win64_mkl

(continued)

Package Name	Version	Build String
(continued)		
libpng	1.6.39	h19919ed_0
libsqlite	3.40.0	hcfcfb64_0
libtiff	4.5.0	hc3b8658_5
libwebp-base	1.2.4	h8ffe710_0
libxcb	1.13	hcd874cb_1004
libxml2	2.10.3	hc3477c8_0
libzlib	1.2.13	hcfcfb64_4
m2w64-gcc-libgfortran	5.3.0	6
m2w64-gcc-libs	5.3.0	7
m2w64-gcc-libs-core	5.3.0	7
m2w64-gmp	6.1.0	2
m2w64-libwinpthread-git	5.0.0.4634.697f757	2
matplotlib-base	3.7.1	py311h6e989c2_0
mkl	2022.1.0	h6a75c08_874
msys2-conda-epoch	20160418	1
munkres	1.1.4	pyh9f0ad1d_0
numpy	1.24.2	py311h0b4df5a_0
openjpeg	2.5.0	ha2aaf27_2
openpyxl	3.0.10	py311h2bbff1b_0
openssl	3.0.8	hcfcfb64_0
packaging	23	pyhd8ed1ab_0
pandas	1.5.3	py311hf63dbb6_0
pcre2	10.4	h17e33f8_0
pillow	9.4.0	py311h339194b_0
pip	23.0.1	pyhd8ed1ab_0
pixman	0.40.0	h8ffe710_0
pthread-stubs	0.4	hcd874cb_1001
pthreads-win32	2.9.1	hfa6e2cd_3
pycairo	1.23.0	py311h99894aa_0
pyparsing	3.0.9	pyhd8ed1ab_0
python	3.11.0	hcf16a7b_0_cpython
python-dateutil	2.8.2	pyhd8ed1ab_0
python_abi	3.11	3_cp311
pytz	2022.7.1	pyhd8ed1ab_0
rdkit	2022.09.5	py311h5f60fae_0
reportlab	3.6.12	py311hae2e3ae_2

setuptools	67.5.1	pyhd8ed1ab_0
six	1.16.0	pyh6c4a22f_0
sqlalchemy	2.0.5.post1	py311ha68e1ae_0
tbb	2021.8.0	h91493d7_0
tk	8.6.12	h8ffe710_0

(continued)

Package Name	Version	Build String
(continued)		
typing-extensions	4.4.0	hd8ed1ab_0
typing_extensions	4.4.0	pyha770c72_0
tzdata	2022g	h191b570_0
ucrt	10.0.22621.0	h57928b3_0
vc	14.3	hb6edc58_10
vs2015_runtime	14.34.31931	h4c5c07a_10
wheel	0.38.4	pyhd8ed1ab_0
xorg-libxau	1.0.9	hcd874cb_0
xorg-libxdmcp	1.1.3	hcd874cb_0
xz	5.2.6	h8d14728_0
zlib	1.2.13	hcfcfb64_4
zstd	1.5.2	h12be248_6

Table S2 Software environment for modelling

Package Name	Version	Build String
_tflow_select	2.3.0	mkl
absl-py	1.4.0	py310haa95532_0
aiohttp	3.9.0	py310h2bbff1b_0
aiosignal	1.2.0	pyhd3eb1b0_0
astunparse	1.6.3	py_0
async-timeout	4.0.2	py310haa95532_0
attrs	23.1.0	py310haa95532_0
blas	1	mkl
blinker	1.6.2	py310haa95532_0
bottleneck	1.3.7	py310h9128911_0
brotli	1.0.9	h2bbff1b_7
brotli-bin	1.0.9	h2bbff1b_7
brotli-python	1.0.9	py310hd77b12b_7
bzip2	1.0.8	he774522_0
ca-certificates	2024.3.11	haa95532_0
cachetools	4.2.2	pyhd3eb1b0_0
certifi	2024.2.2	py310haa95532_0
cffi	1.16.0	py310h2bbff1b_0
charset-normalizer	2.0.4	pyhd3eb1b0_0
click	8.1.7	py310haa95532_0

cloudpickle	2.2.1	py310haa95532_0
colorama	0.4.6	py310haa95532_0
contourpy	1.2.0	py310h59b6b97_0
cryptography	41.0.3	py310h3438e0d_0
cycler	0.11.0	pyhd3eb1b0_0

(continued)

Package Name	Version	Build String
(continued)		
et_xmlfile	1.1.0	py310haa95532_0
flatbuffers	2.0.0	h6c2663c_0
fonttools	4.25.0	pyhd3eb1b0_0
freetype	2.12.1	ha860e81_0
frozenlist	1.4.0	py310h2bbff1b_0
gast	0.4.0	pyhd3eb1b0_0
giflib	5.2.1	h8cc25b3_3
glib	2.78.4	hd77b12b_0
glib-tools	2.78.4	hd77b12b_0
google-auth	2.22.0	py310haa95532_0
google-auth-oauthlib	0.4.4	pyhd3eb1b0_0
google-pasta	0.2.0	pyhd3eb1b0_0
grpcio	1.42.0	py310hc60d5dd_0
gst-plugins-base	1.18.5	h9e645db_0
gstreamer	1.18.5	hd78058f_0
h5py	3.9.0	py310hfc34f40_0
hdf5	1.12.1	h51c971a_3
icc_rt	2022.1.0	h6049295_2
icu	58.2	ha925a31_3
idna	3.4	py310haa95532_0
intel-openmp	2023.1.0	h59b6b97_46320
joblib	1.2.0	py310haa95532_0
jpeg	9e	h2bbff1b_1
keras	2.10.0	py310haa95532_0
keras-preprocessing	1.1.2	pyhd3eb1b0_0
kiwisolver	1.4.4	py310hd77b12b_0
krb5	1.19.4	h5b6d351_0
lerc	3	hd77b12b_0
libbrotlicommon	1.0.9	h2bbff1b_7
libbrotlidec	1.0.9	h2bbff1b_7
libbrotlienc	1.0.9	h2bbff1b_7
libclang	14.0.6	default_hb5a9fac_1
libclang13	14.0.6	default_h8e68704_1
libcurl	8.4.0	h86230a5_1
libdeflate	1.17	h2bbff1b_1
libffi	3.4.4	hd77b12b_0

libglib	2.78.4	ha17d25a_0
libiconv	1.16	h2bbff1b_2
libogg	1.3.5	h2bbff1b_1
libpng	1.6.39	h8cc25b3_0
libprotobuf	3.20.3	h23ce68f_0

(continued)

Package Name	Version	Build String
(continued)		
libssh2	1.10.0	hcd4344a_2
libtiff	4.5.1	hd77b12b_0
libvorbis	1.3.7	he774522_0
libwebp-base	1.3.2	h2bbff1b_0
llvmlite	0.42.0	py310hf2fb9eb_0
lz4-c	1.9.4	h2bbff1b_0
markdown	3.4.1	py310haa95532_0
markupsafe	2.1.1	py310h2bbff1b_0
matplotlib	3.8.0	py310haa95532_0
matplotlib-base	3.8.0	py310h4ed8f06_0
mkl	2023.1.0	h6b88ed4_46358
mkl-service	2.4.0	py310h2bbff1b_1
mkl_fft	1.3.8	py310h2bbff1b_0
mkl_random	1.2.4	py310h59b6b97_0
multidict	6.0.4	py310h2bbff1b_0
munkres	1.1.4	py_0
numba	0.59.1	py310hd77b12b_0
numexpr	2.8.7	py310h2cd9be0_0
numpy	1.26.2	py310h055cbcc_0
numpy-base	1.26.2	py310h65a83cf_0
oauthlib	3.2.2	py310haa95532_0
openjpeg	2.4.0	h4fc8c34_0
openpyxl	3.0.10	py310h2bbff1b_0
openssl	1.1.1w	h2bbff1b_0
opt_einsum	3.3.0	pyhd3eb1b0_1
packaging	23.1	py310haa95532_0
pandas	2.1.4	py310h4ed8f06_0
pcre2	10.42	h0ff8eda_0
pillow	10.2.0	py310h2bbff1b_0
pip	23.3.1	py310haa95532_0
ply	3.11	py310haa95532_0
protobuf	3.20.3	py310hd77b12b_0
pyasn1	0.4.8	pyhd3eb1b0_0
pyasn1-modules	0.2.8	py_0
pycparser	2.21	pyhd3eb1b0_0
pyjwt	2.4.0	py310haa95532_0

pyopenssl	23.2.0	py310haa95532_0
pyparsing	3.0.9	py310haa95532_0
pyqt	5.15.10	py310hd77b12b_0
pyqt5-sip	12.13.0	py310h2bbff1b_0
pysocks	1.7.1	py310haa95532_0

(continued)

Package Name	Version	Build String
(continued)		
python	3.10.3	hbb2ffb3_5
python-dateutil	2.8.2	pyhd3eb1b0_0
python-flatbuffers	2	pyhd3eb1b0_0
python-tzdata	2023.3	pyhd3eb1b0_0
pytz	2023.3.post1	py310haa95532_0
qt-main	5.15.2	he8e5bd7_8
requests	2.31.0	py310haa95532_0
requests-oauthlib	1.3.0	py_0
rsa	4.7.2	pyhd3eb1b0_1
scikit-learn	1.3.0	py310h4ed8f06_0
scipy	1.11.4	py310h309d312_0
setuptools	68.0.0	py310haa95532_0
shap	0.42.1	py310h4ed8f06_0
sip	6.7.12	py310hd77b12b_0
six	1.16.0	pyhd3eb1b0_1
slicer	0.0.7	pyhd3eb1b0_0
snappy	1.1.9	h6c2663c_0
sqlite	3.41.2	h2bbff1b_0
tbb	2021.8.0	h59b6b97_0
tensorboard	2.10.0	py310haa95532_0
tensorboard-data-server	0.6.1	py310haa95532_0
tensorboard-plugin-wit	1.8.1	py310haa95532_0
tensorflow	2.10.0	mkl_py310hd99672f_0
tensorflow-base	2.10.0	mkl_py310h6a7f48e_0
tensorflow-estimator	2.10.0	py310haa95532_0
termcolor	2.1.0	py310haa95532_0
threadpoolctl	2.2.0	pyh0d69192_0
tk	8.6.12	h2bbff1b_0
tomli	2.0.1	py310haa95532_0
tornado	6.3.3	py310h2bbff1b_0
tqdm	4.65.0	py310h9909e9c_0
typing_extensions	4.7.1	py310haa95532_0
tzdata	2023c	h04d1e81_0
urllib3	1.26.18	py310haa95532_0
vc	14.2	h21ff451_1
vs2015_runtime	14.27.29016	h5e58377_2

werkzeug	2.2.3	py310haa95532_0
wheel	0.41.2	py310haa95532_0
win_inet_pton	1.1.0	py310haa95532_0
wrapt	1.14.1	py310h2bbff1b_0
xz	5.4.5	h8cc25b3_0

(continued)

Package Name	Version	Build String
(continued)		
yarl	1.9.3	py310h2bbff1b_0
zlib	1.2.13	h8cc25b3_0
zstd	1.5.5	hd43e919_0

Table S3 Comparisons of *BDE* results from simulation and experiment

SMILES for antioxidant	BDE_{sim} (kJ/mol)	BDE_{exp}[10] (kJ/mol)	Relative Error
Nc1ccc(N)cc1	353.9	359.8	1.64 %
CN(C)c1ccc(NC)cc1	338.7	342.4	1.08 %
c2ccccc2Nc1ccc(Nc3ccccc3)cc1	347.8	352.8	1.42 %
c1c(NC(C)C)ccc(Nc2ccccc2)c1	345.0	350.2	1.48 %
c1c(Nc3ccc(Nc4ccc5ccccc5c4)cc3)cc2ccccc2c1	351.1	354.6	0.99 %
c1ccccc1NCCNc2ccccc2	363.3	372.7	2.52 %

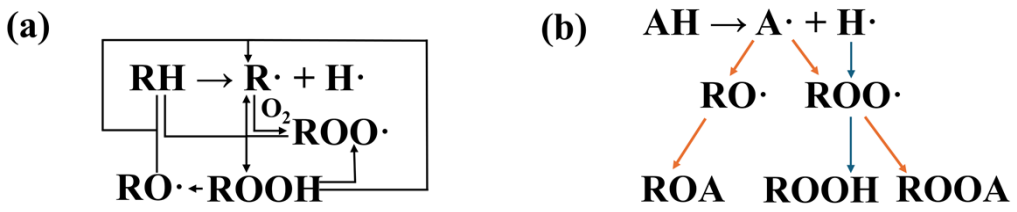

Fig. S1 (a) Oxidant process of edible oil; (b) Antioxidant process of antioxidant molecule (AH)

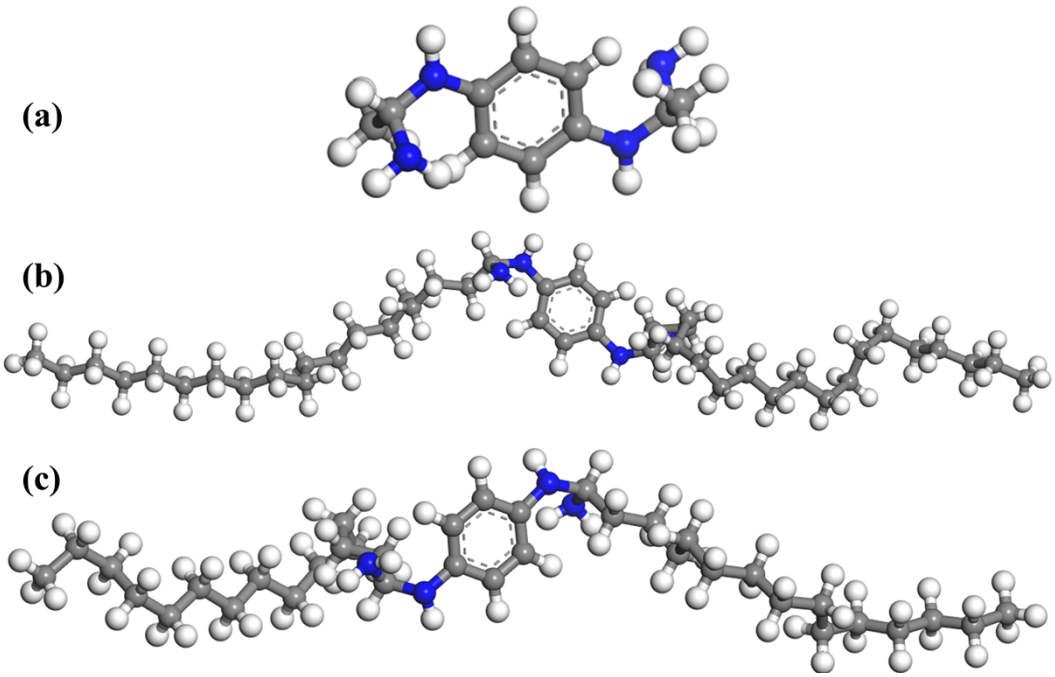

Fig. S2 Molecular structures of the: (a) 'N-2-1' molecule (smallest δ); (b) ' N-19-1' molecule (smallest $\Delta\delta$); (c) 'N-15-1' molecule (smallest κ)

Notes and references

1 E. T. Denisov and T. G. Denisova, Dissociation energy of N-H bonds in diatomic aromatic amines, *Kinet. Catal.*, 2014, **55**, 28–34.

2 F. Li, F. Sun, Z. Li, Z. Zheng and W. Wang, Molecular dynamics simulation of the interaction between palmitic acid and high pressure CO2, *R. Soc. Open Sci*, 2023, **10**, 231141.

3 H. Zhu, S. Zhang, H. Zheng and G. Wang, Enhanced triglyceride adsorption by steam-activated bamboo charcoal based on molecular dynamics investigations, *Sci Rep*, 2024, **14**, 6237.

4 Z. Wu, X. Shi, X. Wang, Y. Luo and X. Wang, Modeling and analysis of synergistic phenolic antioxidants in edible oil: Integrating molecular simulation and artificial neural network approach, *J. Mol. Struct.*, 2024, 138959.

5 X. Wang, X. Chen, M. Song, Q. Wang, W. Zheng, H. Song, Z. Fan and A. Myat Thu, Effects of hindered phenol organic molecules on enhancing thermo-oxidative resistance and damping capacity for nitrile butadiene rubber: Insights from Experiments and Molecular Simulation, *Ind. Eng. Chem. Res.*, 2020, **59**, 11494–11504.

6 X. Wang, M. Song, S. Liu, S. Wu and A. M. Thu, Analysis of phthalate plasticizer migration from PVDC packaging materials to food simulants using molecular dynamics simulations and artificial neural network, *Food Chem.*, 2020, **317**, 126465.

7 S. Du, X. Wang, R. Wang, L. Lu, Y. Luo, G. You and S. Wu, Machine-learning-assisted molecular design of phenylnaphthylamine-type antioxidants, *Phys. Chem. Chem. Phys.*, 2022, **24**, 13399–13410.

8 X. Wang, C. Huang, X. Wang, Y. Luo and X. Wang, Multiscale simulation study on radiation aging of EPDM and preparation of radiation-resistant materials, *Compos. Sci. Technol.*, 2024, **252**, 110595.

9 T. Fearn, Assessing calibrations: SEP, RPD, RER and R2, *NIR N.*, 2002, **13**, 12–13.

10 E. T. Denisov and T. G. Denisova, Dissociation energy of N-H bonds in diatomic aromatic amines, *Kinet. Catal.*, 2014, **55**, 28–34.