Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Supplementary Material: Materials Funnel 2.0 - Data-driven hierarchical search for exploration of vast chemical spaces

Raul Ortega Ochoa,[a] Bardi Benediktsson,[a] Renata Sechi,[a] Peter Bjørn Jørgensen,[a] and Arghya Bhowmik[a*]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

1 Supplementary section

Influence of stochasticity in the generation of 3d coordinates on the predictions of the surrogate model.

To transform SMILES[1] to XYZ coordinates, we first create an initial 3D geometry with *OpenBabel*[2,3] followed by optimization of the structure using UFF forcefield. During initial structure guessing, random numbers are used * resulting in slightly different final 3D coordinates each time we do the transformation. Given the stochastic nature of obtaining the 3D structure from the SMILES strings, it is worth studying its effect on the predictions made by the surrogate model.

The surrogate model was trained on a dataset which lacked stochasticity in structures. This is because re-running the structure optimization for every molecule for every epoch would be too time-consuming. To assess the effect of the random initialization of the XYZ coordinates on the surrogate predictions, 32 SMILES are selected from the 4600 generated molecules during the *Funnel Pipeline validation*, covering the range 1.6-4.6 eV of the HOMO-LUMO gap in steps of approximately 0.1 eV. For each of the 32 selected SMILES, *OpenBabel* is used to generate 100 UFF-optimized 3D structures per molecule with different random initializations, resulting in 3200 samples. For each of the 3200 samples, the surrogate model was used to predict the HOMO-LUMO gap.

The surrogate predictions are then aggregated grouping the predicted HOMO-LUMO gap values by molecule, and the mean, maximum, and minimum of the predictions for each of the 32 molecules are calculated. These aggregated results are then shown in the form of violin plots, where the y axis corresponds to the surrogate HOMO-LUMO predictions, and the x axis corresponds to the DFT-computed HOMO-LUMO values from the dataset. The mean, maximum, and minimum of the surrogate predictions are then shown in the molecule's violin plot.

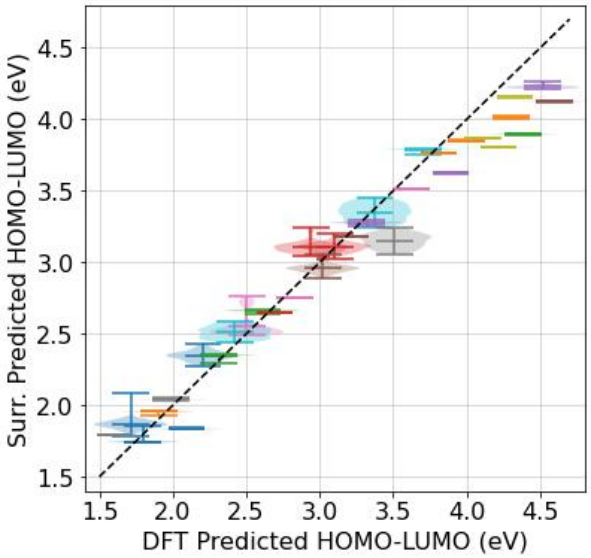

Fig. S1 Surrogate predictions of the HOMO-LUMO gap versus DFT-predicted values for 32 selected molecules from the CEP dataset[4] on 100 different random initialization of the initial 3D structure. The surrogate predictions are grouped by molecule and the mean, maximum, and minimum of the predictions are calculated. The violin plots allow for visualization of the distribution of the prediction for each molecule beyond the aggregated metrics of mean, maximum, and minimum. The black diagonal line represents the ideal correlation of the predictions.

From Fig. S1, the influence of random initialization of the 3D structure results in surrogate predictions that differ at most 0.2

[a]*Technical University of Denmark, DTU Energy, Anker Engelunds Vej, Building 301 2800 Kgs. Lyngby, Denmark, E-mail: arbh@dtu.dk.*

* See discussion in *OpenBabel* `https://github.com/openbabel/openbabel/issues/1934`

eV (min-max range in predictions). This maximum error is of the order of the mean absolute error (MAE) of the predictions of the surrogate model. Hence we can conclude that the stochastic nature of the 3D-coordinate generation affects the predictions, but it is within the accuracy of the model.

Validation of the surrogate model as a binary classifier

The objective of the surrogate model within the HTCS 2.0 is to rapidly pre-filter candidate materials into a smaller, curated set of promising materials that can be then further pruned using more expensive such as DFT into the final output of the HTCS 2.0.

To that purpose, the surrogate model must be able to compute property predictions rapidly and accurately. The accuracy of the surrogate model was thoroughly evaluated in the main work in terms of the distribution of errors in its predictions with respect to the higher-fidelity DFT calculations. In this section, we present an alternative way of quantifying the ability of the surrogate model to act as the prefilter for the more expensive DFT calculations.

The role of the surrogate model can be interpreted as a classification problem with two classes: *promising* and *non-promising* molecules. A *promising* molecule according to the surrogate is one whose predicted property is close to the desired property value, the *prompted condition*. Given some molecule and some *prompted condition* the surrogate model predicts a property of the molecule. If the prediction is close to the *prompted condition* the molecule is classified as *promising*, otherwise, it is classified as *non-promising*. The *closeness* criteria for the classification is some predefined threshold.

By interpreting the task of the surrogate model as a classification problem, we can analyze the number of molecules that are classified correctly or wrongly. The true class of some sample is the class it belongs to in accordance with its DFT value, and the predicted class of some sample is the class it belongs to by its surrogate-predicted value.

An example: Given some desired property value c (*prompted condition*), some threshold δ, some molecule $\mathscr{M}$ whose property value predicted by the surrogate is $\hat{c}_{surr}$ and its predicted value from the DFT calculation in the dataset is $\hat{c}_{dft}$, then for the predictions from the surrogate:

$$abs(c-\hat{c}_{surr}) \rightarrow \begin{cases} \text{predicted } \textit{promising} & if \leq \delta \\ \text{predicted } \textit{non-promising} & if > \delta \end{cases}$$

Similarly, based on the DFT predictions:

$$abs(c-\hat{c}_{dft}) \rightarrow \begin{cases} \text{Actual } \textit{promising} & if \leq \delta \\ \text{Actual } \textit{non-promising} & if > \delta \end{cases}$$

Note that since the predictions from DFT and the Surrogate model are not the same, there are four possible scenarios for the given molecule:

- **True *promising***: If *predicted promising* and *actual promising*.
- **False *promising***: If *predicted promising* and *actual non-promising*.
- **False *non-promising***: If *predicted non-promising* and *actual promising*.
- **True *non-promising***: If *predicted non-promising* and *actual non-promising*.

We are interested in quantifying how many of the candidates that the surrogate classifies fall on each of the four cases because it allows, for example, to obtain a measure of how many good candidates we erroneously discard in the HTCS 2.0 (False *non-promising*). To do so, we can use the data from the *Funnel Pipeline validation*. The data consist of 4600 molecules generated for a range of 46 different conditions spanning the range of the HOMO-LUMO gap from 1.0-5.5 eV. Of the 4600 molecules generated, 2221 were present in the CEP dataset (*not novel* molecules), so the DFT-calculated value of the HOMO-LUMO gap is known from the dataset. This is the value that will be taken as a reference, $\hat{c}_{dft}$.

The 2221 molecules are then divided into the four different cases described. The raw counts and the percentage of the total number of molecules are shown as a confusion matrix in *table* 1. The threshold chosen was $\delta = 0.149$, in accordance with the experiment in *Validating the Surrogate*.

Predicted/Actual	promising	non-promising
promising	628 (**28.27 %**)	128 (**5.76 %**)
non-promising	270 (**12.15 %**)	1195 (**53.80 %**)

Table 1 Confusion matrix on the classification of generated molecules based on the Surrogate model predictions of the HOMO-LUMO gap. The Actual/True values are taken from the DFT-predicted values for the molecules in the dataset.

From the confusion matrix in *table* 1 82.07% of the molecules were correctly classified by the Surrogate filter (diagonal in the confusion matrix) and approximately one-third (34.03% = 28.27%+5.76%) of the total molecules are classified by the filter as *promising*. This demonstrates that employing the Surrogate model to pre-filter molecules reliably (with a low percentage of error) reduces the load on heavier computationally expensive DFT calculations to 1/3 of its original load (without the surrogate filter).

Un-conditional versus Conditional generators for HTCS 2.0

The proposed approach for HTCS 2.0 includes a conditional generator to produce an initial set of candidate molecules subject to some design condition (*prompted* condition) that are then narrowed down in the subsequent blocks of the HTCS 2.0 funnel to form a highly curated set of candidates. One of the key characteristics of the generator used is its capacity to create molecular structures that are biased to have the target property, so that the initial set of produced molecules has a higher population of good

candidate molecules for the design condition than if the initial set were produced by randomly sampling the entire chemical space.

The objective of this section is to illustrate the advantage of employing conditional generators instead of nonconditional generators. We do it by comparing the probability of both types of generators to produce candidate molecules that would be present in the final curated list output of the HTCS 2.0 funnel.

To simplify the problem, we can directly look at the probability of some molecule generated to be within some neighborhood of the *prompted condition* c.

A nonconditional generator learns to approximate the underlying dataset distribution $p(m)$ over the sample space $m \in \mathscr{M}$ of molecules. Sampling a molecule from the learned distribution $m \sim q_\theta(m)$ we should expect for the property of the sampled molecule $\hat{c}$ to follow the distribution in the original dataset. This distribution for the HOMO-LUMO gap resembles a Gaussian distribution $\mathscr{N}(\mu_{dset} = 2.79, \sigma_{dset} = 0.44)$, so we can expect $\hat{c} \sim \mathscr{N}(\mu_{dset}, \sigma_{dset})$. The probability of some molecule to have a property value within the neighborhood δ of the *prompted condition* can be written as $P(|\hat{c} - c| \leq \delta)$. Since $\hat{c} \sim \mathscr{N}(\mu_{dset}, \sigma_{dset})$ we can then write:

$$P_{uncond}(|\hat{c} - c| \leq \delta) = \int_{c-\delta}^{c+\delta} \frac{1}{\sigma_{dset}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_{dset}}{\sigma_{dset}}\right)^2} dx \qquad (1)$$

The conditional generator learns to sample molecules from the learned distribution while being subjected to some condition. From the analysis in *Validating the Generator*, we showed that for a *prompted condition* c *in-distribution* the conditional generator is well calibrated: For a molecule created subject to a *prompted condition* c we can expect the property value $\hat{c}$ to be following a Gaussian-like distribution centered in c with standard deviation $\sigma_{gen} \approx 0.13$. Then we can write for the conditional generator:

$$P_{cond}(|\hat{c} - c| \leq \delta) = \int_{c-\delta}^{c+\delta} \frac{1}{\sigma_{gen}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-c}{\sigma_{gen}}\right)^2} dx \qquad (2)$$

Using these two expressions for the probability that a generated molecule has a property value within $(c-\delta, c+\delta)$, we can numerically compute probabilities for different conditions c. In Fig. S2 for a condition $c = 2.3$ eV, the probability distribution of the property value for a sample molecule from the conditional (blue) and unconditional (green) model is shown. The area under the curves, delimited by the $(c-\delta, c+\delta)$ values corresponds to the probabilities $p_{cond} = P_{cond}(|\hat{c} - c| \leq \delta)$, and $p_{uncond} = P_{uncond}(|\hat{c} - c| \leq \delta)$.

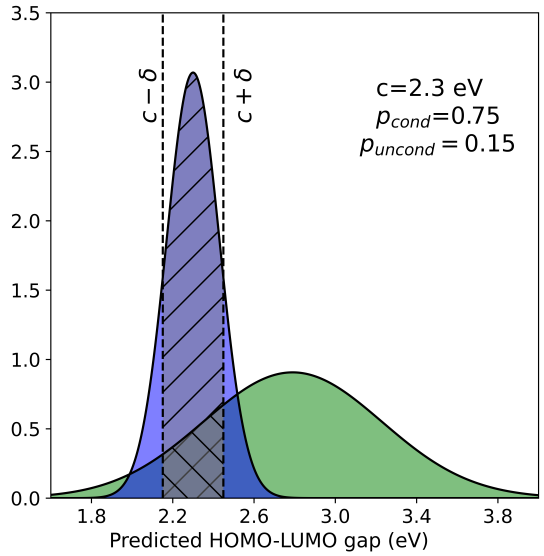

Fig. S2 Probability distribution of the property values for a sample molecule from the conditional (blue) and unconditional (green) model for $c = 2.3$ eV. The area under the curves, delimited by the $(c-\delta, c+\delta)$ values corresponds to the probabilities p_{cond}, and p_{uncond}.

Fig. S2 illustrates the advantage of the conditional model versus the unconditional model. We can then compute the ratio p_{cond}/p_{uncond} for condition values in the range of HOMO-LUMO values from percentile 5-95 (approximately 2.1-3.5 eV) to illustrate the advantage of the conditional model as a function of the condition given. Beyond percentile 5-95, the approximation $\hat{c} \sim \mathscr{N}(c, \sigma_{gen} = 0.13)$ for the conditional generator does not hold. As was shown in *Validating the Generator*, near the boundary of the 5-95 percentiles, the mean of the prediction values departs from the ideal correlation (s. Fig. 7 from the main article). For this reason, we focus the study region in the percentile 5-95 range.

Fig. S3 demonstrates that the probability of success from the conditional model is always greater than the one from the unconditional. The range of ratios between probabilities shown range from the minima at 2.82 to 10, indicating that the conditional model is at worst 280% more efficient than the unconditional, and at its best within the percentiles 5-95 range, 1000% more efficient.

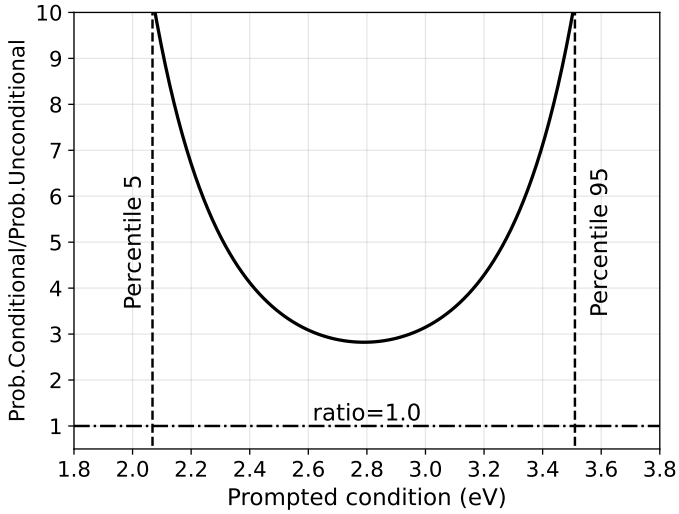

Fig. S3 Ratio between the probability of the conditional and unconditional model to produce a sample molecule whose property lies within $(c-\delta, c+\delta)$. The black curved line corresponds to $P_{cond}(|\hat{c}-c| \leq \delta))/P_{uncond}(|\hat{c}-c| \leq \delta))$ as a function of the *prompted condition* c, the HOMO-LUMO gap. The ratio is computed only for *prompted condition* c values within the percentiles 5-95 of the distribution of values in the dataset. The two dashed vertical lines indicate the percentiles 5, and 95. The dash-dotted horizontal line indicates the ratio value 1.0, where the conditional and unconditional generator would have equal probability. The minimum of the ratio function is ratio=2.82.

Spin multiplicity analysis of *novel* molecules.

id	SMILES	Singlet (Ha)	Triplet (Ha)	Quintet (Ha)
37c5448086154e6905c24e6b135d9f3eb31a0a04	C1=c2ccc3sc4c5oc(-c6ccco6)cc5c5nsnc5c4c3c2=C[SiH2]1	**-2113.61**	-2113.574	-2113.497
b1e95c76ea46696db4dc985482ac5d80914b3819	C1=CCC(c2ccc(-c3cnc(-c4scc5nccnc45)s3)c3cocc23)=C1	**-1881.638**	-1881.597	-1881.533
c5158d9277edf919c9bc15e8f614e1e8ead51287	C1=c2c(c3c([nH]c4cc(-c5cccc6c[nH]cc56)sc43)c3[se]ccc23)=C[SiH2]1	**-4045.063**	-4045.023	-4044.952
c7d2a3d94a7989ea363ff2bbd5c68e84274faa03	C1=Cc2csc(-c3scc4c3[se]c3c5cc6ccccc6cc5ccc43)c2[SiH2]1	**-4409.963**	-4409.898	-4409.82
52a61d8bc82a255fb0a4ae1be48c7af375148cde	C1=CCC(c2cnc(-c3sc(-c4scc5occc45)c4cc[se]c34)c3nsnc23)=C1	**-4680.041**	-4680.008	-4679.923
09d6f0c87a53a0f384fa534adee5d31290ff0f05	c1ccc(-c2[se]cc3c2[se]c2ccc4ccc5ccccc5c4c23)nc1	**-5741.686**	-5741.611	-5741.514
7422901f9feab557f1c44418646a659afe864004	C1=Cc2csc(-c3c4cocc4cc4c3oc3oc5[se]ccc5c34)c2[SiH2]1	**-4082.575**	-4082.515	-4082.422
a4304a4f034eaf203760d2eb25b558788db7207b	C1=Cc2c(csc2-c2sc(-c3sc(-c4nccs4)c4ccsc34)c3nccnc23)C1	**-2999.71**	-2999.684	-2999.595
1d6a6dadfcb9e1f0a8ffe4800e70abb0afb3dec9	C1=Cc2c(c3ccc4occ(-c5cccc6nsnc56)c4c3c3ccncc23)C(c2nccc3nsnc23)C1	**-2352.488**	-2352.412	-2352.34
2ca58b75dbabfae2257f3dbd2a4bb16352f39093	c1cc2cc3c(nc2cn1)oc1c3ccc2nccc(-c3cc4ccsc4s3)c21	**-1918.615**	-1918.535	-1918.438
07031182607a9dab51968858bef30495af60e79b	C1=Cc2csc(-c3c4cocc4cc4c3oc3oc5occc5c34)c2[SiH2]1	**-1756.328**	-1756.27	-1756.177
a1e4873f6840d5969e8913c85bd0f5449dbd2768	C1=c2c(c3c4[se]c(-c5cccc6cocc56)cc4[se]c3c3sccc23)=CC1	**-6159.656**	-6159.604	-6159.54
6ca23738fb6ffcdfc167c7f2da661b412b009755	C1=Cc2csc(-c3c4c(cc5c6cc[nH]c6c6nsnc6c35)=CCC=4)c2C1	**-1806.425**	-1806.393	-1806.299
a73c760e82a0fa90b74b7a1505180f0493e7b7d1	C1=Cc2c(csc2C2=Cc3[se]c4c(sc5ccc6c(c54)=C[SiH2]C=6)c3[SiH2]2)C1	**-4585.068**	-4585.034	-4584.98
96ed5acde42f06cba280569d8b24a550b8fe88c8	C1=Cc2c(-c3cc4ccsc4s3)sc(C3=CC=C4c5c(sc6[se]ccc56)SC4C3)c2[SiH2]1	**-5604.37**	-5604.324	-5604.242
ab7e6bdde5f39a6a65a8d8deea6a695735d0ca92	c1ccc2c(c1)ccc1scc(-c3cncc4nsnc34)c12	**-1613.6**	-1613.531	-1613.436
150449e2ca10ed5353e85cefdcf759b11eb33efa	C1=Cc2c(c3oc4cc(-c5cccc6c[nH]cc56)[se]c4c3c3nsnc23)C1	**-3845.672**	-3845.616	-3845.532
3c17bf295b40fa1a51ad7b5480c778f154fa1e8b	C1=Cc2csc(-c3c4cscc4cc4c3oc3[se]c5sccc5c34)c2[SiH2]1	**-4728.498**	-4728.437	-4728.344
d5947f44656c0af53b5e47e52471e76978ea5f8a	c1cc2c(cn1)c1c[nH]cc1c1cnc3c(-c4cc5ccsc5s4)c[nH]c3c21	**-1860.642**	-1860.558	-1860.456
e38b884ca81d9e5ccbfc399b52804510c7760a68	C1=Cc2csc(-c3cc4cc5c(cnc6ccccnc65)cc4o3)c2C1	**-1390.103**	-1390.028	-1389.926
0c310c805c11292df789d828c0f1b465f8927814	c1ccc(-c2cc3ccccc3c3cc4c(cc23)oc2cnccc24)nc1	**-1107.468**	-1107.381	-1107.271
4d936bdd9c0bf7c8d7934aee1df40edf8a5ee8f8	C1=C(c2scc3[se]ccc23)Cc2c1c1c(c3c4[nH]ccc4c4c[nH]cc4c23)=C[SiH2]C=1	**-4160.538**	-4160.511	-4160.441
a68707583ff08f441cc46a828977c5bb27fd348c	C1=Cc2c(-c3cccs3)sc(-c3c(-c4ccc[se]4)ccc4c[nH]cc34)c2[SiH2]1	**-4389.067**	-4388.996	-4388.919
f48c5c219dc5ff03d51e9a2c8cd5df3b535d1b02	C1=Cc2c(cnc3c2C=C2C(c4ccco4)=CNC2C3)C1	**-878.81**	-878.748	-878.638
0608dca27ce23310c8f114191593988e9ba6f787	C1=Cc2c(csc2-c2cnc(-c3sc(-c4ccco4)c4ccoc34)c3nsnc23)C1	**-2353.809**	-2353.751	-2353.672
d2592089dde92901f48d635d68c14e287717763f	c1ccc(-c2cc3ccc4c5cccnc5ccc4c3[nH]2)nc1	**-934.018**	-933.927	-933.829
1a9cbf38df65acbb706d54a69f6274424cb1ab8d	c1cc(-c2c(-c3scnc3-c3cncc4nsnc34)ccc3c[nH]cc23)c[nH]1	**-1893.895**	-1893.814	-1893.743
6d16614c6a5bc4be540bb184206fc1af5c6a56d0	C1=CC(c2occc2-c2cc3cc[se]c3s2)CC(c2ccc3c(ccc4nccc(-c5cc6nccnc6s5)c43)n2)=C1	**-4799.214**	-4799.151	-4799.056
a8cf110619a27ef5fd358cd1d0f485d9427eb3e9	C1=Cc2c(csc2-c2sc(-c3cncc4nsnc34)c3ccoc23)C1	**-2125.01**	-2124.966	-2124.863
49aa069940e471514ae968e21955d1dcd5ee1e0d	c1ccc2c(c1)ccc1[nH]cc(-c3cncc4nsnc34)c12	**-1270.792**	-1270.727	-1270.627
f3f56f9df3b56b56f55564504f65fcbfd39a746c	C1=C(c2scc3occc23)Cc2c1c1c(c3c2c2c(c4cc[se]c43)=CCC=2)=C[SiH2]C=1	**-4164.309**	-4164.289	-4164.212
0a9a29712af0fa1c0074bb2bf5dc02615172e836	C1=Cc2csc(-c3[nH]cc4c3sc3c5cccnc5ccc43)c2C1	**-1674.935**	-1674.846	-1674.754
5b69debcab20f383091d432db5954f9cacaf0a50	c1cc2c(cn1)c1cscc1c1cnc3[nH]cc(-c4cc5ccsc5s4)c3c21	**-2203.447**	-2203.36	-2203.263
e517b75a17d330c1d68371dae3d2d178b2c9d515	c1cc2cc(-c3nccc4ccsc34)ccc2cn1	**-1123.213**	-1123.123	-1123.01
488d1ff72f6be03e1b4700792c3f96465fe48dc3	c1ccc(-c2nccc3c4oc5occc5c4c4cc5sccc5cc4c23)nc1	**-1579.595**	-1579.507	-1579.424
0bfadbeb4d28675ac1de0ada05c091f8d958d96d	C1=Cc2csc(-c3cc4cc5ccc6ncccc6c5cc4o3)c2C1	**-1374.056**	-1373.98	-1373.88
cb31741b39bd869863e44c127238c18d6aafdd25	C1=Cc2csc(-c3c4cocc4cc4c3[se]c3oc5occc5c34)c2[SiH2]1	**-4082.572**	-4082.514	-4082.421
9e35cac88c6cfab2a4e65ff9dd5b41075014ed80	C1=Cc2c(-c3sccc3-c3sccc3-c3cc4c(s3)[SiH2]C=C4)sc(C3=CC=C4c5cc[se]c5SC4C3)c2C1	**-5874.623**	-5874.548	-5874.501
9fd3b1b689f45d2387c593950b7ffd0de72b1ca5	c1cc2c(cn1)cnc1c2ccc2nccc(-c3cc4ccsc4s3)c21	**-1767.192**	-1767.109	-1767.014
04058245832acce4a0a46f5b080821a541c74a14	C1=Cc2c(cnc3ccc4c(-c5cccc6cscc56)csc4c23)[SiH2]1	**-1948.357**	-1948.279	-1948.209
f2d39ead8928386ccbba494e245657a390133d0e	c1ccc(-c2ccc3cc4c(cc3c2)c2cocc2c2cc[nH]c24)nc1	**-1069.358**	-1069.279	-1069.188
ac3e3296f59d061b87690a391695d040cf340c92	C1=Cc2csc(-c3c4c[nH]cc4cc4c3[se]c3[se]c5[se]ccc5c34)c2[SiH2]1	**-8715.215**	-8715.148	-8715.056
3fb0c2dacc4da4d384487ac6c9685b740e3cdeb9	c1ccc(-c2cccc3c2ccc2c4ccccc4oc32)nc1	**-937.831**	-937.741	-937.621
782179af12ccbece2a906390eb434bb8202ed931	c1cc2c(cn1)cnc1ncc3c4cc(-c5nccs5)cnc4ccc3c12	**-1478.541**	-1478.455	-1478.359
d65f27823c7ecdea4e402f905c6386c91b972dc0	C1=C(c2ccccc2)[SiH2]c2c1sc1c2cnc2ccc3ccccc3c21	**-1627.622**	-1627.564	-1627.451
a2634d68d4d69ffdae7ce1ba719a0521bb7a2a3c	C1=Cc2csc(-c3c4cocc4cc4c3[nH]c3sc5[nH]ccc5c34)c2[SiH2]1	**-2039.571**	-2039.519	-2039.427
9102221874344fa8071112d7dc90f98786c65518	C1=Cc2c(oc3[nH]c4c(-c5scc6c5CC=C6)c5nsnc5cc4c23)C1	**-1804.229**	-1804.171	-1804.072
77a76e46d25ed2836dfc4a1872d0911ab85fe733	C1=Cc2c(-c3ncncn3)sc(-c3c(-c4ccc[nH]4)ccc4ccccc34)c2[SiH2]1	**-1792.431**	-1792.348	-1792.267
10b2f019d06e3a0b7f1a2506d3154eb03739f7cd	c1cc(-c2nccc3nsnc23)c2ccc3c4ccncc4cnc3c2c1	**-1478.536**	-1478.464	-1478.365
5a5c14ca1023f866d0e2965199e70eeae2d1dcfc	C1=c2ccc3cc4[se]c5cc(-c6cccs6)c6ccccc6c5c4cc3c2=CC1	**-3837.821**	-3837.762	-3837.688
ecc292697a2aa20d75469006034b99e23fe852b1	C1=Cc2csc(-c3[se]ccc3-c3[nH]ccc3-c3cccc4c[nH]cc34)c2[SiH2]1	**-4046.257**	-4046.179	-4046.103
ef80545efb1d7ceda6fbcad776d98d5f7f06e5b4	c1ccc(-c2ncc3c4[nH]c5[se]ccc5c4c4cc5occc5cc4c23)nc1	**-3563.029**	-3562.936	-3562.852
2821a62a151d6c28fff11dad4315b4051a8c3c54	C1=Cc2c(-c3cccc4c3=CCC=4)sc(-c3sccc3-c3nccs3)c2[SiH2]1	**-2385.749**	-2385.709	-2385.634
239975947bc1d49f79401a21a6713b6bf7be6484	C1=Cc2c(cnc3c4c(c5c(c23)C(c2cccs2)=C[SiH2]5)=CCC=4)C1	**-1551.33**	-1551.287	-1551.193
a8b979111c91e0234527a034d3c51ff23e181035	c1cc2ccc3c(ccc4scc(-c5nccc6nsnc56)c43)c2cn1	**-1783.243**	-1783.171	-1783.073
2debb65c2b9d955f9191b79da5357199bd23bf99	c1ccc(-c2[nH]cc3ccc4cc5cnccc5cc4c23)nc1	**-933.996**	-933.917	-933.837
190984cc7ac57bd5c0f5c727a012a90def6fb98d	c1ccc(-c2[nH]cc3ccc4c5ccsc5ccc4c23)nc1	**-1238.707**	-1238.627	-1238.529
e222e28fb5a3ad90964eaa4af90a28d2352a216d	C1=c2ccc3cc4oc5cc(-c6ccccc6)sc5c4cc3c2=CC1	**-1357.988**	-1357.93	-1357.854
b7524000bcdc6f4efdd90a4ea2e17ccb2caad55d	C1=CC(c2nccs2)CC(C2=C(c3scc4c3CC=C4)Cc3[se]c4c([nH]c5c6c(ccc54)=C[SiH2]C=6)c32)=C1	**-4790.818**	-4790.783	-4790.723
604f33c37f6477f9765d89e7498624100b290c4f	c1ccc2c(c1)cc(-c1ccc3cnccc3c1)c1cncc12	**-956.053**	-955.963	-955.859
ca14174f48a96f8aa47b3e2009fa8c9fb6a990fc	c1ccc(-c2cc3c(cnc4c5cccnc5ccc34)[nH]2)nc1	**-950.065**	-949.974	-949.862
a7d4ac489f9b7f580da9137c976f2e286de4e341	C1=c2ccc3[se]c4c5sc(-c6scc7[nH]ccc67)cc5sc4c3c2=C[SiH2]1	**-4708.622**	-4708.589	-4708.526
859f5ceb74a46bfed9ddb2d5def9480437d7ff82	c1ccc(-c2cnccc3ccc4ccc5occc5c4c23)nc1	**-953.848**	-953.748	-953.66
bdf318c3c633b5cc835423c89606eb65d8d06cda	c1ccc(-c2cc3c(s2)sc2c4cc5ccccc5cc4ccc32)nc1	**-1735.124**	-1735.057	-1734.971
ae7da14e63e341c6968a4f489029a5e543a6b778	c1ccc(-c2coc3cc4cnccc4cc23)c2cocc2c1	**-935.605**	-935.52	-935.451
edbe27699ed1b2b7a1faf691751af63920a05c1d	c1ccc(-c2cc3nsnc3c3cc4c(cc23)oc2ncccc24)nc1	**-1460.323**	-1460.243	-1460.139
4b263a4ac076ecff73f854b10faf96612722f6bc	C1=Cc2c(c3[nH]c4cc(-c5scc6[nH]ccc56)c5c[nH]cc5c4c3c3nsnc23)C1	**-1954.062**	-1954.006	-1953.918
adf719b366a637a837b44cdf876acd3b2be0599e	Cc1cncc(-c2ccc3cscc3c2-c2scc3c2CC=C3)c1	**-1660.065**	-1659.996	-1659.868
ed8b701ecf44c148c7af507406982e157515ea05	c1ccc(-c2scc3c2[se]c2c4cc5ccccc5cc4ccc32)nc1	**-3738.411**	-3738.346	-3738.259
ae970a4c591158c97ee34d1a624112db181e5aad	c1cc(-c2cncc3cnccc23)c2c(c1)ccc1ccncc12	**-972.083**	-971.992	-971.893
e68ba914eab011a226973a9ae52fb31a98d8ded2	C1=Cc2csc(-c3cc4cc5ccc6sccc6c5cc4o3)c2C1	**-1678.768**	-1678.692	-1678.593
bf33ff51904973a2654f8cabd8f51c30e69f7091	c1ccc2c(-c3ccc4ccncc4n3)cncc2c1	**-818.496**	-818.407	-818.3
3ae9f6262ec9ffd788347072c919b15a87cdce3b	C1=Cc2csc(-c3c4nsnc4cc4c3oc3nc5[se]ccc5cc34)c2C1	**-4204.541**	-4204.481	-4204.374
285208d821ff6672e175eb2f86ad3a7ef02937ae	c1ccc(-c2nccc3c2[se]c2cnc4ccsc4c23)nc1	**-3616.916**	-3616.82	-3616.706
37e50301c602d97862b2922753003be80a477a65	c1ccc(-c2[nH]cc3ccc4c5ccncc5ncc4c23)nc1	**-950.042**	-949.963	-949.86
ba91a07f038251055b51b896ebb386bb603f09e3	C1=Cc2c(-c3ccc(-c4scc5c4C=C[SiH2]5)cc3)sc(-c3cccnc3)c2C1	**-2065.021**	-2064.951	-2064.853
73ca42d448f3dff1931da2d2ff7ab074b21d237a	c1ccc(-c2cncc3ccc4ccc5[se]ccc5c4c23)nc1	**-3280.081**	-3279.995	-3279.898
be214f7666b71c8e4ff01ebe6deafec0b99e0410	C1=CC2C=C[Se]C2C(c2ccc3cnccc3c2)=C1	**-3111.591**	-3111.525	-3111.421
27abef5c4154f60f0c658a413ba008f7053c2abc	C1=C(c2nccc3nsnc23)[SiH2]c2c1c1c(c3c2c2c(c4cc[nH]c43)=C[SiH2]C=2)=CCC=1	**-2119.924**	-2119.9	-2119.839
675f7e4034ed552aed8ea12c32593b17df7f34de	C1=Cc2c(-c3ccc(-c4scc5sccc45)s3)sc(-c3cncc4nsnc34)c2C1	**-2999.708**	-2999.661	-2999.583
6cb2c1d7b7b33ece46c4ed0ebd576593787486d6	c1ccc2c(c1)c1cnccc1c1c3cnccc3ncc21	**-894.699**	-894.609	-894.494
a8303cb28953a6d4c2d58196b510ad724ef8fff2	Cc1cccc2c1ccc1c2ncc2c3c[nH]cc3c3cnccc3c21	**-1049.503**	-1049.412	-1049.314

Notes and references

1 D. Weininger, *Journal of Chemical Information and Computer Sciences*, 1988, **28**, 31–36.

2 N. O'Boyle, M. Banck, C. James, C. Morley, T. Vandermeersch and G. Hutchison, *Journal of cheminformatics*, 2011, **3**, 33.

3 N. Yoshikawa and G. Hutchison, *Journal of Cheminformatics*, 2019, **11**,.

4 J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. Brockway and A. Aspuru-Guzik, *The Journal of Physical Chemistry Letters*, 2011, **2**, 2241–2251.