Deep generative models enable navigation in sparsely populated chemical space

Michael A. Skinnider[1*], **R. Greg Stacey**[1], **David S. Wishart**[2,3,4,5], **and Leonard J. Foster**[1,6*]

[1] Michael Smith Laboratories, University of British Columbia, Vancouver, British Columbia, Canada
[2] Department of Biological Sciences, University of Alberta, Edmonton, AB, Canada
[3] Department of Computing Science, University of Alberta, Edmonton, AB, Canada
[4] Department of Laboratory Medicine and Pathology, University of Alberta, Edmonton, AB, Canada
[5] Faculty of Pharmacy and Pharmaceutical Sciences, University of Alberta, Edmonton, AB, Canada
[6] Department of Biochemistry and Molecular Biology, University of British Columbia, Vancouver, BC, Canada
[*] email: michael.skinnider@msl.ubc.ca, foster@msl.ubc.ca

Deep generative models are powerful tools for the exploration of chemical space, enabling the on-demand generation of molecules with desired physical, chemical, or biological properties. However, these models are typically thought to require training datasets comprising hundreds of thousands, or even millions, of molecules. This perception limits the application of deep generative models in regions of chemical space populated by only a small number of examples. Here, we systematically evaluate and optimize generative models of molecules for low-data settings. We carry out a series of systematic benchmarks, training more than 5,000 deep generative models and evaluating over 2.6 billion generated molecules. We find that robust models can be learned from far fewer examples than has been widely assumed. We further identify strategies that dramatically reduce the number of molecules required to learn a model of equivalent quality, and demonstrate the application of these principles by learning models of chemical structures found in bacterial, plant, and fungal metabolomes. The structure of our experiments also allows us to benchmark the metrics used to evaluate generative models themselves. We find that many of the most widely used metrics in the field fail to capture model quality, but identify a subset of well-behaved metrics that provide a sound basis for model development. Collectively, our work provides a foundation for directly learning generative models in sparsely populated regions of chemical space.

Chemical space is vast. The number of small, synthetically-accessible organic molecules alone exceeds 10^{60} (ref.[1]). Humans have explored only infinitesimal regions of this vast space over the course of recorded history. Yet this limited exploration has yielded an arsenal of bioactive small molecules that form the basis for most therapeutic regimens. These successes, against overwhelming odds, lead to optimism that more efficient ways of navigating through chemical space could help address many of the most pressing challenges facing humanity.

Historically, many of the most prominent approaches to chemical space exploration aimed to enumerate the set of molecules comprising an explicitly defined space, often using exhaustive graph theoretical approaches[2–5] or genetic algorithms[6–8]. More recently, deep generative models have emerged as an immensely powerful tool to explore chemical space[9]. These models leverage deep neural networks to learn the chemistries implicitly embedded within a training set of molecules. Once trained, deep generative models are capable of stochastically sampling unseen molecules from the target chemical space. Many of the most successful approaches to generative modeling have exploited the analogies between chemistry and human language[10] by learning to generate textual representations of molecules, commonly in the SMILES (Simplified Molecular Input Line Entry System) format[11] (**Fig. 1a**). This strategy allows practitioners to borrow powerful neural network architectures from the field of natural language processing, known as recurrent neural networks (**Fig. 1b**)[12–19]. Although a plethora of alternative approaches have been proposed, such as learning to generate two-dimensional chemical graphs[20,21] or to assemble molecules from smaller substructures[22], systematic benchmarks have not shown these to outperform recurrent neural network-based models of SMILES strings[23,24].

Deep generative models have attracted intense interest for their potential to generate molecules with arbitrary physicochemical, structural, or biological properties on demand, and thereby solve what has been termed the 'inverse design' problem[25]. A major outstanding challenge, however, is that these models are typically seen to require very large amounts of training data—on the order of hundreds of thousands to millions of molecules[9]. It is very often the case that the chemical space targeted for exploration is simply not populated by a commensurate number of known molecules. For instance, investigators wishing to design novel molecules active against a particular receptor are unlikely to have knowledge of more than a few hundred existing agonists. Similarly, entire categories of naturally occurring molecules—for instance, bacterial terpenoids[26]—may have only a thousand or so representatives. Accordingly, bespoke methods based on reinforcement learning (RL)[14,16,27–29] or transfer learning (TL)[13,15,30–32] have been developed to enable generative modelling in low-data regimes. In these paradigms, models are first 'pre-trained' on a large and generic database of chemical structures, and thereafter undergo a second round of 'fine-tuning' meant to gradually guide them into a more

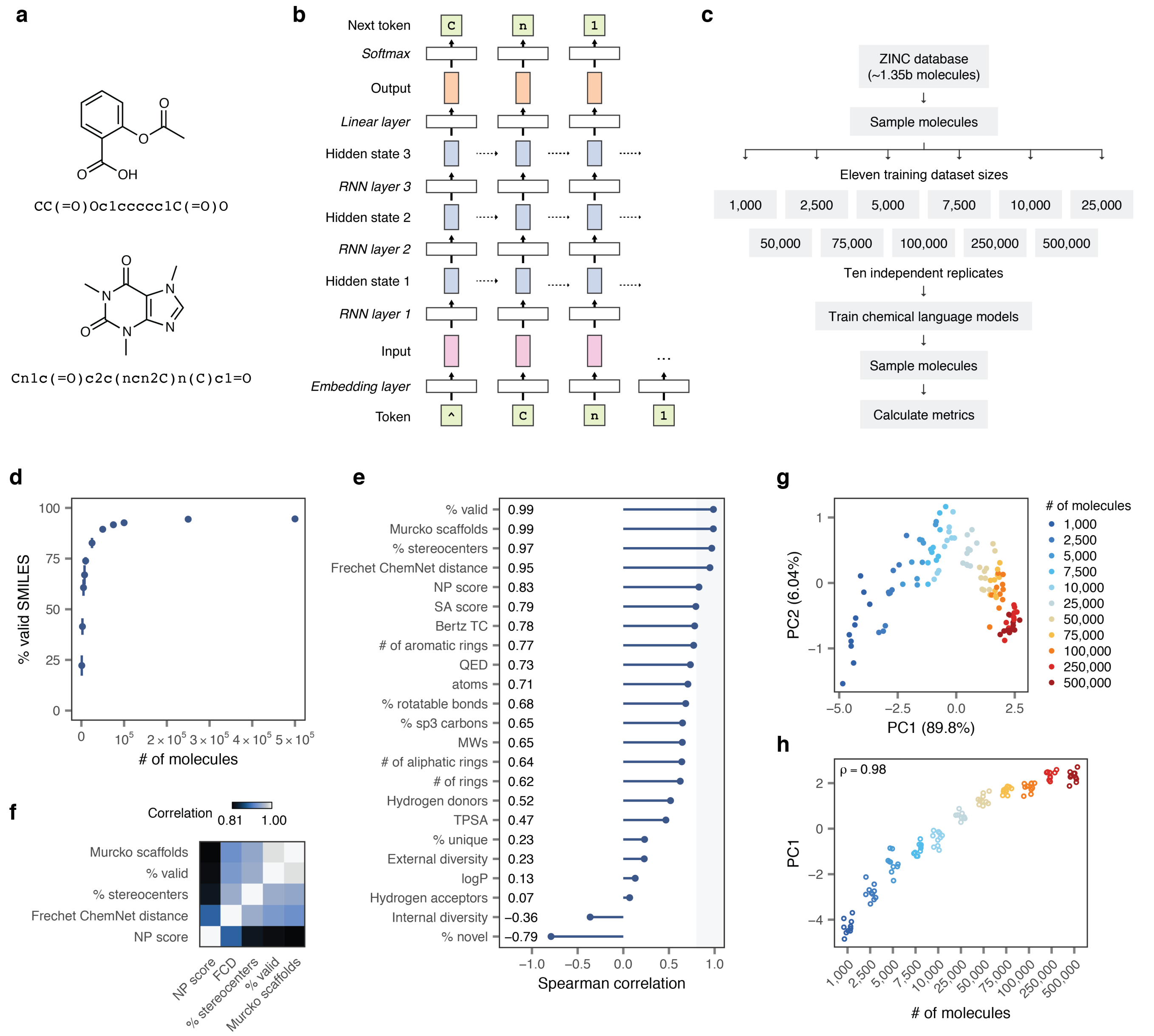


Fig. 1 | Learning generative models of molecules from limited training examples.
a, Molecular structures and canonical SMILES representations of two exemplary molecules, aspirin (top) and caffeine (bottom).
b, Architecture of a three-layer recurrent neural network (RNN) trained to generate SMILES strings.
c, Overview of the experimental design. **d,** Proportion of valid SMILES generated by chemical language models trained on one of varying numbers of molecules sampled from the ZINC database. The mean and standard deviation of ten independent replicates are shown.
e, Spearman correlations between training dataset size (number of molecules) and each of 23 proposed metrics for the evaluation of chemical generative models trained on the ZINC database. Shaded area highlights metrics with a rank correlation ≥ 0.8 to training dataset size.
f, Matrix of Spearman correlations between the values of the five top-performing metrics across $n = 110$ chemical language models.
g, PCA of top-performing metrics for molecules generated by $n = 110$ chemical language models trained on varying numbers of molecules sampled from ZINC, colored by the size of the training dataset.
h, PC1 scores for $n = 110$ chemical language models trained on varying numbers of molecules sampled from ZINC. Inset text shows the Spearman correlation.

restricted chemical space. These approaches, however, have a number of shortcomings. Both RL- and TL-based approaches may suffer from mode collapse, whereby the fine-tuned model loses the ability to generate diverse molecules, or catastrophic forgetting, whereby the general principles of molecule generation learned from the large dataset are forgotten during fine-tuning. In RL-based approaches, the more powerful generative model may learn to exploit unforeseen deficiencies in the reward function, leading to the generation of unrealistically simple but high-scoring molecules[14,33]. Finally, both strategies yield results that vary depending on the duration of the fine-tuning step, and despite some investigation[32], there is no obvious *a priori* basis to infer an optimal duration.

Ideally, it would be possible to directly learn a generative model of a restricted chemical space from a limited number of training examples and bypass the need for RL- or TL-based approaches. At present, however, it is unclear what the lower bound might be on the number of molecules needed to learn a robust generative model, or whether this lower bound might vary as a function of the target chemical space. Moreover, despite some pioneering efforts[18,34], it remains largely unclear whether specific strategies could help optimize generative models for the low-data regime. Such strategies might include varying the textual representation of the input molecules, the architecture or hyperparameters of the recurrent neural network, the process by which the network is trained, or strategies for augmentation.

Here, we systematically evaluate the ability of deep generative models to learn from limited training data. We train generative models on varying numbers of molecules sampled at random from four large chemical databases, allowing us to quantitatively dissect the relationship between the size of the training dataset and the quality of the generative model for the first time. We demonstrate that remarkably robust generative models can be learned from a small number of examples. However, this number varies with the structural complexity of the target chemical space, with fewer examples needed to learn models of simpler molecules. We identify strategies that reduce the minimum amount of training data required to learn a generative model of equivalent quality, most notably including data augmentation by non-canonical SMILES enumeration[35]. Conversely, our systematic benchmarks indicate that many of the strategies that have been proposed in the literature for this purpose are largely ineffective. We demonstrate the application of the principles that emerge from our analysis by training generative models of bacterial, plant, and fungal metabolites that learn to faithfully reproduce highly complex chemical spaces from only thousands of input molecules.

A secondary outcome of our work is that the structure of our experiments provides an opportunity to compare the metrics that are currently used to evaluate generative models themselves. Specifically, in the absence of any other perturbation, we expect that a model trained on 100,000 molecules should essentially always outperform a model trained on only 10,000, which should in turn outperform one trained on only 1,000. We leverage this expectation by comparing 23 metrics proposed for the evaluation of generative models against the experimental ground truth (that is, the number of molecules in the training dataset). Surprisingly, we find that many of the most widely used metrics in the field entirely fail to capture model quality. This observation raises the alarming possibility that relying on these metrics to evaluate generative models has hindered progress. We identify a small subset of metrics that are robustly correlated with the size of the training dataset. However, we also show that relying on any individual metric can lead to problematic conclusions. We develop a holistic framework to integrate multiple orthogonal lines of evidence about model quality, thus providing a sound foundation for model development. Collectively, our analyses chart a path toward directly learning generative models of sparsely populated areas of chemical space.

Results

Deep generative models learn from limited training data. Chemical language models are powerful tools for exploring chemical space, but are generally thought to require very large training datasets—on the order of hundreds of thousands, if not millions of molecules. However, the degree to which this perception is true has not been empirically investigated. We therefore initially set out to determine the minimum number of molecules required to train a robust generative model capable of generating valid and unseen molecules from a target chemical space.

To address this question, we drew random samples of 1,000 to 500,000 molecules from the ZINC database of commercially available compounds[36] (**Fig. 1c**). We then trained a chemical language model on these molecules, as represented by their SMILES strings. After the model had finished training, a total of 500,000 SMILES were sampled from the trained model. To quantify variability in model performance, we repeated this process ten times for each sample size.

As an initial check on the quality of the trained models, we calculated the proportion of valid SMILES strings generated by each model ("% valid"), a metric that has been widely used to evaluate the performance of chemical generative models. To appreciate the relationship between the size of the training dataset and model quality, we plotted this proportion against the number of SMILES strings used to train each model. The proportion of valid molecules increased rapidly as the size of the training dataset increased: from only 6.7% when learning from a dataset of 1,000 molecules, to 69.1% when trained on 25,000 molecules (**Fig. 1d**). Remarkably, we found that trained models were able to generate valid molecules at a rate above 50% with a training dataset of only 5,000 SMILES strings. On the other hand, performance saturated rapidly after approximately 50,000 molecules had been added to the training set. As the size of the training dataset grew from 50,000 to 500,000 molecules, the proportion of valid molecules generated by the models increased by only 5.1%, from 89.5% to 94.6%.

Widely used metrics fail to capture the performance of generative models. Together, these observations suggested that robust generative models of molecules can be learned from surprisingly small training datasets. However, the proportion of valid SMILES captures only one aspect of model performance. If a model has learned to generate valid SMILES strings, but the generated molecules bear little resemblance to those in the training set, then clearly the model has not learned a useful generalization of the target chemical space. We therefore sought to achieve a more holistic evaluation of model performance.

To accomplish this goal, we calculated a suite of 23 different metrics that have previously been proposed for the evaluation of generative models of molecules[18,23,24,34,37–39]. In addition to the proportion of valid SMILES strings, we also computed the proportions of unique and novel molecules generated by the model (**Supplementary Fig. 1a**). We additionally computed 17 different structural or physicochemical properties for each generated molecule, including properties such as the molecular weight, topological complexity[40], or natural product-likeness score[41] (see the Methods for a complete description). We then quantified the similarity of the distributions observed for generated molecules and the training set using the Jensen-Shannon divergence. To specifically assess the diversity of the generated molecules, we calculated the mean Tanimoto coefficient between random pairs of generated molecules, or random pairs of generated and training set molecules, to obtain the internal and external diversities, respectively[37]. Finally, we computed the Fréchet ChemNet distance[38], a metric based on the predicted biological activities of the generated molecules that was developed specifically for the evaluation of chemical generative models.

Collectively, this suite of metrics allowed us to comprehensively survey the methods that have been proposed to evaluate generative models of molecules. In the absence of a 'ground truth', however, it has been unclear which of these metrics most faithfully capture the quality of the underlying generative model. We reasoned that the structure of our experiment imposed a strong expectation on the anticipated outcomes that could be used to ascertain the most useful metrics. Specifically, we reasoned that as the size of the training set increased, so too should measures of model performance. In other words, a model trained on 500,000 molecules should outperform a model trained on only 5,000 molecules, and this difference should be reflected quantitatively in the value of the performance measure. To formalize this notion, we calculated the Spearman rank correlation between the number of SMILES strings in the training dataset and the value of each metric. We then compared the 23 metrics based on their correlations to the size of the training dataset.

Surprisingly, we observed enormous variation in the performance of the 23 previously proposed metrics (**Fig. 1e**). A handful of metrics were strongly correlated to the number of molecules in the training dataset, including the proportion of valid molecules, the Fréchet ChemNet distance, the proportion of stereocenters, and the Murcko scaffolds of the generated molecules (**Supplementary Fig. 1b**). However, the majority were at best moderately correlated to this experimental 'ground truth,' with little guarantee that an increase in the size of the training dataset would produce a consistent change in the value of a given metric (**Supplementary Fig. 1c**). Worryingly, a subset of metrics exhibited no statistically significant correlation at all to the size of the training dataset (**Supplementary Fig. 1d**). Among these were two of the most widely used metrics in the field: the proportion of unique molecules (adjusted p-value = 0.20) and the computed logP of generated molecules (adjusted p-value > 0.99). Our observation that these metrics entirely failed to capture an intervention that dramatically impacted model performance suggests they are ill-suited for the evaluation of chemical generative models. More broadly, the observation that many of the most widely used metrics are at best weakly correlated with model performance raises the possibility that existing models have been optimized to maximize unsound measures of model quality.

Holistic evaluation of chemical generative models. We sought to integrate information from several of the top-performing metrics in order to arrive at a single, holistic measure of model performance. However, the optimal manner by which to accomplish this was initially not clear. Metrics such as the proportion of valid molecules, the Fréchet ChemNet distance, and the Jensen-Shannon divergence of Murcko scaffolds are measured on very different scales (**Supplementary Fig. 1b**), and had a complex correlation structure (**Fig. 1f**). Both of these factors precluded simple approaches, such as simply taking the mean across top-performing metrics.

We reasoned that in the context of this experiment, the size of the training dataset would represent the primary source of variation in the values of these metrics. Consequently, we hypothesized that in a principal component analysis (PCA) of the top-performing metrics, trained models would naturally segregate along the first principal component (PC1) according to the size of the training dataset. This hypothesis was borne out by a PCA of the 110 models trained on samples from the ZINC database. We found these models clearly segregated by the number of molecules in the training dataset along PC1, which explained 89.8% of the variance (**Fig. 1g**). Plotting PC1 against the size of the training dataset recapitulated the expected rapid increase in performance, followed by a plateau (**Fig. 1h**). However, integrating information from multiple metrics revealed that model performance continued to improve above the plateau suggested by the proportion of valid molecules. Instead, PC1 scores continued to increase until approximately ~250,000 molecules had been added to the training dataset. These observations suggest that as the size of the training set increases, chemical language models first learn to produce valid SMILES, and only later learn to match the structural and physicochemical properties of the target molecules. Consequently, integrating multiple distinct sources of information is necessary in order to achieve a holistic evaluation of these generative models.

Taken together, these experiments leverage an experimental 'ground truth' setting to compare metrics that have been proposed for the evaluation of chemical generative models. We found that many of these, including some of the most widely used metrics in the field, are uncorrelated with the ground truth. However, PCA provides a framework to holistically capture model performance by integrating evidence from multiple orthogonal top-performing metrics.

Learning generative models of distinct chemical spaces. Our results thus far have focused on learning generative models of molecules from the ZINC database. We next asked whether the number of molecules required to train a robust generative model would vary as a function of the target chemical space. In particular, we hypothesized that fewer examples would be needed to learn models of simple chemical structures, compared to structurally complex molecules such as natural products.

To test this hypothesis, we repeated our initial experiment, but with molecules sampled from three additional databases of chemical structures: the GDB-13 database, which enumerates all possible small organic molecules containing up to 13 atoms[4]; the ChEMBL database, which contains bioactive small molecules with drug-like properties[42]; and the COCONUT database of natural products[43] (**Fig. 2a**). The molecules contained in each of these databases have distinct structural and physicochemical properties, with molecules from COCONUT generally being the most complex, followed by ChEMBL, ZINC, and GDB (**Fig. 2b**).

We began by comparing the proportion of valid molecules generated by models trained on each of the four databases. This comparison strongly supported our hypothesis that the complexity of the target chemical space determines the minimum number of examples required to learn a robust model (**Fig. 2c**). Models trained on small organic compounds from GDB, for instance, always produced a higher proportion of valid SMILES strings than models trained on an equivalent number of molecules from ZINC. In contrast, even when trained on more than 250,000 molecules, generative models of the COCONUT database never produced valid SMILES at a rate exceeding 82%.

We next asked whether our holistic evaluation framework based on PCA could be applied to models trained on divergent chemical spaces. To evaluate this, we first asked whether each of the 23 metrics exhibited the same relationship to model performance as observed in ZINC. We confirmed that these relationships were highly concordant across chemical spaces (**Fig. 2d**). Remarkably, the same five metrics achieved a rank correlation ≥ 0.8 in all four databases (**Supplementary Fig. 2a**). Conversely, we confirmed that the majority of previously proposed metrics were weakly, inconsistently, or non-significantly correlated to the experimental ground truth (**Supplementary Fig. 2b-c**).

Having established that the same five metrics were strongly associated with the number of training examples in four distinct areas of chemical space, we then performed a combined PCA of all n = 440 generative models based on these metrics. Again, we observed a strong tendency for models to separate along PC1 based on the size of the training dataset (**Fig. 2d**). Interestingly, models also separated by their target chemical space along PC2 (**Fig. 2e**). However, this did not compromise the correlation between PC1 scores and the number of training examples, with a mean rank correlation of 0.97 across the four databases (**Fig. 2f** and **Supplementary Fig. 2d**). We obtained similar results when performing PCA within each database separately, supporting the robustness of the approach (**Supplementary Fig. 3a**). Moreover, we obtained similar results when withholding one database at a time from the PCA, and then projecting the withheld models onto the coordinate basis of the PCA space of the other three databases (**Supplementary Fig. 3b**). This latter finding indicates the loadings learned from a PCA of a diverse set of generative models can be applied to unseen models, and thus supports the notion that the PC1 scores provide a generically applicable measure of model performance.

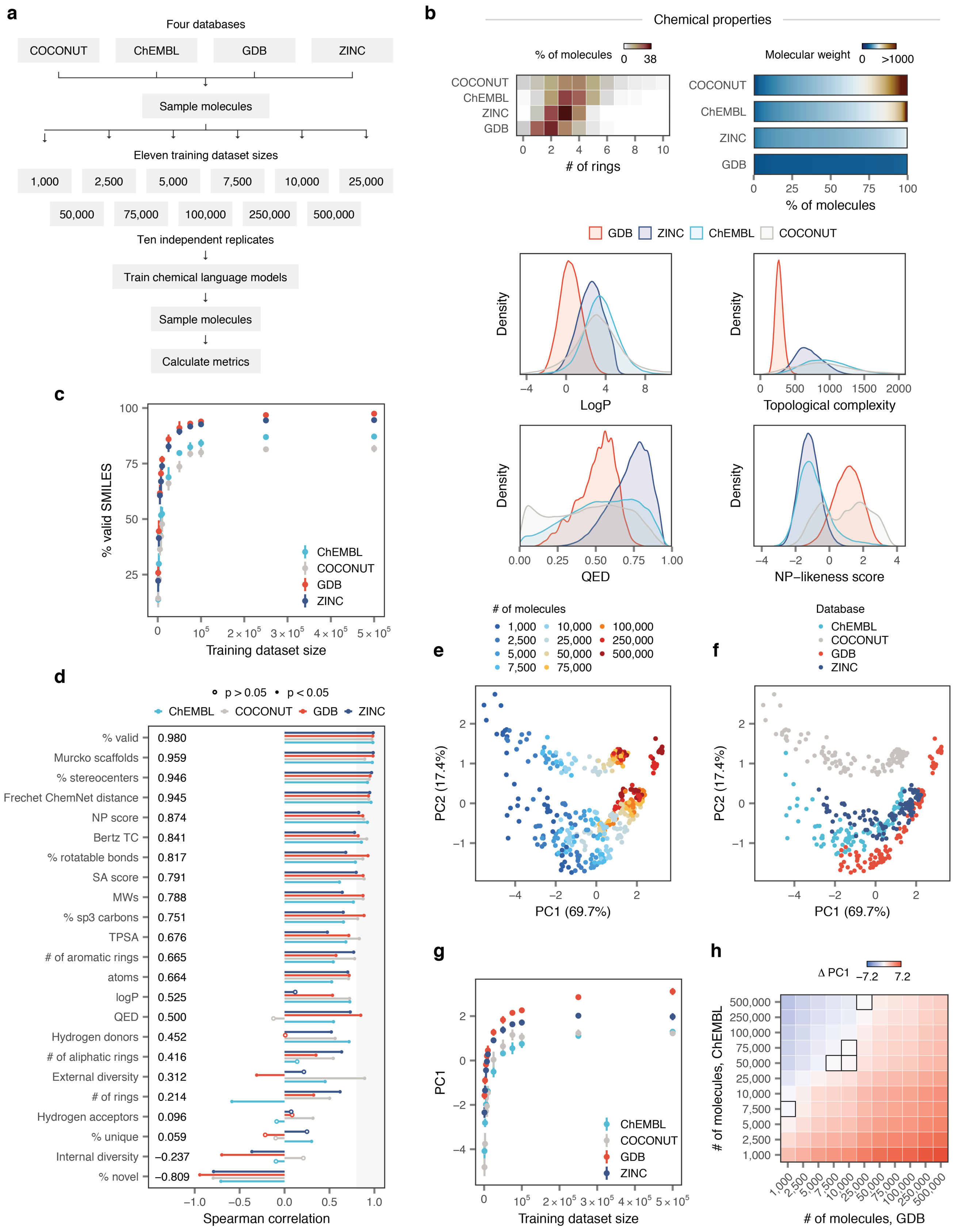
Last, we sought to leverage the results of our PCA analysis to directly compare data requirements across different chemical spaces. We performed statistical comparisons of PC1 scores for models trained on between 1,000 and 500,000 molecules in each pair of databases. These comparisons allowed us to estimate the number of training examples required to learn a generative model of equivalent quality in a second chemical space. This analysis revealed unexpectedly large differences in 'data hungriness' across chemical spaces (**Supplementary Fig. 4**). For instance, a training dataset of 500,000 molecules was required to learn a generative model of ChEMBL that was statistically indistinguishable from a model of the GDB learned from only 25,000 examples (**Fig. 2h**). This observation raises the possibility that results obtained from the analysis of the GDB database may not be generally applicable to learning generative models of more complex molecules[17,46]. More broadly, these results strongly suggest that generative models should be evaluated in more than one chemical space, particularly when they seek to optimize learning from limited training examples.

Evaluating molecular representations for low-data generative modelling. Our results thus far have shown that the minimum number of molecules required to learn a robust generative model depends strongly on the target chemical space. Specifically, as the target chemical space grows more structurally complex, the minimum number of required training examples increases. We next asked if we could identify strategies to reduce this minimum, and thus learn more accurate generative models from fewer examples.

As a first step, we asked if alternative molecular representations could help a chemical language model learn from limited training data. To date, the SMILES format has been the most common textual representation used to train RNNs. However, this approach forces generative models to learn the syntax of the SMILES format, in addition to the properties of the target chemical space. As a result, generative models trained on SMILES strings often generate a large proportion of invalid molecules, particularly when trained on small datasets (**Fig. 1d**), which some have identified as a key limitation of this format[47–50].

Two prominent alternatives to the SMILES format have been proposed. The DeepSMILES variant introduces two modifications to the SMILES syntax to remove long-term dependencies associated with the representation of rings and branches[47]. These modifications are designed to make the DeepSMILES syntax easier to learn than that of conventional SMILES, and thereby increase the proportion of valid molecules generated. Self-referencing embedded strings (SELFIES) are an entirely different molecular representation based on a Chomsky type-2 grammar, in which every SELFIES string specifies a valid chemical graph[48]. The impact of either representation on generative modelling in the low-data regime is not well-understood, and to date, the arguments supporting either representation have primarily been theoretical rather than empirical.

To explore the impact of alternative textual repre-



sentations, we trained generative models on SMILES, DeepSMILES, and SELFIES representations of identical samples from the ChEMBL, COCONUT, GDB, and ZINC databases (**Fig. 3a**). Inspecting the proportion of valid molecules confirmed that models trained on SELFIES strings did indeed produce valid chemical graphs at a much higher

Fig. 2 | Low-data generative models of distinct chemical spaces.
a, Overview of the experimental design.
b, Structural and physicochemical properties of molecules from the four chemical databases analyzed in this study. Top left, number of rings per molecule. Top right, molecular weight spectrum of molecules from each database. Center left, octanol-water partition coefficients (logP)[44]. Center right, Bertz topological complexities[40] of each molecule. Bottom left, quantitative estimate of drug-likeness (QED) scores[45]. Bottom right, natural product (NP)-likeness scores[41].
c, Proportion of valid SMILES generated by chemical language models trained on one of varying numbers of molecules sampled from one of four chemical databases. The mean and standard deviation of ten independent replicates are shown.
d, Spearman correlations between training dataset size (number of molecules) and each of 23 proposed metrics for the evaluation of chemical generative models in four chemical databases. Text shows the mean Spearman correlation. **e,** PCA of top-performing metrics for molecules generated by n = 440 chemical language models, trained on molecules sampled from four different databases, colored by the size of the training dataset.
f, As in **e**, but colored by the chemical database on which the generative models were trained.
g, PC1 scores for chemical language models trained on varying numbers of molecules sampled from one of four chemical databases. The mean and standard deviation of ten independent replicates are shown.
h, Mean difference in PC1 scores ($\triangle$PC1 = PC1_{GDB} – $\text{PC1}_{\text{ChEMBL}}$) between chemical language models trained on varying numbers of molecules sampled from GDB, x-axis, or ChEMBL, y-axis. Dark squares indicate pairs without statistically significant differences (uncorrected p > 0.05, two-sided t-test).

rate than the other two representations (>99.9%; **Fig. 3a** and **Supplementary Fig. 5a**). Surprisingly, models trained on DeepSMILES did not produce valid molecules at a substantially higher rate than ones trained on canonical SMILES. Thus, the proposed modifications to the SMILES syntax, though theoretically grounded, do not appear to empirically improve the robustness of chemical generative models.

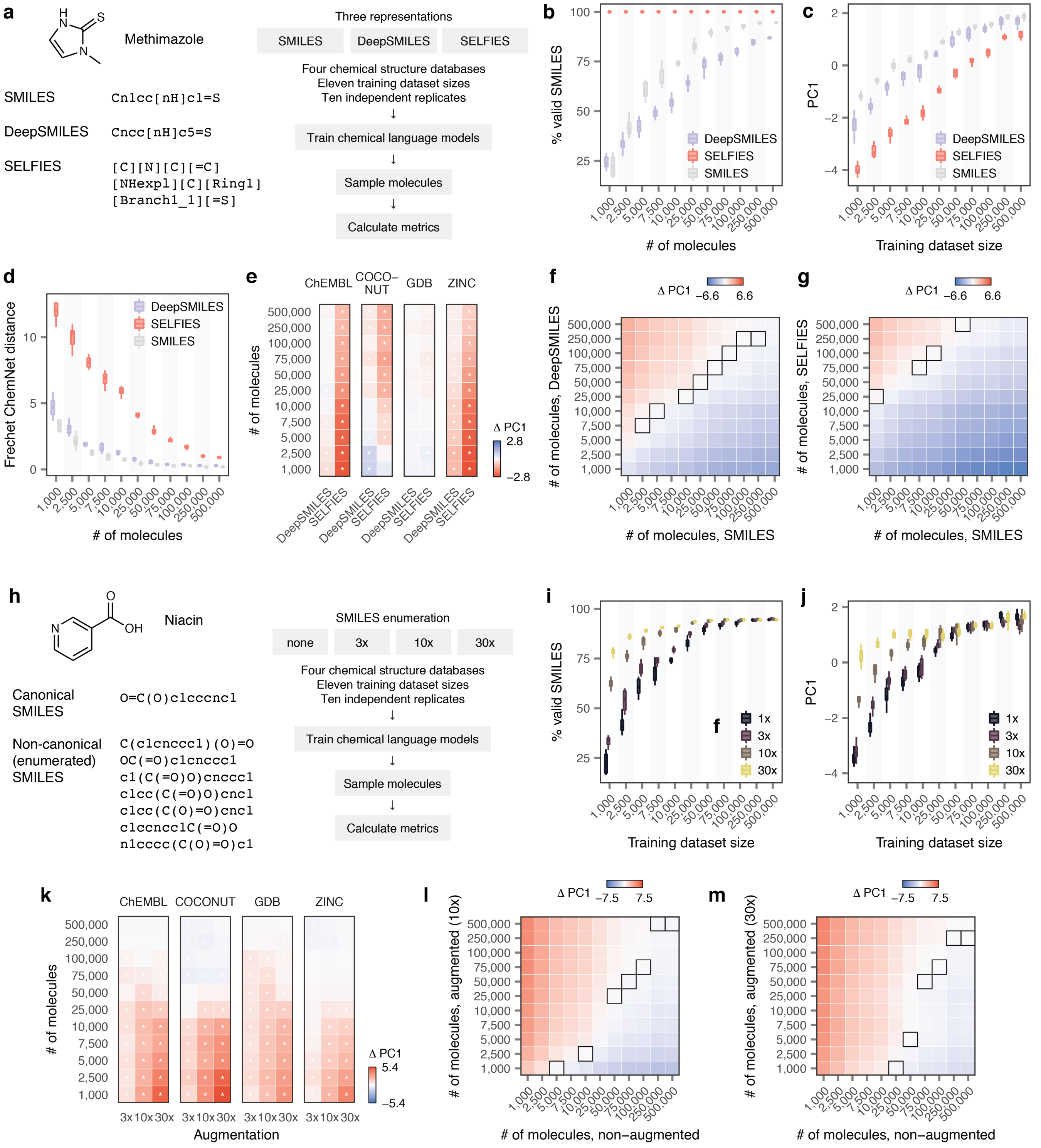
To investigate how well models trained on each textual representation learned the structural and physicochemical properties of the target chemical space, and not just their respective syntaxes, we again performed PCA (**Supplementary Fig. 5b**). Surprisingly, we found that models trained to generate SELFIES strings consistently achieved lower PC1 scores than models trained on SMILES or DeepSMILES representations of the same molecules (**Fig. 5c** and **Supplementary Fig. 5c**). Inspecting individual metrics corroborated this trend; for instance, models trained on SELFIES also had a higher Fréchet ChemNet distance to the training set (**Fig. 5d** and **Supplementary Fig. 5d**). For some very small sample sizes ($n \geq 5{,}000$), models trained on SELFIES or DeepSMILES did occasionally achieve higher PC1 scores (**Fig. 5e**), but these differences were modest, marginally significant, and not consistent across chemical spaces. The net result was that substantially more DeepSMILES or SELFIES were required to learn a model of equivalent quality to one trained on SMILES strings (**Fig. 5f-g** and **Supplementary Fig. 5e**).

Although the tendency of generative molecules trained on SMILES strings to produce invalid outputs has been seen as a central limitation of these models, our results suggest that this may actually represent an unrecognized strength. After filtering out these invalid molecules with a simple and fast post-processing step, generative models trained on SMILES more closely mimicked the structural and physicochemical properties of the target chemical space than models trained on alternative representations. Our results thus suggest that recently proposed representations are not more conducive to learning generative models of molecules than the longstanding SMILES representation, and therefore that SMILES strings should remain the representation of choice for chemical language models.

Paradoxical effects of data augmentation on chemical generative models. Our experiments demonstrated that alternatives to the SMILES representation did not improve the robustness of generative models trained on small datasets. A related concept is to perform data augmentation by varying the SMILES representation itself[35]. By convention, each chemical structure possesses a unique SMILES representation that is typically referred to as its ‘canonical’ SMILES. However, depending on the order in which the atoms in the molecule are traversed, hundreds or thousands of ‘non-canonical’ SMILES representations are also possible (**Fig. 3h**). Enumeration of non-canonical SMILES has been employed to learn continuous representations of chemical structures, by training sequence-to-sequence models[51,52], and emerging evidence suggests that SMILES enumeration can improve the quality of generative models[18,34]. We tested whether SMILES enumeration could decrease the number of training examples needed to learn a robust generative model of molecules.

We trained chemical language models on canonical SMILES representations or on non-canonical SMILES after varying degrees of enumeration (**Fig. 3h**). Models trained on enumerated SMILES generated valid molecules at a dramatically higher rate, especially in the smallest training datasets (**Fig. 3i**). For example, when training models on just 1,000 molecules from the ZINC database, the proportion of valid SMILES improved from 22.2% to 78.4% after enumerating 30 non-canonical SMILES for each canonical SMILES in the training set. We observed consistent patterns across different chemical spaces, with SMILES enumeration consistently lowering the number of examples required to achieve a given proportion of valid SMILES strings (**Supplementary Fig. 6a-c**). The lone exception was for the most structurally complex databases, in which very high degrees of data augmentation sometimes appeared to degrade the quality of models learned from large training datasets (**Supplementary Fig. 6b**).

To corroborate these trends, we again performed PCA using multiple top-performing metrics (**Supplementary Fig. 6d**). This analysis highlighted the context-specific effects of SMILES enumeration (**Fig. 3j** and **Supplementary Fig. 6e**).



In general, data augmentation had by far the largest effect on models learned from very small training datasets. When the training dataset comprised at least ~50,000 molecules, the effects of SMILES enumeration were much more subtle. Moreover, in the largest training datasets, we occasionally observed a negative effect of SMILES enumeration (**Fig. 3k**). Together, these findings suggest that data augmentation is best reserved for the low-data regime, particularly when modelling structurally complex molecules.

To quantify the improvement in performance attributable to SMILES enumeration, we compared models trained on augmented datasets to non-augmented datasets of varying sizes (**Fig. 3l-m**). For very small training datasets, data augmentation by a factor of ten yielded a performance increase on par with quadrupling the number of unique molecules in the training set (**Fig. 3l** and **Supplementary Fig. 6f**). For instance, after data augmentation, models trained on 2,500 molecules from the ZINC database achieved PC1 scores that

Fig. 3 | Alternative molecular representations for low-data generative models.
a, Left, three string-based molecular representations of an example molecule, the thyroperoxidase inhibitor methimazole. Right, overview of the experimental design.
b, Proportion of valid SMILES generated by chemical language models trained on one of three string representations of molecules from the ZINC database.
c, PC1 scores of chemical language models trained on one of three string representations of molecules from the ZINC database.
d, Fréchet ChemNet distances of chemical language models trained on one of three string representations of molecules from the ZINC database.
e, Mean difference in PC1 scores (ΔPC1) between chemical language models trained on matching numbers of SELFIES or DeepSMILES, as compared to SMILES, from one of four chemical databases. Asterisks indicate statistically significant differences (uncorrected $p < 0.05$, two-sided t-test).
f, Mean difference in PC1 scores between chemical language models trained on varying numbers of molecules sampled from ZINC, represented either as DeepSMILES, y-axis, or SMILES, x-axis. Dark squares indicate pairs without statistically significant differences (uncorrected $p > 0.05$, two-sided t-test).
g, As in **f**, but with models trained on SELFIES on the y-axis.
h, Left, canonical SMILES and seven enumerated non-canonical SMILES for an example molecule, the nutrient and cholesterol-lowering agent niacin. Right, overview of the experimental design.
i, Proportion of valid SMILES generated by chemical language models trained on molecules sampled from the ZINC database after varying degrees of non-canonical SMILES enumeration.
j, PC1 scores of chemical language models trained on molecules sampled from the ZINC database after varying degrees of non-canonical SMILES enumeration.
k, Mean ΔPC1 between chemical language models trained on non-canonical SMILES with varying degrees of data augmentation from one of four chemical databases, as compared to canonical SMILES. Asterisks indicate statistically significant differences (uncorrected $p < 0.05$, two-sided t-test).
l, Mean ΔPC1 between chemical language models trained on molecules from the ZINC database represented as canonical SMILES, x-axis, or non-canonical SMILES after 10x augmentation, y-axis. Dark squares indicate pairs without statistically significant differences (uncorrected $p > 0.05$, two-sided t-test).
m, As in **l**, but with an augmentation factor of 30x.

were statistically indistinguishable from a model trained on 10,000 molecules without enumeration. Augmentation by a factor of 30 had even more drastic effects, allowing a model trained on only 5,000 molecules to achieve PC1 scores comparable to one trained on 50,000 SMILES without enumeration (**Fig. 3m**). Conversely, we confirmed quantitatively that the improvement in performance afforded by data augmentation diminished as the size of the training dataset increased, and was attenuated completely when training on a dataset of 500,000 molecules.

Taken together, these analyses highlight the potentially dramatic impacts of data augmentation by SMILES enumeration. When learning generative models from very small training datasets, data augmentation by a factor of 10 can improve performance to a degree equivalent to quadrupling the amount of training data. On the other hand, our experiments expose a previously overlooked potential for 'over-enumeration' when learning models of structurally complex molecules from very large training datasets, whereby even relatively low levels of data augmentation can decrease model performance. Notwithstanding this issue, which to the best of our knowledge has not been previously reported, we suggest that for datasets with less than ~50,000 molecules, SMILES enumeration by a factor of ten can significantly improve the performance of generative models at essentially no cost.

Data, not architecture, dictates model performance in the low-data regime. Our experiments to this point have focused on varying the data provided as input to generative models, including the number of molecules in the training dataset, the chemical space from which those molecules were sampled, and how the molecules are represented in textual form. We next asked whether we could optimize the model itself for the low-data regime. Specifically, we hypothesized that decreasing the total number of neurons in the model, or adding dropout layers[53], could both prevent the model from overfitting to a small number of training examples.

To test this hypothesis, we systematically varied each of six hyperparameters in turn, and evaluated the quality of the resulting models (**Fig. 4a**). These hyperparameters included the sizes of both the embedding and hidden layers, as well as the total number of hidden layers. We also compared different architectures of RNNs altogether, including gated recurrent units (GRU), long short-term memory (LSTM), and 'vanilla' RNNs with two different activation functions (tanh and ReLU). In addition, we experimented with adding varying amounts of dropout between each layer. Finally, to gauge whether the manner by which the models were trained could also affect performance, we varied the size of the minibatches used to train the networks[54]. In order to explore this larger parameter space, we limited our analysis to two of the four chemical databases, ZINC and ChEMBL, and analyzed only five replicates per hyperparameter combination instead of ten.

We performed PCA on a total of 1,210 models trained on the ZINC database (**Fig. 4b**), and observed that models appeared to segregate along PC1 by the size of the training dataset. This observation suggested that the impact of hyperparameter tuning was small, in comparison to the size of the training dataset. To more formally quantify this notion, we plotted the mean PC1 score against the size of the training dataset for each hyperparameter in turn (Fig. 4c). For the parameters that controlled the capacity of the neural network (including hidden layer size, embedding layer size, and the number of hidden layers), intermediate values typically yielded the best performance. However, comparing the mean PC1 score to the total number of neurons in the model emphasized that even for very small or very large models, the difference was small in comparison to the number of molecules

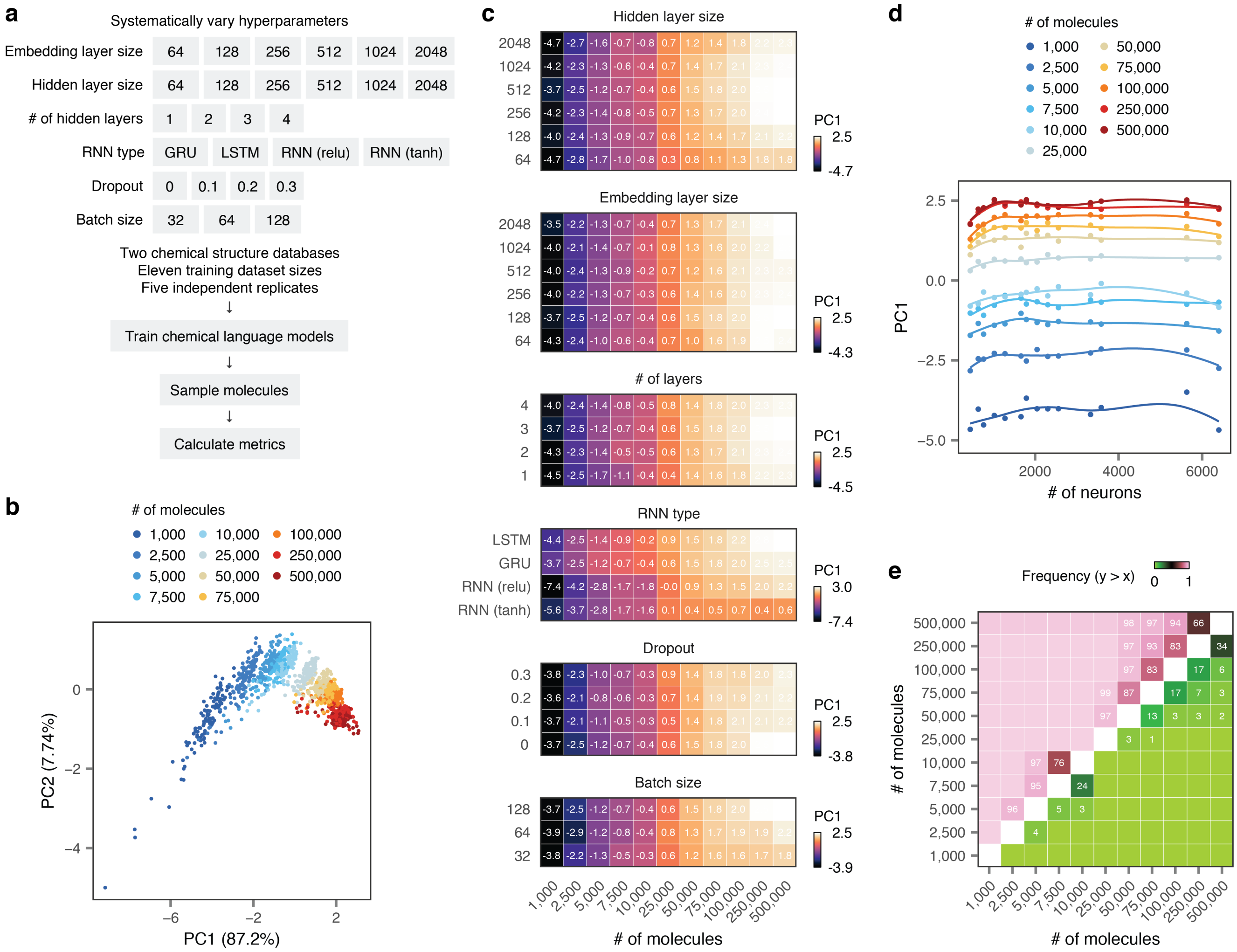


Fig. 4 | Data, not architecture, dictates the performance of low-data generative models.
a, Overview of the experimental design.
b, PCA of top-performing metrics for molecules generated by n = 1,210 chemical language models, trained on varying numbers of molecules from the ZINC database with varying model hyperparameters, colored by the size of the training dataset.
c, Mean PC1 scores for molecules trained on the ZINC database, as a function of both the number of molecules in the training dataset, x-axis, and varying hyperparameters, y-axis. The mean of five independent replicates is shown.
d, Mean PC1 scores of chemical language models as a function of the total number of neurons in the network. Solid lines show local polynomial regression.
e, Proportion of n = 110 chemical language models with varying hyperparameters, trained on the number of molecules shown on the y-axis, that outperformed a model without any hyperparameter tuning trained on the number of molecules shown on the x-axis.

in the training dataset (**Fig. 4d**). The architecture of the RNN had a somewhat greater effect, with GRUs and LSTMs achieving roughly identical performance, but 'vanilla' RNNs performing substantially worse. Neither dropout nor batch size markedly affected on model performance.

Taken together, these results emphasize the primacy of the training dataset on the performance of chemical generative models. We explored a large grid of hyperparameters, but found that in the low-data regime, hyperparameter tuning almost never affected performance to a comparable degree as increasing the size of the training dataset (**Fig. 4e**). Conversely, in larger datasets, hyperparameter tuning appeared to have a correspondingly larger effect. Finally, we observed highly concordant results in a second database, ChEMBL (**Supplementary Fig. 7**). Collectively, these results suggest that optimization of hyperparameters, architecture, or training strategies for generative models is unlikely to provide a fruitful approach to learning generative models of molecules from few examples.

Case study: learning low-data generative models of bacterial, fungal, and plant metabolomes. Our experiments systematically elucidated principles for learning generative models of molecules from limited training data. We sought to exemplify these principles through an illustrative case study. To this end, we aimed to learn generative models of bacterial, fungal, and plant metabolomes. Previous work has shown that manual or semi-automated enumeration of hypothetical metabolites, using rule-based systems, enabled the discovery of novel, bioactive natural molecules using mass spectrometry[55,56]. Generative models of metabolomes could be used to more efficiently traverse metabolite chemical space[57], and thereby facilitate the targeted elucidation of unknown metabolites. However, even for the comparatively well-studied human metabolome, only 100,000 unique molecules are known[58].

We assembled databases of bacterial, fungal, and plant metabolites (Methods), but these comprised only 15,000-22,000 molecules each (**Fig. 5a**). These databases are thus far smaller than those typically used to train generative models of much less structurally complex molecules. With this challenge in mind, we asked whether applying the principles we had elucidated for low-data generative models could allow us to directly model bacterial, fungal, and plant metabolomes, without relying on RL or TL strategies.

We trained chemical language models on bacterial, fungal, and plant metabolites. After experimenting with different molecular representations, data augmentation strategies, and RNN architectures, we selected a LSTM architecture with a high degree of SMILES enumeration as the optimal strategy (**Fig. 5b** and **Supplementary Fig. 8**). We then compared the physicochemical properties of real and generated metabolites. Strikingly, despite the limited amount of training data available, the optimized models generated molecules whose property distributions closely matched those of the three target metabolomes (**Supplementary Fig. 9a-d**). One notable exception concerned the number of rings in the generated molecules (**Supplementary Fig. 9e**). This disparity likely reflects the challenge inherent in learning the SMILES syntax for molecules with many rings, which are denoted by increasingly rare tokens as the number of rings increases[17].

Finally, we sought to visualize and compare the chemical space occupied by real and generated metabolites from each taxonomic group. To achieve this, we first embedded the real and generated metabolites into a continuous space using CDDD (Continuous and Data-Driven Descriptors)[52], then visualized this space in two dimensions using UMAP (Uniform Manifold Approximation and Projection)[59]. To compare the chemical spaces occupied by real and generated metabolites, we visualized the UMAP embeddings with either known metabolites, or a random sample of generated metabolites of equal size, overlaid on top of one another (**Fig. 5c**). The resulting plot demonstrated that the generative models almost perfectly reproduced the chemical space of the three target metabolomes, with very few regions of chemical space occupied exclusively by either real or generated metabolites.

Thus, taken together, these experiments demonstrate that, with optimized strategies, generative models can directly learn to reproduce even very complex chemical spaces from a small number of training examples. The hypothetical metabolites generated by these models may be candidates for targeted identification by mass spectrometry, or even number among the ‘dark matter’ of observed but unidentified metabolites in high-throughput metabolomics[60].

Discussion

Deep generative models have emerged as immensely powerful tools for exploring chemical space. However, these models are widely perceived to require very large training datasets. This perception has prompted the development of bespoke strategies, based on reinforcement learning or transfer learning, to explore chemical spaces populated by few known examples. Here, we set out to quantify the minimum number of molecules required to learn a robust generative model of molecules, and identify strategies to reduce this lower bound. To achieve these goals, we devised a series of systematic benchmarks. In total, we trained over 5,100 generative models, and evaluated more than 2.6 billion molecules sampled from the trained models. The scale of this effort allowed us to comprehensively survey strategies for training and evaluating generative models in the low-data regime. Below, we discuss some of our key findings and their implications for the field.

We initially set out to determine the minimum number of molecules required to train a robust generative model. To answer this question empirically, we trained generative models on varying numbers of molecules sampled from four chemical databases. Although many published models have been trained on millions of molecules, we found that performance began to saturate with as little as ~50,000 examples in the training set. This suggests that it is possible to learn robust generative models from far less training data than has previously been appreciated, without requiring any bespoke strategies. However, we also found model performance to be directly contingent on the chemical space being modelled. Specifically, a larger number of training examples were needed to learn models of structurally complex metabolites. Our observations raise the possibility that results obtained from analysis of the structurally simple GDB database may not generalize to more complex molecules, and, in turn, imply that generative models should be evaluated in more than one chemical space.

We then asked whether we could identify strategies to reduce the amount of training data required to learn a model of equivalent quality. To this end, we first explored the effects of varying the textual format used to represent molecules. Much of the research in the field to date has made use of the SMILES representation to learn language models of chemical structures. However, several perceived shortcomings of the SMILES format have been noted. Foremost among these is the fact that even the best chemical language models typically produce some non-zero proportion of invalid SMILES strings. A second issue is that molecules with many rings are under-represented in model output, a phenomenon that has been attributed to their representation within the SMILES syntax[17]. We evaluated two alternative representations, DeepSMILES and SELFIES, that have been proposed specifically for the setting of chemical language models. Implicit in the design of these representations is the notion that a model’s ability to produce valid molecules is a critical indicator of its quality. However, surprisingly, we found that while models trained on SELFIES strings produced valid molecules

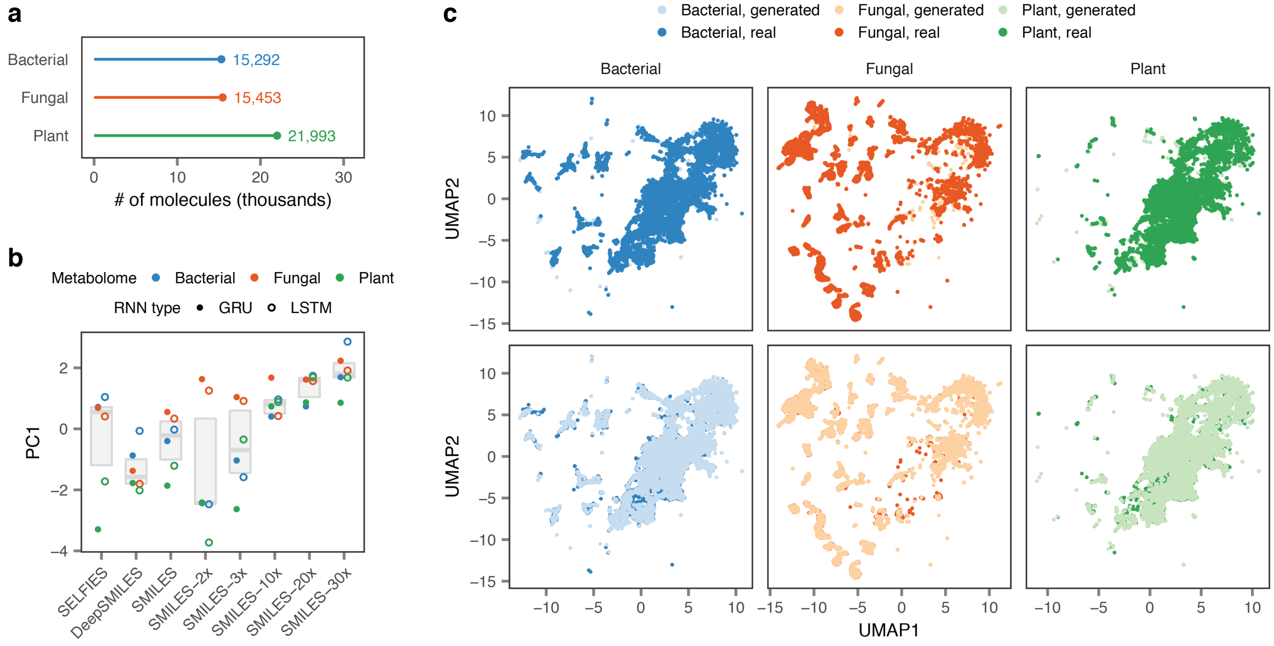


Fig. 5 | Low-data generative models of bacterial, fungal, and plant metabolomes.
a, Number of bacterial, fungal, and plant metabolites used to train chemical language models.
b, PC1 scores of generative models of metabolomes trained with different molecular representations (SMILES, DeepSMILES, or SELFIES), data augmentation strategies (non-canonical SMILES enumeration with an augmentation factor of between 2x and 30x), and RNN architectures (GRU or LSTM).
c, UMAP visualization of the known bacterial, fungal, and plant metabolomes, and an equal number of hypothetical metabolites sampled at random from generative models. Top, real metabolites superimposed over generated metabolites. Bottom, generated metabolites superimposed over real metabolites.

at a near-perfect rate, these molecules failed to match the target chemical space as those generated by a model trained on SMILES. This finding suggests an alternative perspective of the generation of invalid SMILES. In practice, if a model is able to generate many valid molecules that accurately match the target chemical space, then the appearance of invalid SMILES strings is little more than a minor inconvenience, since these can be filtered from model output using a simple post-processing step. On the other hand, while the proportion of valid molecules can be a useful measure of model performance, focusing solely on the validity of the generated molecules (and ignoring their chemical properties) can yield misleading conclusions. Surprisingly, models trained on the DeepSMILES format failed to produce valid molecules at a rate exceeding those trained on SMILES. While the modifications to the SMILES syntax embodied in the DeepSMILES format are theoretically grounded, this finding underscores the primary of empirical evaluation.

By far the most successful strategy we identified to improve generative modelling in the low-data regime involved enumerating multiple non-canonical SMILES for each molecule in the training set. That this form of data augmentation can improve the quality of generative models has previously been noted[18,34], but predominantly in data-rich settings. Our results extend these observations in several important ways. First, the structure of our experiments allows us to precisely quantify the improvement afforded by data augmentation, particularly in comparison to increasing the size of the training dataset. In the low-data setting, for instance, we find that data augmentation by a factor of ten improves model performance to an extent comparable to quadrupling the number of unique molecules. Second, we identify a profound interaction between the effects of data augmentation and the size of the training dataset. We find that data augmentation has the most dramatic effects in the smallest training datasets evaluated here, comprising only 1,000 molecules. Conversely, these effects are greatly attenuated, or even entirely ablated, when training on hundreds of thousands of examples. Third, we expose a risk of 'over-augmentation,' particularly in more structurally complex datasets, that to our knowledge has not previously been noted. This paradoxical effect was discovered only by exploring model performance across multiple distinct regions of chemical space, underscoring the value in this mode of analysis. Notably, in stark contrast to interventions that affected the input data itself, we found that modifying the architecture, hyperparameters, or training strategy of the generative models had little effect. This observation suggests that developing new strategies for molecular representation and data augmentation is likely to present a more fruitful direction for future research than altering the structure of the neural network itself.

The structure of our experiments also allowed us to benchmark the metrics themselves that are used to evaluate generative models. That there is little agreement within the field on how generative models of molecules ought to be evaluated has been noted by several commentators[33,61,62]. The lack of an 'even playing field' for model evaluation hinders comparisons of published models, making it difficult to discern which computational strategies have been successful and which have not. Alarmingly, we found that many of the most widely used metrics in the field were weakly correlated, or even entirely uncorrelated, to our experimental ground truth (that is, the size of the training dataset). This calls into question their use for model evaluation, and raises the worrying possibility that published models have been tuned to optimize dubious measures of performance. However, we identified a smaller subset of metrics that consistently exhibited strong correlations to this ground truth across four distinct chemical spaces. These included both the proportion of valid molecules generated by the model, as well as a series of metrics designed to evaluate the chemical similarity of real and generated molecules. We developed a framework to integrate these metrics using PCA, and showed the resulting metric captured multiple orthogonal sources of information about model performance. The top-performing metrics identified here, and our proposed framework for their integration, collectively provide a sound foundation for model development and evaluation. More broadly, we propose that our experimental design provides a powerful benchmark for the evaluation of generative models, as well as newly proposed metrics themselves.

Collectively, our findings delineate general principles for learning low-data generative models of molecules from a limited number of examples, and develop a rigorous framework for the evaluation of these models. We have made all of the code and data generated in this study publicly available to support future work.

Methods

Input data. Our experiments focused on learning generative models of molecules from four databases of chemical structures: ChEMBL[42], COCONUT[43], GDB[4], and ZINC[36]. Molecules from the ChEMBL database (version 24.1) were obtained from http://ftp.ebi.ac.uk/pub/databases/chembl/ChEMBLdb/releases/chembl_24_1/chembl_24_1_chemreps.txt.gz. Molecules from the COCONUT database were obtained from the Zenodo upload accompanying the original publication at http://zenodo.org/record/3778405/files/COCONUTapril.zip. A random sample of 1 million molecules from the GDB-13 database[17] was obtained from the Reymond group website at http://gdbtools.unibe.ch:8080/cdn/gdb13.1M.freq.ll.smi.gz. A random sample of 1 million molecules was constructed from the ZINC database by first downloading each tranche separately from the ZINC website, then concatenating all 1,669 tranches into a single file and sampling from that file.

For each database, duplicate SMILES and SMILES strings that could not be parsed by the RDKit were removed. Salts or solvents were removed by splitting molecules into fragments and retaining only the heaviest fragment containing at least three heavy atoms, using code adapted from the Mol2vec package[63]. Charged molecules were neutralized using a list of neutralization reactions provided in the RDKit Cookbook. Molecules with atoms other than Br, C, Cl, F, H, I, N, O, P, or S were removed, and molecules were converted to their canonical SMILES representations using the RDKit. Finally, SMILES strings were tokenized, and molecules containing extremely rare tokens (present in less than 0.01% of molecules in the database), as well as SMILES strings longer than 250 characters, were removed. Samples of between 1,000 and 500,000 SMILES were then drawn from the preprocessed databases. SMILES strings were subsequently converted to DeepSMILES[47] or SELFIES[48] using versions 1.0.1 and 1.0.2 of the deepsmiles (http://github.com/baoilleach/deepsmiles) and selfies (http://github.com/aspuru-guzik-group/selfies) packages, respectively. Enumeration of non-canonical SMILES was performed using the SmilesEnumerator class available from http://github.com/EBjerrum/SMILES-enumeration, with augmentation factors of 3, 10, or 30. All of the datasets used in this work are available from Zenodo at http://doi.org/10.5281/zenodo.4419886.

Chemical language models. Recurrent neural networks were trained on samples of 1,000–500,000 molecules from the four chemical structure databases, using code adapted from the REINVENT package (http://github.com/MarcusOlivecrona/REINVENT). SMILES were tokenized by considering individual characters as tokens, except atomic symbols with more than one character (Br, Cl) and environments within square brackets, such as [nH]. SELFIES were tokenized using the split_selfies function from the selfies package. The vocabulary of the RNN then consisted of all unique tokens detected in the training data, as well as start-of-string and end-of-string characters and a padding token. Except where otherwise noted, the architecture of the language models consisted of a three-layer GRU with a hidden layer of 512 dimensions, an embedding layer of 128 dimensions, and no dropout layers. Models were trained using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_1 = 0.999$, with a batch size of 128 (except where otherwise noted) and a learning rate of 0.001, using teacher forcing. 10% of the molecules in the training set were reserved as a validation set and used to perform early stopping with a patience of 50,000 minibatches. After completion of model training, a total of 500,000 strings were sampled from each trained model. All of the code used to train chemical language models is available from GitHub at http://github.com/skinnider/low-data-generative-models.

Evaluating model performance. To quantify the performance of the trained models, we implemented Python source code to calculate a suite of 23 metrics that have previously been proposed for the evaluation of generative models of molecules. These metrics were as follows:

- The proportion of valid molecules generated by the model, where "valid" molecules are those that can be parsed by the RDKit ("% valid").
- The proportion of novel molecules (that is, molecules not found in the training set) generated by the model ("% novel").
- The proportion of unique molecules generated by the model ("% unique").
- The internal diversity[37], defined as the mean Tanimoto coefficient between all pairs of molecules generated by the model. Extended connectivity fingerprints[64] with a diameter of 3 and a length of 1,024 bits were used as input to the calculation of the Tanimoto coefficient. Because calculating the entire matrix of Tanimoto coefficients is prohibitive for very large numbers of molecules, a random sample of 10,000 pairs of molecules was analyzed.
- The external diversity[37], defined as the mean Tanimoto coefficient between all pairs comprising one molecule generated by the model and one molecule from the training set. Again, a random sample of 10,000 pairs of molecules was analyzed rather than computing the entire matrix of Tanimoto coefficients.
- The Fréchet ChemNet distance[38] between the training and generated molecules ("FCD"). The PyTorch implementation available from http://github.com/insilicomedicine/fcd_torch was used to calculate the FCD.
- The Jensen-Shannon divergences between the distributions of 17 structural or physicochemical properties, comparing molecules generated by the chemical language model to the molecules comprising the training dataset. These properties, and their abbreviations used in the figures, were as follows:
  - The number of aliphatic rings in each molecule ("# of aliphatic rings")
  - The number of aromatic rings in each molecule ("# of aromatic rings")
  - The total number of rings in each molecule ("# of rings")
  - The proportion of rotatable bonds in each molecule ("% rotatable bonds")
  - The proportion of carbon atoms in each molecule that are sp3 hybridized ("% sp3 carbons")
  - The proportion of atoms in each molecule that were stereocenters ("% stereocenters")
  - The total proportions of each heavy atom across all molecules in the dataset ("atoms")
  - The topological complexity[40] of each molecule ("Bertz TC")
  - The number of hydrogen acceptors in each molecule ("# of hydrogen acceptors")
  - The number of hydrogen donors in each molecule ("# of hydrogen donors")

- The calculated partition coefficient[44] of each molecule ("logP")
- The frequencies of Murcko scaffolds[65] of all molecules in the dataset ("Murcko scaffolds")
- The molecular weight of each molecule ("MWs")
- The natural product-likeness score[41] for each molecule ("NP score")
- The quantitative estimate of drug-likeness (QED) score[45] for each molecule ("QED")
- The synthetic accessibility (SA) score[66] ("SA score")
- The topological polar surface area[67] of each molecule.

In addition to the Jensen-Shannon distance, we also benchmarked two other measures of differences between property distributions, the Wasserstein distance and Kullback-Leibler divergence, but found JSD was most strongly correlated to experimental ground truth (that is, the size of the training dataset) (**Supplementary Fig. 10**).

Code used to compute all 23 metrics is available from GitHub at http://github.com/skinnider/low-data-generative-models.

Despite the large number of metrics that have been proposed for the evaluation of generative models of molecules, there is little consensus on which should be used to gauge model quality. We initially evaluated the utility of these metrics themselves by correlating the values of each of the 23 metrics to the size of the training dataset, using the Spearman rank correlation to allow for non-linear relationships. We reasoned that because increasing the size of the training dataset from 1,000 to 500,000 molecules would be expected a priori to have a dramatic effect on the performance of a generative model, this analysis could allow us to benchmark the metrics themselves that have been proposed for model evaluation. Five metrics consistently achieved a Spearman correlation ≥ 0.80 to the size of the training dataset in four different chemical databases (% valid, FCD, % stereocenters, Murcko, and NP score). To combine information from all five top-performing metrics, while accounting for the covariance between metrics, we performed PCA on the centered and scaled matrix using the R function 'princomp'. The loadings of each model on the first principal component, PC1, were used for model evaluation. To ensure that these scores accurately captured model performance, we additionally inspected and visualized the proportion of valid molecules generated by each model. Pairwise comparisons of models trained with different input data or different hyperparameters were performed using a two-tailed t-test. The complete set of outcomes calculated for all 5,108 chemical language models analyzed in this study is provided as **Supplementary Table 1**.

Generative models of metabolomes. To train generative models of bacterial, fungal, and plant metabolomes, we compiled databases of known metabolites from the following sources. Bacterial metabolites were assembled from the *E. coli* Metabolome Database (ECMDB)[68], the *P. aeruginosa* Metabolome Database (PAMDB)[69], StreptomeDB[70], NPASS[71], and BioCyc[72]; for the latter two databases, only molecules linked to a bacterial producing organism were retained. Plant metabolites were assembled from the Phenol-Explorer[73], PhytoHub (http://phytohub.eu/), NPASS, and BioCyc databases (keeping only metabolites linked to a plant producing organism in the latter two cases). Fungal metabolites were obtained from the Yeast Metabolome Database (YMDB)[74]. We then trained a total of 48 chemical generative models on the three metabolomes. In addition to the input metabolome, we varied the RNN model (comparing LSTM and GRU architectures), the representation (comparing SMILES, DeepSMILES, and SELFIES), and performed varying degrees of non-canonical SMILES enumeration (with augmentation factors of 2x, 3x, 10x, 20x, or 30x). After inspecting the PC1 scores of all 48 models, as well as the values of individual metrics, we selected the three LSTM networks trained on non-canonical SMILES with the highest augmentation factor for further analysis. To visualize the global chemical space of the real and generated molecules, we computed a continuous, 512-dimensional representation of each molecule using the CDDD package[52] (available from http://github.com/jrwnter/cddd). We then sampled a matching number of real and generated metabolites, and embedded real and generated molecules from all three metabolomes into two dimensions using UMAP[59] (as implemented in the R package 'uwot'), with the following parameters: n_neighbors = 50, alpha = 2, and beta = 1.

Data availability. Input datasets used to train chemical language models are available from Zenodo at http://doi.org/10.5281/zenodo.4419886. Calculated metrics for all 5,108 models discussed in this study are provided as **Supplementary Table 1**.

Code availability. Code used to train and evaluate chemical language models is available from GitHub at http://github.com/skinnider/low-data-generative-models.

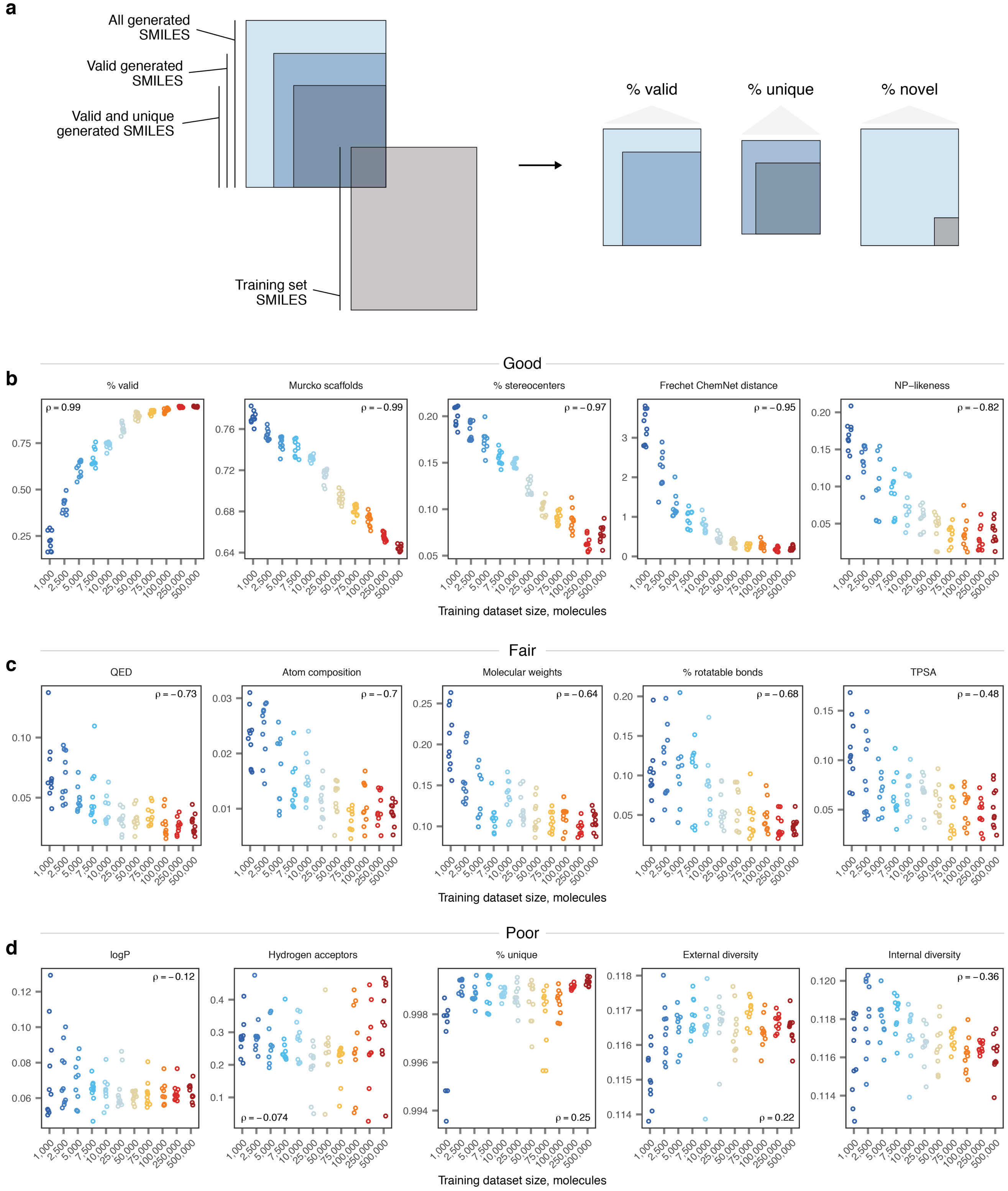
Acknowledgements. This work was supported by funding from Genome Canada, Genome British Columbia, and Genome Alberta (projects 284MBO and 214PRO), and computational resources provided by WestGrid, Compute Canada, and Advanced Research Computing at the University of British Columbia. M.A.S. acknowledges support from a CIHR Vanier Canada Graduate Scholarship, a Roman M. Babicki Fellowship in Medical Research, a Borealis AI Graduate Fellowship, a Walter C. Sumner Memorial Fellowship, and a Vancouver Coastal Health–CIHR–UBC MD/PhD Studentship. The authors thank J. Liigand and F. Wang for helpful discussions.

Competing interests. The authors declare no competing interests.

References

1. Bohacek, R. S., McMartin, C. & Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.* **16,** 3–50 (1996).
2. Pollock, S. N., Coutsias, E. A., Wester, M. J. & Oprea, T. I. Scaffold topologies. 1. Exhaustive enumeration up to eight rings. *J. Chem. Inf. Model.* **48,** 1304–1310 (2008).
3. Fink, T. & Reymond, J.-L. Virtual exploration of the chemical universe up to 11 atoms of C, N, O, F: assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery. *J. Chem. Inf. Model.* **47,** 342–353 (2007).
4. Blum, L. C. & Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.* **131,** 8732–8733 (2009).
5. Ruddigkeit, L., van Deursen, R., Blum, L. C. & Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.* **52,** 2864–2875 (2012).
6. Lameijer, E.-W., Kok, J. N., Bäck, T. & Ijzerman, A. P. The molecule evoluator. An interactive evolutionary algorithm for the design of drug-like molecules. *J. Chem. Inf. Model.* **46,** 545–552 (2006).
7. Van Deursen, R. & Reymond, J.-L. Chemical space travel. *ChemMedChem* **2,** 636–640 (2007).
8. Virshup, A. M., Contreras-García, J., Wipf, P., Yang, W. & Beratan, D. N. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *J. Am. Chem. Soc.* **135,** 7296–7303 (2013).
9. Elton, D. C., Boukouvalas, Z., Fuge, M. D. & Chung, P. W. Deep learning for molecular design—a review of the state of the art. *Mol. Syst. Des. Eng.* **4,** 828–849 (2019).
10. Cadeddu, A., Wylie, E. K., Jurczak, J., Wampler-Doty, M. & Grzybowski, B. A. Organic chemistry as a language and the implications of chemical linguistics for structural and retrosynthetic analyses. *Angew. Chem. Int. Ed.* **53,** 8108–8112 (2014).
11. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **28,** 31–36 (1988).
12. Gómez-Bombarelli, R. *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4,** 268–276 (2018).
13. Segler, M. H. S., Kogej, T., Tyrchan, C. & Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **4,** 120–131 (2018).
14. Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminform.* **9,** 48 (2017).
15. Merk, D., Friedrich, L., Grisoni, F. & Schneider, G. De novo design of bioactive small molecules by artificial intelligence. *Mol. Inform.* **37,** 1700153 (2018).
16. Popova, M., Isayev, O. & Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **4,** eaap7885 (2018).
17. Arús-Pous, J. *et al.* Exploring the GDB-13 chemical space using deep generative models. *J. Cheminform.* **11,** 20 (2019).
18. Moret, M., Friedrich, L., Grisoni, F., Merk, D. & Schneider, G. Generative molecular design in low data regimes. *Nat. Mach. Intell.* **2,** 171–180 (2020).
19. Kotsias, P.-C. *et al.* Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nat. Mach. Intell.* **2,** 254–265 (2020).
20. Li, Y., Zhang, L. & Liu, Z. Multi-objective de novo drug design with conditional graph generative model. *J. Cheminform.* **10,** 33 (2018).
21. Zhou, Z., Kearnes, S., Li, L., Zare, R. N. & Riley, P. Optimization of molecules via deep reinforcement learning. *Sci. Rep.* **9,** 10752 (2019).
22. Jin, W., Barzilay, R. & Jaakkola, T. S. Junction tree variational autoencoder for molecular graph generation. Preprint at http://arxiv.org/abs/1802.04364 (2018).
23. Brown, N., Fiscato, M., Segler, M. H. S. & Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **59,** 1096–1108 (2019).
24. Polykovskiy, D. *et al.* Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Front. Pharmacol.* **11,** 565644 (2020).
25. Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **361,** 360–365 (2018).
26. Rudolf, J. D., Alsup, T. A., Xu, B. & Li, Z. Bacterial terpenome. *Nat. Prod. Rep.* doi:10.1039/d0np00066c (2020).
27. Neil, D. *et al.* Exploring deep recurrent models with reinforcement learning for molecule design (2018).
28. Ståhl, N., Falkman, G., Karlsson, A., Mathiason, G. & Boström, J. Deep Reinforcement Learning for Multiparameter Optimization in de novo Drug Design. *J. Chem. Inf. Model.* **59,** 3166–3176 (2019).
29. Liu, X., Ye, K., van Vlijmen, H. W. T., IJzerman, A. P. & van Westen, G. J. P. An exploration strategy improves the diversity of de novo ligands using deep reinforcement learning: a case for the adenosine A2A receptor. *J. Cheminform.* **11,** 35 (2019).
30. Awale, M., Sirockin, F., Stiefl, N. & Reymond, J.-L. Drug analogs from fragment-based long short-term memory generative neural networks. *J. Chem. Inf. Model.* **59,** 1347–1356 (2019).
31. Merk, D., Grisoni, F., Friedrich, L. & Schneider, G. Tuning artificial intelligence on the de novo design of natural-product-inspired retinoid X receptor modulators. *Commun. Chem.* **1,** 68 (2018).
32. Amabilino, S., Pogány, P., Pickett, S. D. & Green, D. V. S. Guidelines for recurrent neural network transfer learning-based molecular generation of focused libraries. *J. Chem. Inf. Model.* **60,** 5699–5713 (2020).
33. Renz, P., Van Rompaey, D., Wegner, J. K., Hochreiter, S. & Klambauer, G. On failure modes in molecule generation and optimization. *Drug Discov. Today Technol.* **32-33,** 55–63 (2019).
34. Arús-Pous, J. *et al.* Randomized SMILES strings improve the quality of molecular generative models. *J. Cheminform.* **11,** 71 (2019).
35. Bjerrum, E. J. SMILES enumeration as data augmentation for neural network modeling of molecules. Preprint at http://arxiv.org/abs/1703.07076 (2017).
36. Irwin, J. J. & Shoichet, B. K. ZINC–a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model.* **45,** 177–182 (2005).
37. Benhenda, M. Can AI reproduce observed chemical diversity? Preprint at http://doi.org/10.1101/292177 (2018).

38. Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S. & Klambauer, G. Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *J. Chem. Inf. Model.* **58,** 1736–1741 (2018).
39. Van Deursen, R., Ertl, P., Tetko, I. V. & Godin, G. GEN: highly efficient SMILES explorer using autodidactic generative examination networks. *J. Cheminform.* **12,** 22 (2020).
40. Bertz, S. H. The first general index of molecular complexity. *J. Am. Chem. Soc.* **103,** 3599–3601 (1981).
41. Ertl, P., Roggo, S. & Schuffenhauer, A. Natural product-likeness score and its application for prioritization of compound libraries. *J. Chem. Inf. Model.* **48,** 68–74 (2008).
42. Mendez, D. *et al.* ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **47,** D930–D940 (2019).
43. Sorokina, M. & Steinbeck, C. Review on natural products databases: where to find data in 2020. *J. Cheminform.* **12,** 20 (2020).
44. Wildman, S. A. & Crippen, G. M. Prediction of physicochemical parameters by atomic contributions. *J. Chem. Inf. Comput. Sci.* **39,** 868–873 (1999).
45. Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **4,** 90–98 (2012).
46. Sanchez-Lengeling, B., Outeiral, C., Guimaraes, G. L. & Aspuru-Guzik, A. Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). Preprint at http://doi.org/10.26434/chemrxiv.5309668.v3 (2017).
47. O'Boyle, N. & Dalke, A. DeepSMILES: An adaptation of SMILES for use in machine-learning of chemical structures. Preprint at http://doi.org/10.26434/chemrxiv.7097960.v1 (2018).
48. Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **1,** 045024 (2020).
49. Kusner, M. J., Paige, B. & Hernandez-Lobato, J. M. Grammar variational autoencoder. Preprint at http://arxiv.org/abs/1703.01925 (2017).
50. Dai, H., Tian, Y., Dai, B., Skiena, S. & Song, L. Syntax-directed variational autoencoder for structured data. Preprint at http://arxiv.org/abs/1802.08786 (2018).
51. Bjerrum, E. J. & Sattarov, B. Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders. *Biomolecules* **8,** 131 (2018).
52. Winter, R., Montanari, F., Noé, F. & Clevert, D.-A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.* **10,** 1692–1701 (2019).
53. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15,** 1929–1958 (2014).
54. Smith, S. L., Kindermans, P.-J. & Le, Q. V. Don't decay the learning rate, increase the batch size. Preprint at http://arxiv.org/abs/1711.00489 (2017).
55. Johnston, C. W. *et al.* An automated Genomes-to-Natural Products platform (GNP) for the discovery of modular natural products. *Nat. Commun.* **6,** 8421 (2015).
56. Zhang, Q. *et al.* Structural investigation of ribosomally synthesized natural products by hypothetical structure enumeration and evaluation using tandem MS. *Proc. Natl. Acad. Sci. U.S.A.* **111,** 12031–12036 (2014).
57. Zheng, S. *et al.* QBMG: quasi-biogenic molecule generator with deep recurrent neural network. *J. Cheminform.* **11,** 5 (2019).
58. Wishart, D. S. *et al.* HMDB 4.0: the human metabolome database for 2018. *Nucleic Acids Res.* **46,** D608–D617 (2018).
59. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for dimension reduction. Preprint at http://arxiv.org/abs/1802.03426 (2018).
60. Da Silva, R. R., Dorrestein, P. C. & Quinn, R. A. Illuminating the dark matter in metabolomics. *Proc. Natl. Acad. Sci. U.S.A.* **112,** 12549–12550 (2015).
61. Coley, C. W., Eyke, N. S. & Jensen, K. F. Autonomous discovery in the chemical sciences part II: outlook. *Angew. Chem. Int. Ed.* **59,** 23414–23436 (2020).
62. Vanhaelen, Q., Lin, Y.-C. & Zhavoronkov, A. The advent of generative chemistry. *ACS Med. Chem. Lett.* **11,** 1496–1505 (2020).
63. Jaeger, S., Fulle, S. & Turk, S. Mol2vec: Unsupervised machine learning approach with chemical intuition. *J. Chem. Inf. Model.* **58,** 27–35 (2018).
64. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50,** 742–754 (2010).
65. Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. Molecular frameworks. *J. Med. Chem.* **39,** 2887–2893 (1996).
66. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **1,** 8 (2009).
67. Ertl, P., Rohde, B. & Selzer, P. Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *J. Med. Chem.* **43,** 3714–3717 (2000).
68. Sajed, T. *et al.* ECMDB 2.0: A richer resource for understanding the biochemistry of E. coli. *Nucleic Acids Res.* **44,** D495–501 (2016).
69. Huang, W. *et al.* PAMDB: a comprehensive Pseudomonas aeruginosa metabolome database. *Nucleic Acids Res.* **46,** D575–D580 (2018).
70. Moumbock, A. F. A. *et al.* StreptomeDB 3.0: an updated compendium of streptomycetes natural products. *Nucleic Acids Res.* doi:10.1093/nar/gkaa868 (2020).
71. Zeng, X. *et al.* NPASS: natural product activity and species source database for natural product research, discovery and tool development. *Nucleic Acids Res.* **46,** D1217–D1222 (2018).
72. Karp, P. D. *et al.* The BioCyc collection of microbial genomes and metabolic pathways. *Brief. Bioinform.* **20,** 1085–1093 (2019).
73. Neveu, V. *et al.* Phenol-Explorer: an online comprehensive database on polyphenol contents in foods. *Database* **2010,** bap024 (2010).
74. Ramirez-Gaona, M. *et al.* YMDB 2.0: a significantly expanded version of the yeast metabolome database. *Nucleic Acids Res.* **45,** D440–D445 (2017).



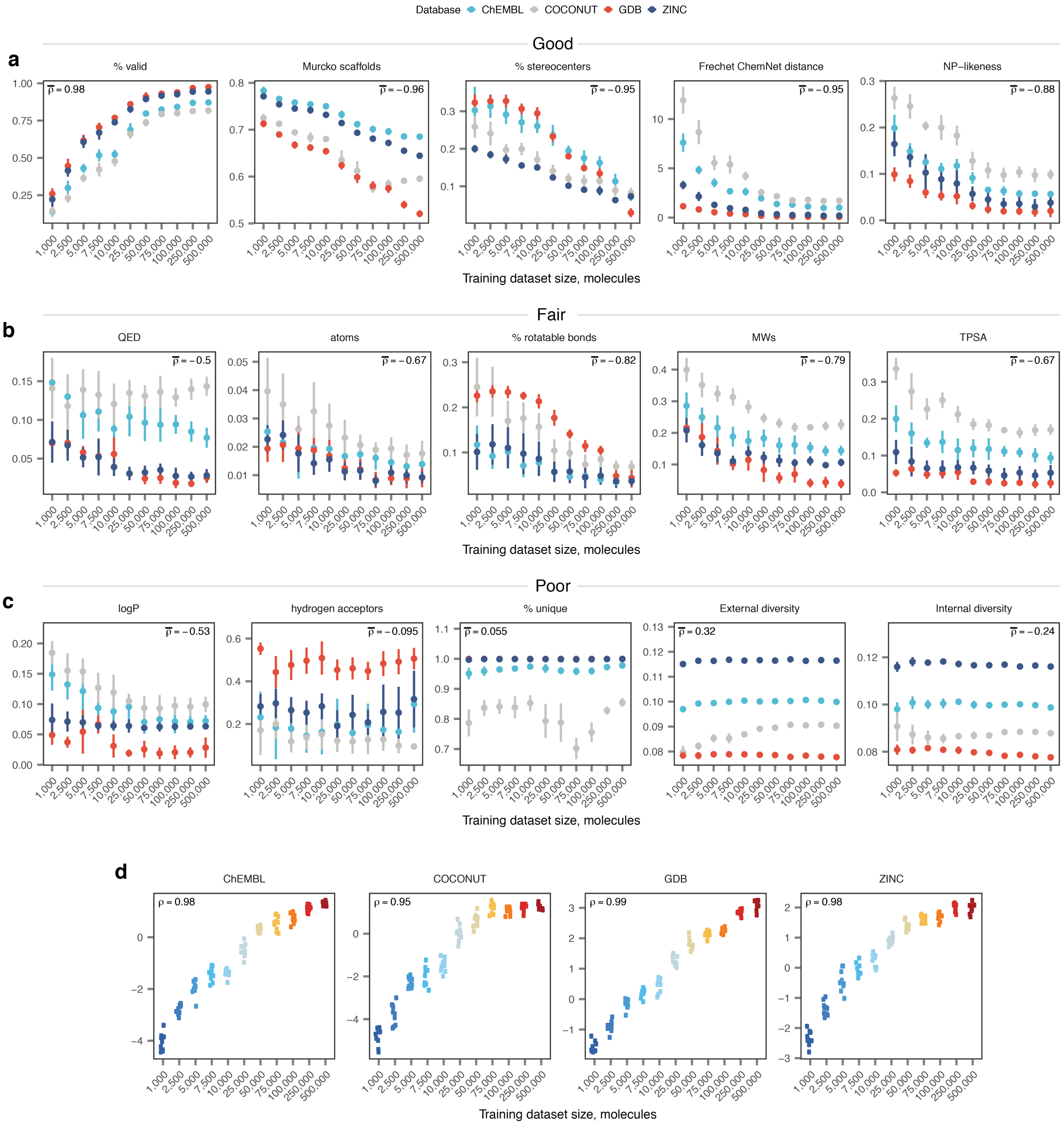
Supplementary Fig. 1 | Evaluating low-data generative models of purchasable chemical space.

a, Schematic overview of the "% valid", "% unique", and "% novel" metrics.

b, Values of the five top-performing metrics with the strongest correlations ($\rho \geq 0.82$) to training dataset size for n = 110 generative models trained on varying numbers of molecules from the ZINC database.

c, Values of five exemplary metrics with moderate to weak correlations ($0.48 \leq \rho \leq 0.73$) to training dataset size for n = 110 generative models trained on varying numbers of molecules from the ZINC database.

d, Values of five exemplary metrics with little or no correlation ($\rho \leq 0.36$) to training dataset size for n = 110 generative models trained on varying numbers of molecules from the ZINC database.



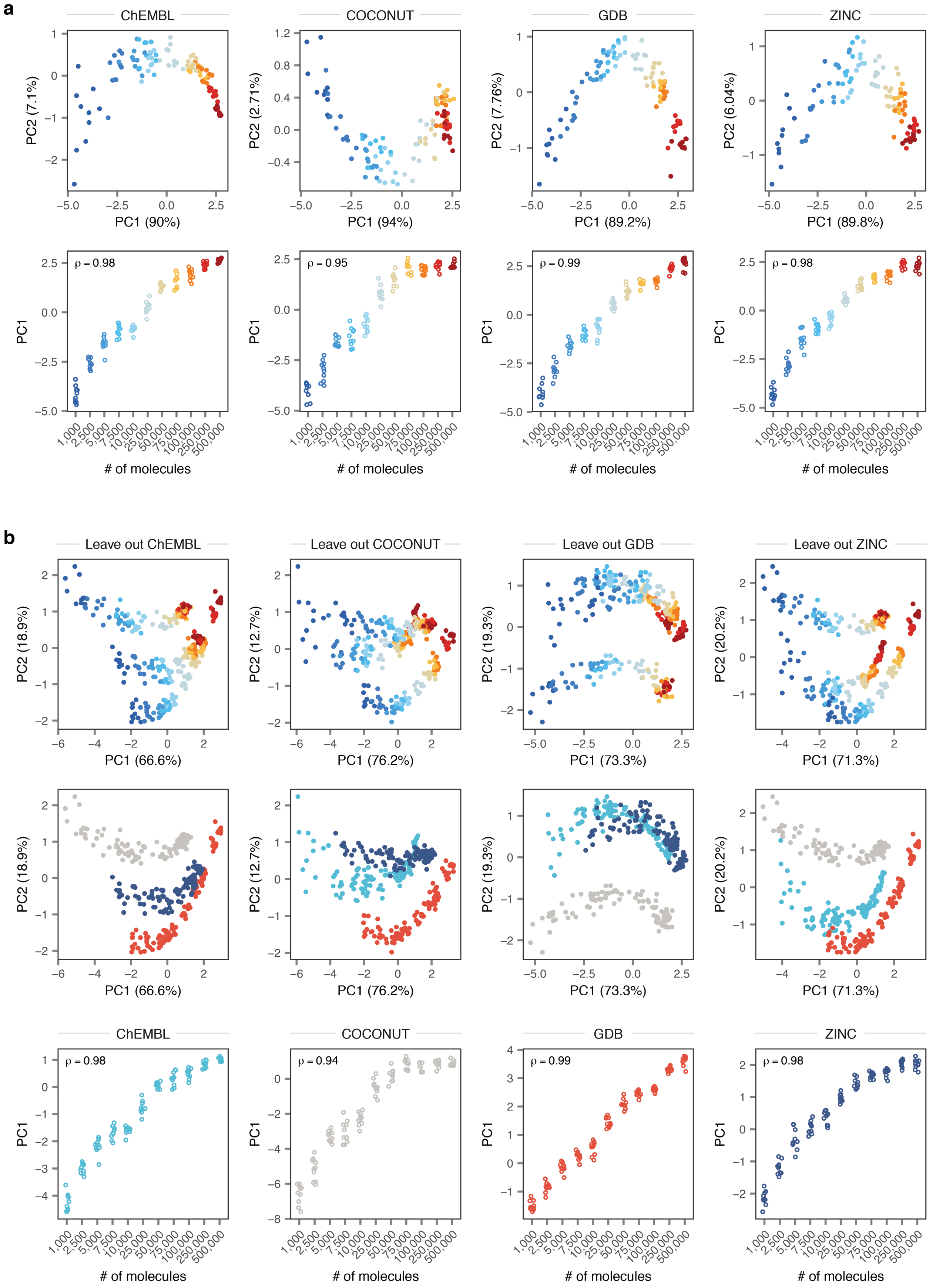
Supplementary Fig. 2 | Evaluating low-data generative models of divergent chemical spaces.

a, Values of the five top-performing metrics with the strongest correlations (average rank correlation ≥ 0.80) to training dataset size for $n = 440$ generative models trained on varying numbers of molecules from the ChEMBL, COCONUT, GDB, or ZINC databases. Points and error bars show the mean and standard deviation, respectively, of ten independent replicates.

b, Values of five exemplary metrics with moderate to weak correlations to training dataset size for $n = 440$ generative models trained on varying numbers of molecules from the ChEMBL, COCONUT, GDB, or ZINC databases.

c, Values of five exemplary metrics with little or no correlation to training dataset size for $n = 440$ generative models trained on varying numbers of molecules from the ChEMBL, COCONUT, GDB, or ZINC databases.

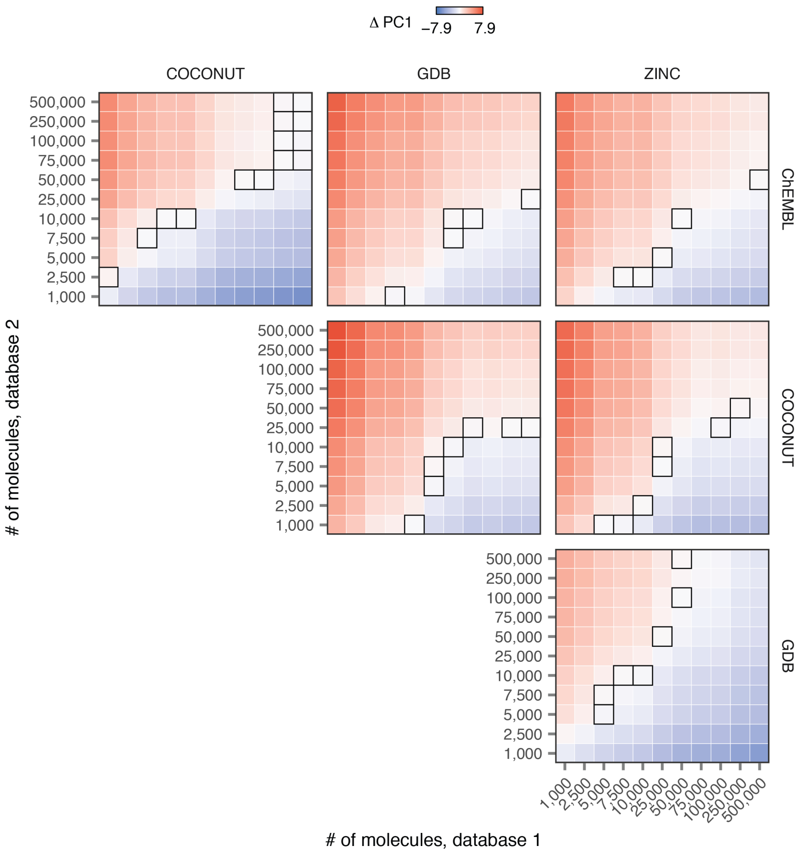
d, PC1 scores for $n = 440$ chemical language models trained on varying numbers of molecules sampled from the ChEMBL, COCONUT, GDB, or ZINC databases. Inset text shows the Spearman correlation.


a
ChEMBL
COCONUT
GDB
ZINC
PC2 (7.1%)
PC1 (90%)
PC2 (2.71%)
PC1 (94%)
PC2 (7.76%)
PC1 (89.2%)
PC2 (6.04%)
PC1 (89.8%)
ρ = 0.98
ρ = 0.95
ρ = 0.99
ρ = 0.98
PC1
1,000 2,500 5,000 7,500 10,000 25,000 50,000 75,000 100,000 250,000 500,000
# of molecules
b
Leave out ChEMBL
Leave out COCONUT
Leave out GDB
Leave out ZINC
PC2 (18.9%)
PC1 (66.6%)
PC2 (12.7%)
PC1 (76.2%)
PC2 (19.3%)
PC1 (73.3%)
PC2 (20.2%)
PC1 (71.3%)
ChEMBL
COCONUT
GDB
ZINC
ρ = 0.98
ρ = 0.94
ρ = 0.99
ρ = 0.98
PC1
1,000 2,500 5,000 7,500 10,000 25,000 50,000 75,000 100,000 250,000 500,000
# of molecules

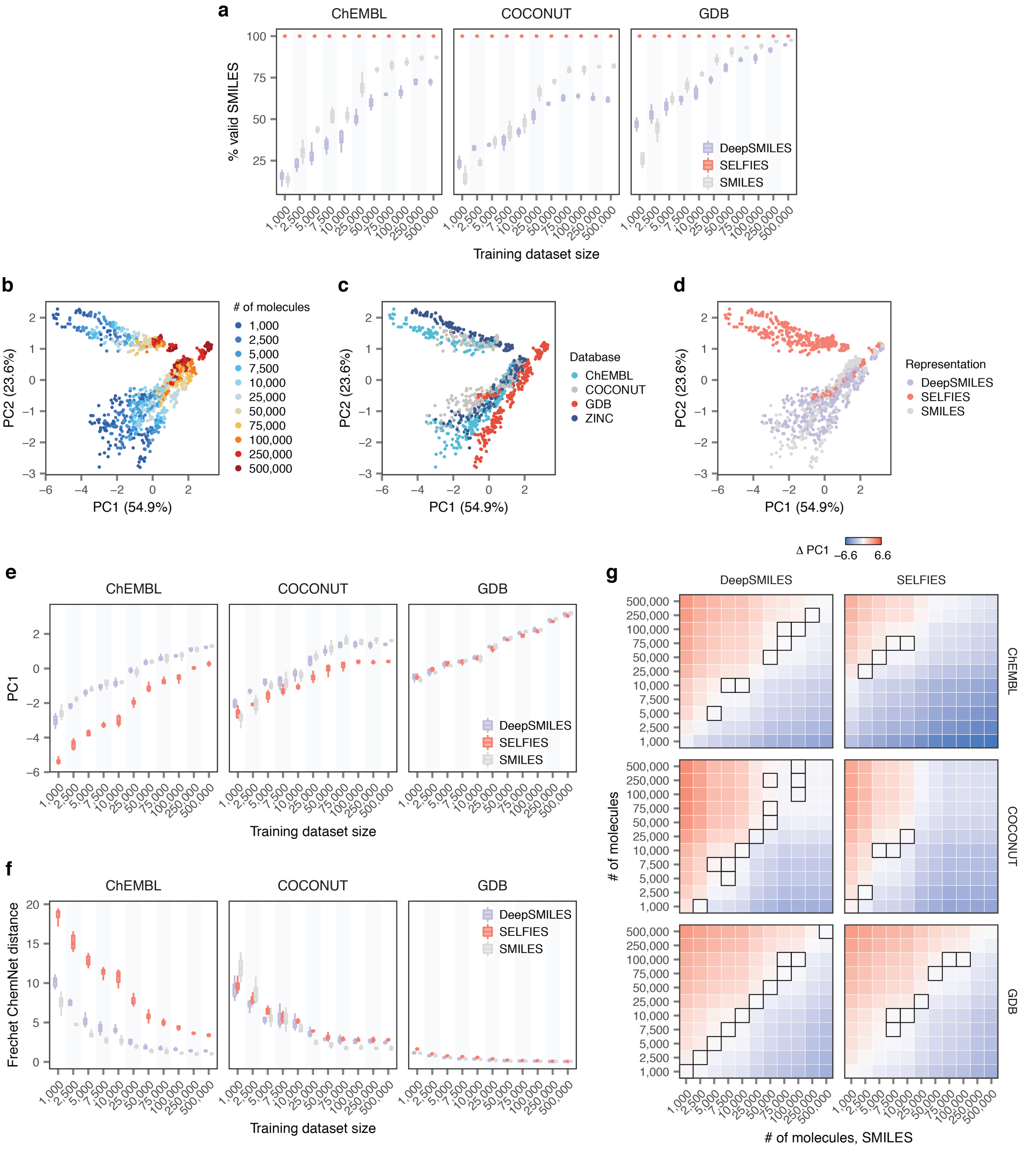
Supplementary Fig. 3 | Robustness of principal component analysis for the evaluation of chemical generative models.

a, PCA of top-performing metrics, top, and PC1 scores, bottom, for chemical language models trained on varying numbers of molecules sampled from the ChEMBL, COCONUT, GDB, and ZINC database, with PCA performed separately for each database. Bottom, inset text shows the Spearman correlation.

b, PCA of top-performing metrics for chemical language models trained on varying numbers of molecules sampled from three of four databases, colored by the size of the training dataset, top, or the chemical database on which the generative models were trained, middle. Bottom, PC1 scores for models trained on the withheld database, projected onto the coordinate basis of the other three databases. Inset text shows the Spearman correlation.



Supplementary Fig. 4 | Training dataset size requirements in different chemical spaces.
Mean difference in PC1 scores between chemical language models trained on varying numbers of molecules sampled from each pair of chemical structure databases. Dark squares indicate pairs without statistically significant differences (uncorrected p > 0.05, two-sided t-test).


a
ChEMBL
COCONUT
GDB
% valid SMILES
100
75
50
25
DeepSMILES
SELFIES
SMILES
1,000
2,500
5,000
7,500
10,000
25,000
50,000
75,000
100,000
250,000
500,000
Training dataset size
b
PC2 (23.6%)
PC1 (54.9%)
# of molecules
1,000
2,500
5,000
7,500
10,000
25,000
50,000
75,000
100,000
250,000
500,000
c
Database
ChEMBL
COCONUT
GDB
ZINC
d
Representation
DeepSMILES
SELFIES
SMILES
e
PC1
Training dataset size
f
Frechet ChemNet distance
20
15
10
5
0
Training dataset size
g
Δ PC1
−6.6
6.6
DeepSMILES
SELFIES
ChEMBL
COCONUT
GDB
# of molecules
# of molecules, SMILES

Supplementary Fig. 5 | Evaluating alternative molecular representations for low-data generative models in distinct chemical spaces.

a, Proportion of valid SMILES generated by chemical language models trained on one of three string representations of molecules from the ChEMBL, COCONUT, and GDB databases.

b, PCA of top-performing metrics for molecules generated by *n* = 1,320 chemical language models trained on one of three string representations of molecules from the ChEMBL, COCONUT, and GDB databases, colored by the size of the training dataset.

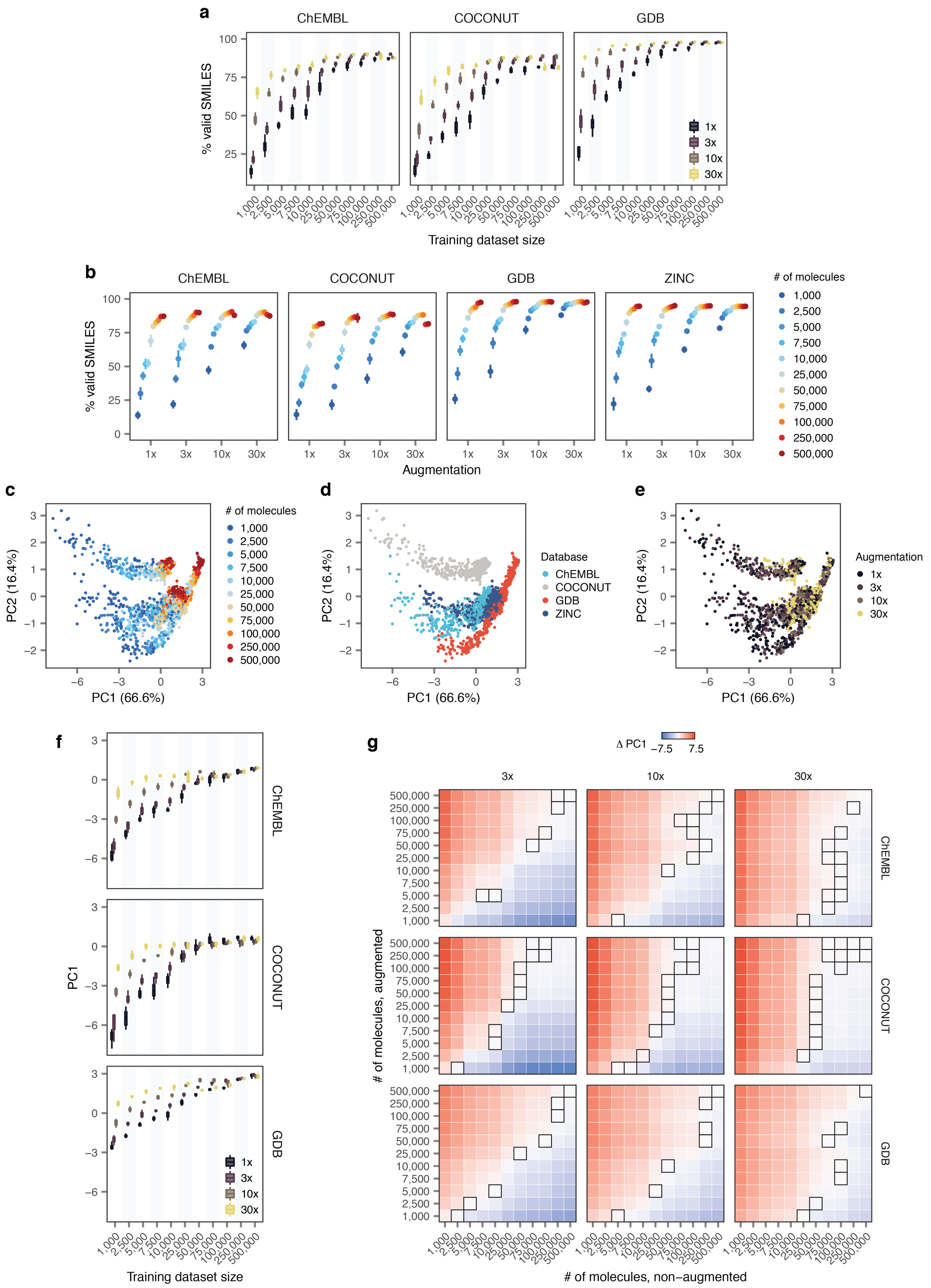
c, As in **b**, but colored by the chemical database on which the generative models were trained.

d, As in **b**, but colored by molecular representation.

e, PC1 scores of chemical language models trained on one of three string representations of molecules from the ChEMBL, COCONUT, and GDB databases.

f, Fréchet ChemNet distances of chemical language models trained on one of three string representations of molecules from the ChEMBL, COCONUT, and GDB databases.

g, Mean difference in PC1 scores between chemical language models trained on varying numbers of molecules sampled from the ChEMBL, COCONUT, and GDB databases, represented either as DeepSMILES or SELFIES, y-axis, or SMILES, x-axis. Dark squares indicate pairs without statistically significant differences (uncorrected p > 0.05, two-sided t-test).


a
ChEMBL
COCONUT
GDB
% valid SMILES
100
75
50
25
1x
3x
10x
30x
1,000
2,500
5,000
7,500
10,000
25,000
50,000
75,000
100,000
250,000
500,000
Training dataset size
b
ChEMBL
COCONUT
GDB
ZINC
% valid SMILES
100
75
50
25
0
1x
3x
10x
30x
Augmentation
# of molecules
1,000
2,500
5,000
7,500
10,000
25,000
50,000
75,000
100,000
250,000
500,000
c
d
e
PC2 (16.4%)
PC1 (66.6%)
3
2
1
0
−1
−2
−6
−3
0
3
# of molecules
Database
ChEMBL
COCONUT
GDB
ZINC
Augmentation
1x
3x
10x
30x
f
PC1
3
0
−3
−6
ChEMBL
COCONUT
GDB
1x
3x
10x
30x
Training dataset size
g
Δ PC1
−7.5
7.5
3x
10x
30x
# of molecules, augmented
ChEMBL
COCONUT
GDB
# of molecules, non−augmented

Supplementary Fig. 6 | Data augmentation by non-canonical SMILES enumeration.

a, Proportion of valid SMILES generated by chemical language models trained on molecules from the ChEMBL, COCONUT, and GDB databases after varying degrees of non-canonical SMILES enumeration.

b, Data as in **a** and **Fig. 3i**, but showing the relationship between the size of the training dataset and the proportion of valid SMILES generated by models for each degree of non-canonical SMILES enumeration separately.

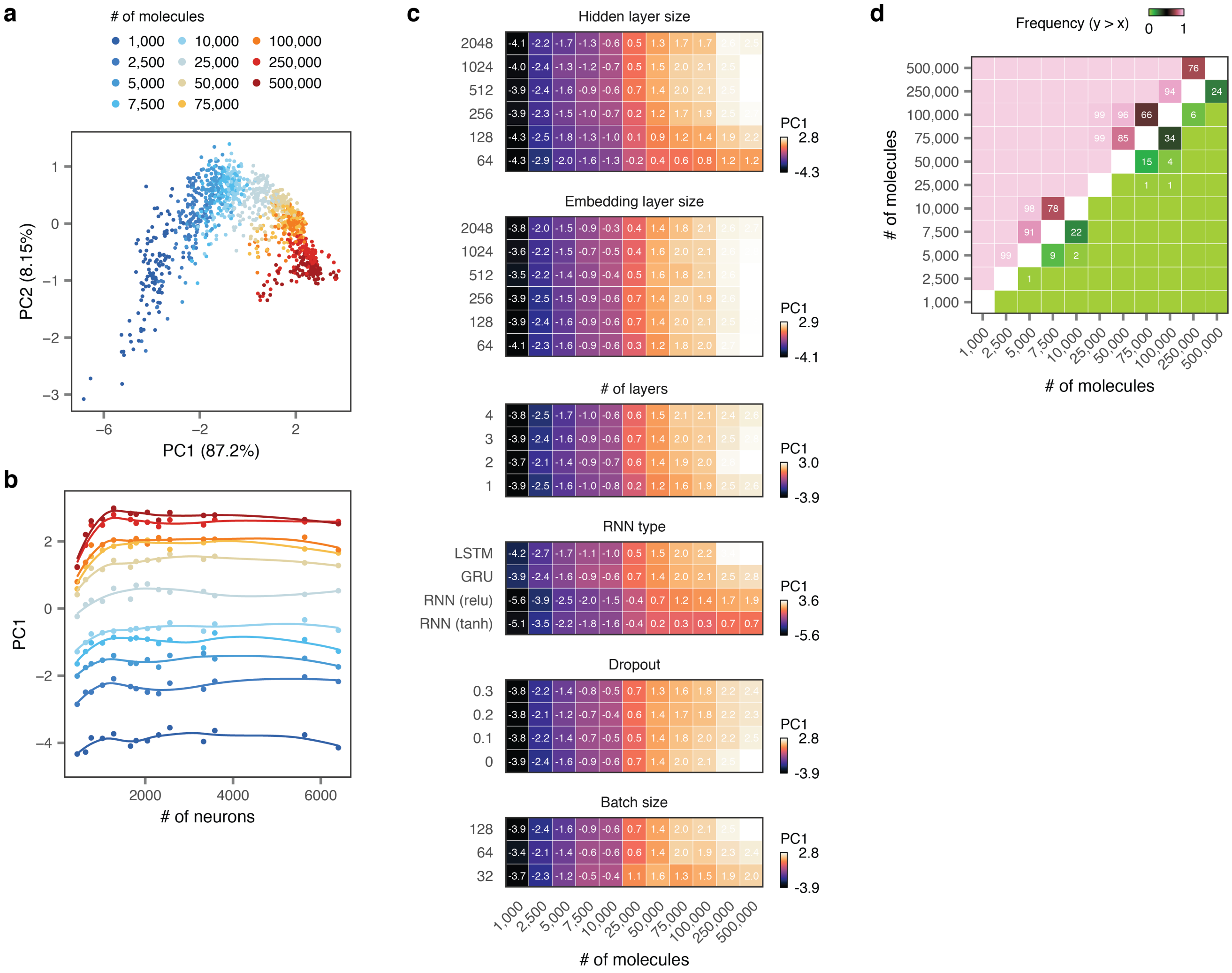
c, PCA of top-performing metrics for molecules generated by n = 1,760 chemical language models trained on molecules from the ChEMBL, COCONUT, and GDB databases after varying degrees of non-canonical SMILES enumeration, colored by the size of the training dataset.

d, As in **c**, but colored by the chemical database on which the generative models were trained.

e, As in **c**, but colored by the amount of SMILES enumeration.

f, PC1 scores of chemical language models trained on molecules from the ChEMBL, COCONUT, and GDB databases after varying degrees of non-canonical SMILES enumeration.

g, Mean difference in PC1 scores between chemical language models trained on molecules from the ChEMBL, COCONUT, and GDB databases represented as canonical SMILES, x-axis, or non-canonical SMILES after varying degrees of data augmentation, y-axis. Dark squares indicate pairs without statistically significant differences (uncorrected $p > 0.05$, two-sided t-test).



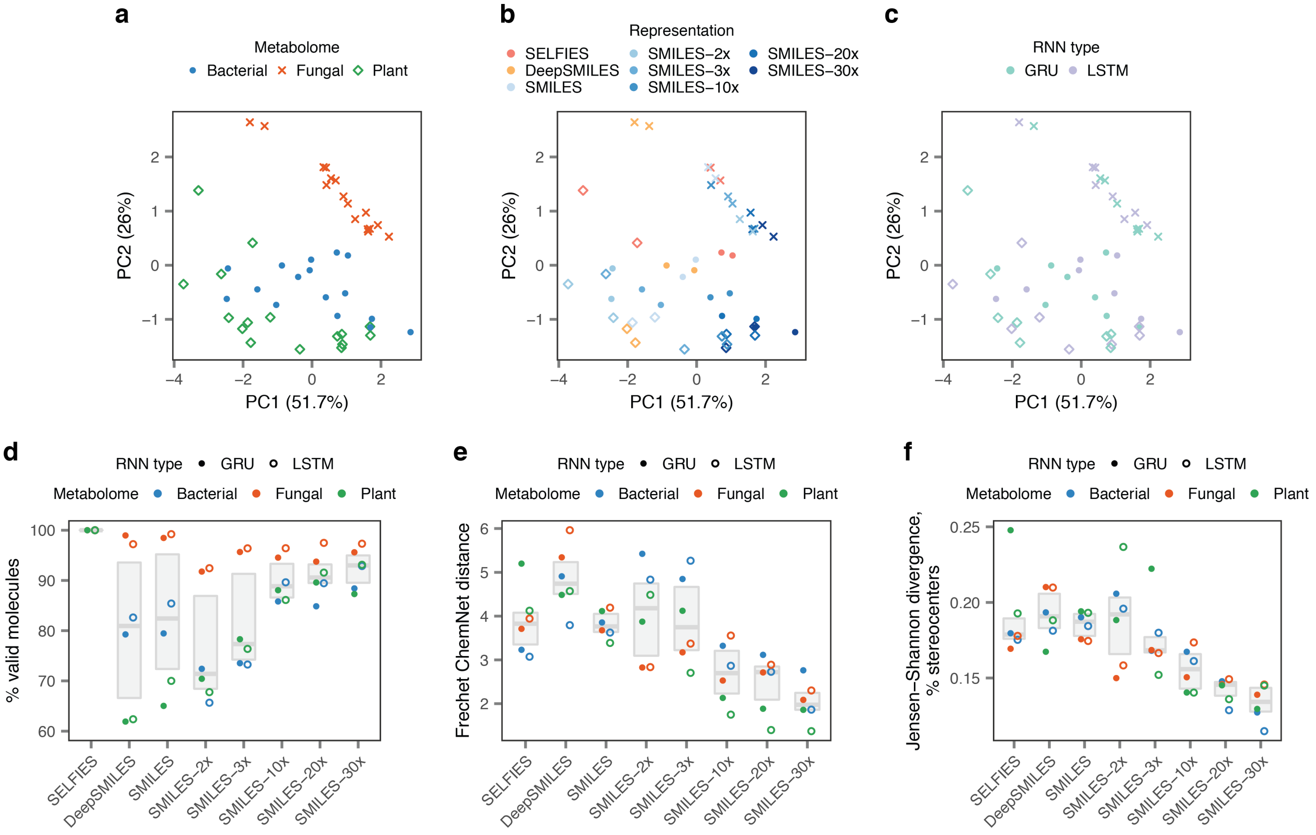
Supplementary Fig. 7 | Hyperparameter tuning in the ChEMBL database.

a, PCA of top-performing metrics for molecules generated by *n* = 1,210 chemical language models, trained on varying numbers of molecules from the ChEMBL database with varying model hyperparameters, colored by the size of the training dataset.

b, Mean PC1 scores of chemical language models as a function of the total number of neurons in the model. Solid lines show local polynomial regression.

c, Mean PC1 scores for molecules trained on the ChEMBL database, as a function of both the number of molecules in the training dataset, x-axis, and varying hyperparameters, y-axis. The mean of five independent replicates is shown.

d, Proportion of *n* = 110 chemical language models with varying hyperparameters, trained on the number of molecules shown on the y-axis, that outperformed a model without hyperparameter tuning trained on the number of molecules shown on the x-axis.



Supplementary Fig. 8 | Optimizing generative models of bacterial, fungal, and plant metabolomes.

a, PCA of top-performing metrics for molecules generated by n = 48 chemical language models, trained on bacterial, fungal, or plant metabolomes with varying inputs and hyperparameters, colored by the target metabolome.

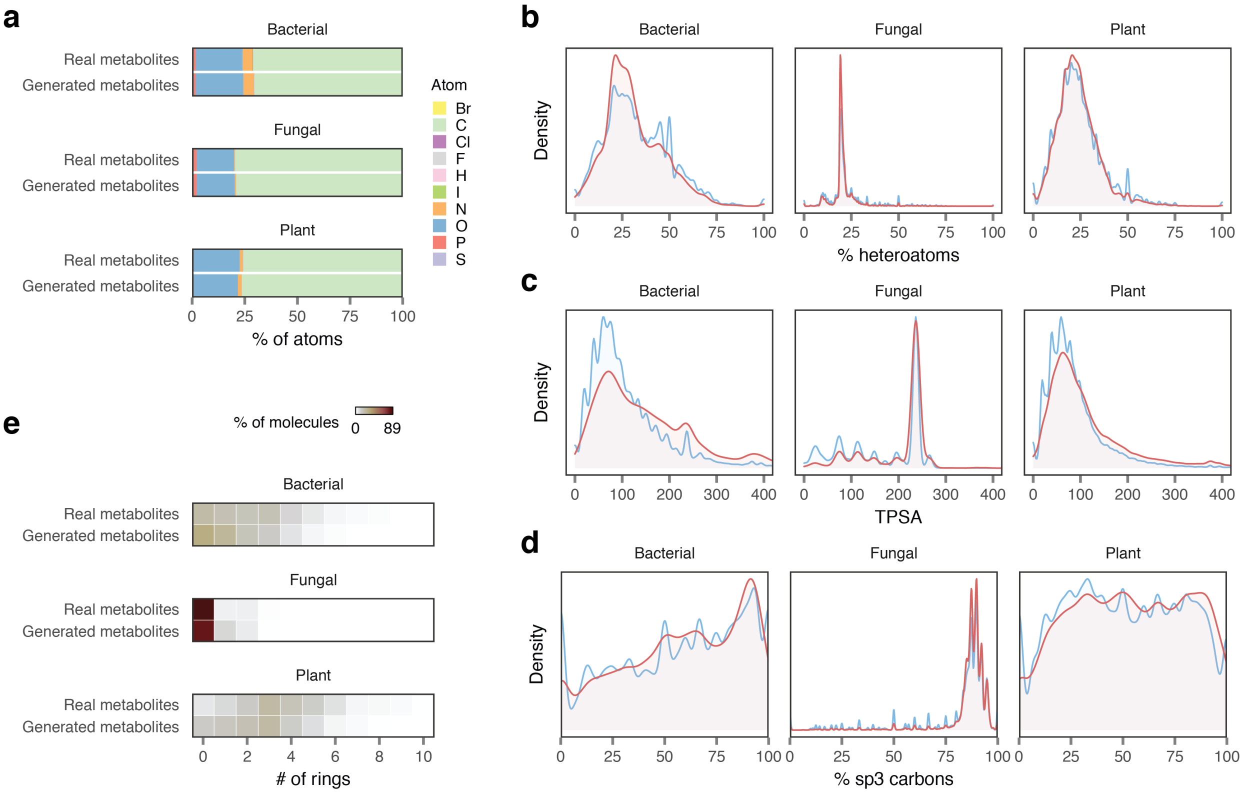
b, As in **a**, but colored by the molecular representation and data augmentation strategy.

c, As in **a**, but colored by the RNN architecture.

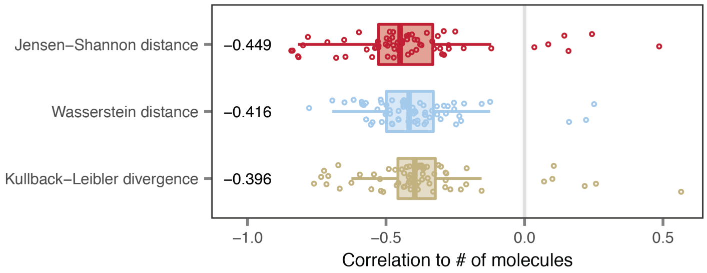
d, Proportion of valid molecules produced by generative models of metabolomes trained with different molecular representations (SMILES, DeepSMILES, or SELFIES), data augmentation strategies (non-canonical SMILES enumeration with an augmentation factor of between 2x and 30x), and RNN architectures (GRU or LSTM).

e, As in **d**, but showing the Fréchet ChemNet distance between generated and real metabolites.

f, As in **d**, but showing the Jensen-Shannon divergence of the proportion of stereocenters between generated and real metabolites.



Supplementary Fig. 9 | Physicochemical properties of generated metabolites.
a, Atom composition of real and generated bacterial, fungal, and plant metabolites.
b, Proportion of heteroatoms in real and generated metabolites.
c, Topological polar surface area of real and generated metabolites.
d, Proportion of sp3 carbons in real and generated metabolites.
e, Number of rings found in real and generated metabolites.



Supplementary Fig. 10 | Comparing measures of differences between property distributions.
Correlation between the Jensen-Shannon distance, Wasserstein distance, or Kullback-Leibler divergences of 17 structural or physicochemical properties between molecules in the training set and molecules generated by chemical language models, and the size of the training dataset, for a total of 440 chemical language models trained on the ChEMBL, COCONUT, GDB, or ZINC databases.