Supplementary Information

"Back to the Future" of Lead Optimization: Benchmarking Compound Prioritization Strategies

Pablo Mas,[ab] Bruno Filoche-Rommé,[b] Marc Bianciotto,[b] and Rodolphe Vuilleumier[a]

[a] *Chimie Physique et Chimie du Vivant, École Normale Supérieure, PSL Université, Sorbonne Université, CNRS, 75005 Paris, France*
[b] *Integrated Drug Discovery, R&D, Sanofi, 94400 Vitry-sur-Seine, France*

1 Datasets information and simulation setup

1.1 FXa

The FXa dataset comprises 1015 molecules characterized by three endpoints: pKi, LogD (pH 7.40), and solubility (pH 7.40, μM). The compounds span a timeline from 2001-07-31 to 2007-09-19. Simulations begin on 2002-10-01, at which point 154 molecules—each documented on all three endpoints—are already present in the project and constitute the initial training set. The simulation uses a monthly timestep over 60 iterations. A batch size of 0.5 is applied, meaning that in each iteration, the acquisition function selects half the number of molecules that appeared in the original project during the corresponding month. Under these parameters, 58% of the dataset is used in each virtual scenario: 15% (154 molecules) in the initial set and 43% selected progressively across iterations.

1.2 Renin

The Renin dataset includes 388 molecules, each annotated with pIC_{50} values. The project spans from 2006-02-14 to 2008-05-06. Simulations start on 2006-12-01 with 128 molecules already present, forming the initial training set. The simulation proceeds with a monthly timestep over 18 iterations. A batch size of 0.5 is used, resulting in 66% of the dataset being selected during the simulation: 33% initially and another 33% acquired throughout the iterations.

1.3 PPARδ

The PPARδ dataset consists of 498 molecules with pEC_{50} values. The project timeline ranges from 2002-11-08 to 2008-10-29. Simulations begin on 2005-09-01, by which time 131 molecules are already available. With a monthly timestep, the simulation runs for 38 iterations. A batch size of 0.5 leads to the selection of 64% of the dataset: 26% initially and 38% over the course of the simulation.

1.4 MMP-8

The MMP-8 dataset comprises 430 molecules, each with a pIC_{50} value. The dataset spans from 2001-04-17 to 2005-04-20. Simulations commence on 2002-03-01, with 148 molecules already available. Given the discrete timeline of this project—only 12 distinct dates of compound additions—a 5-month timestep is used, resulting in 8 iterations. With a batch size of 0.5, the simulation selects 67% of the dataset: 34% in the initial training set and 33% acquired during iterations.

Table 1 Overview of datasets used for retrospective analysis, including size, number of endpoints, and temporal coverage.

Dataset	Size	Endpoints	First date	Final date
FXa	1015	3	2001-07-31	2007-09-19
Renin	388	1	2006-02-14	2008-05-06
PPARδ	498	1	2002-11-08	2008-10-29
MMP-8	430	1	2001-04-17	2005-04-20

Table 2 Parameters used in retrospective simulations, including dataset-specific starting dates, batch sizes, timestep intervals, total number of iterations, and proportion of the full dataset selected.

Dataset	Simulation starting date	Batch size	Timestep	Model	Iterations	% all molecules selected (initial + during iterations)
FXa	2002-10-01	0.5	1 month	Random Forest (std)	60	57% (12% + 45%)
Renin	2006-12-01	0.5	1 month	Random Forest (std)	18	66% (33% + 33%)
PPARδ	2005-07-01	0.5	1 month	Random Forest (std)	40	63% (26% + 37%)
MMP-8	2002-03-01	0.5	5 months	Random Forest (std)	8	67% (34% + 33%)

2 Acquisition Functions

Table 3 Summary of acquisition functions categorized by objective, selection type and reliance on predictive modeling.

Acquisition function	Objective	Selection	Model	Reference	Description
Coverage score	Exploration	Batch	No	Woodward *et al.* [1]	Select batch of compounds that has the highest coverage score, based on bayesian statistics and information entropy.
Desirability	Exploitation	Independent	Yes	Cummins and Bell [2]	Select compounds with the highest predicted desirabilities.
Desirability with AD	Exploitation	Independent	Yes	Sheridan *et al.* [3]	Select compounds with the highest predicted desirability within a distance-based applicability domain.
Desired Coverage	Trade-off	Batch	Yes	This work	Select compounds with the highest composite score combining desirability score and coverage score.
Desired Spread	Trade-off	Batch	Yes	This work	Select compounds with the highest composite score combining desirability score and spread score.
Dissimilarity-to-known	Exploration	Independent	No	-	Select most dissimilar compounds to the ones previously selected.
Dissimilarity-to-known-good	Exploration	Independent	No	-	Select most dissimilar compounds to the good ones previously selected.
GRA	Exploration	Independent	Yes	Adapted from Ju-Long [4]	Select compounds that are the closest predicted values to an ideal solution based on relational coefficients between alternatives. Modified from its original methodology to use a utility function-based normalization.
Greediverse	Trade-off	Batch	Yes	Langevin *et al.* [5]	Select compounds that optimize the predicted desirability scores within a batch, with a penalty if the molecules from the batch are too similar.
K-means	Exploration	Batch	No	MacQueen *et al.* [6]	Select the compound that is the closest to each cluster's centroid.
K-medoids	Exploration	Batch	No	Kaufman and Rousseeuw [7]	Select the compound that is the medoid for each cluster.
Random	Exploration	Independent	No	-	Select random compounds. Proxy to the real project selection at comparable batch size.
Similarity-to-known	Exploration	Independent	No	-	Select most similar compounds to the ones previously selected.
Similarity-to-known-good	Exploration	Independent	No	-	Select most similar compounds to the good ones previously selected.
Spread	Exploration	Batch	No	Kennard and Stone [8]	Select diverse compounds by iteratively selecting the ones that maximize the minimum distance to their closest neighbor among previously selected compounds.
TOPSIS	Exploitation	Independent	Yes	Adapted from Hwang and Yoon [9]	Select compounds that are the closest predicted values to an ideal solution based on euclidean distance.
Uncertainty	Exploration	Independent	Yes	This work	Select compounds that have the highest predicted uncertainty score aggregating uncertainty across the different endpoints.
U/D Harmonic	Trade-off	Independent	Yes	This work	Select compounds that have the highest harmonic score, combining predicted uncertainty and desirability scores.
U/D Ratio	Trade-off	Independent	Yes	This work	Select compounds that have the highest ratio score, combining predicted uncertainty and desirability scores.

2.1 Model-Agnostic Acquisition Functions

2.1.1 Random

The random acquisition function selects molecules from the pool set randomly, disregarding uncertainty, diversity or any predicted endpoints values. While typically viewed as a simplistic or "dummy" strategy in active learning, random sampling offers practical advantages for retrospective analyses. Because our pool consists exclusively of molecules previously synthesized and tested in actual projects, and given that our simulated batch sizes are smaller than those employed originally, random sampling repeatedly selects distinct subsets from the historical pool. Consequently, when executed numerous times, this approach approximates the original selection process, adjusted to align with the current simulation's parameters. In our study, the Random strategy thus serves as an essential baseline, approximating the real project selection at comparable batch size.

2.1.2 K-Means Clustering

K-means clustering [6] promotes diversity by grouping compounds based on their molecular feature representations. In our method, the feature space combines Extended Connectivity Fingerprints (ECFP, 2048 bits, radius of 2, incorporating chirality and count) [10] and 159-bit MACCS keys (with count), computed via the scikit-fingerprints library [11]. From each cluster, the compound closest to the centroid is

selected.

2.1.3 K-Medoids Clustering

K-medoids clustering[7] similarly promotes diversity by clustering compounds within molecular feature space but selects each cluster's medoid rather than its centroid. Although computationally more demanding than k-means, this method is more robust against outliers by using actual data points rather than mean positions, thus reducing sensitivity to noise and extreme values. The default distance metric is Euclidean, but other pairwise distance metrics may also be applied.

2.1.4 Coverage Score

The coverage score[1] is a subset-based selection technique leveraging Bayesian statistics and information entropy to balance representativeness and diversity.

First, we define the compound sets involved in calculating the coverage score:

- S_{pool}: Set from which compounds are selected.
- $S_{selected}$: Set of compounds selected in the current iteration, $S_{selected} \subseteq S_{pool}$.
- S_{train}: Training set containing compounds selected in previous iterations.
- S_{full}: Full set defined as $S_{pool} \cup S_{train}$.
- $S_{sampled}$: Sampled set defined as $S_{selected} \cup S_{train}$.

Each compound is characterized by a molecular fingerprint (concatenation of ECFP4 and MACCS keys) comprising k features. The coverage score for each fingerprint feature is determined based on its frequency within the sampled and full sets. Bayesian statistics are applied to estimate the probability of a fingerprint feature f_k occurring in the sampled set. This probability, smoothed using Laplacian smoothing, is defined as:

$$P_{\text{smooth}}(\text{sampled} \mid f_k) = \frac{F_{f_k,\text{sampled}} + \alpha}{F_{f_k,\text{full}} + \frac{\alpha}{P(\text{sampled})}} \tag{1}$$

where α is an additive smoothing parameter set to one, and $F_{f_k,\text{full}}$ and $F_{f_k,\text{sampled}}$ denote the frequencies of fingerprint feature f_k in the full and sampled sets, respectively.

The term $P(\text{sampled})$ represents the probability of a compound belonging to the sampled set and is calculated as:

$$P(\text{sampled}) = \frac{|S_{\text{sampled}}|}{|S_{\text{full}}|} = \frac{N_{\text{sampled}}}{N_{\text{full}}} \tag{2}$$

where N_{full} and N_{sampled} are the cardinalities of the full and sampled sets, respectively.

The base coverage score for a fingerprint feature, $C_{base}(f_k)$, is defined as the negative log-likelihood ratio:

$$C_{base}(f_k) = -\ln\left(\frac{P_{smooth}(\text{sampled} \mid f_k)}{P_{sampled}}\right) \tag{3}$$

To assess the informativeness of each feature within the sampled set, a Shannon entropy term $H(f_k \mid \text{sampled})$ is incorporated:

$$H(f_k \mid \text{sampled}) = -\frac{p_1 \ln(p_1) + p_2 \ln p_2}{\ln(2)} \tag{4}$$

where $p_1 = P(f_k \mid \text{sampled}) = \frac{F_{f_k,\text{sampled}}}{N_{\text{sampled}}}$ and $p_2 = 1 - p_1$. The division by $\ln(2)$ normalizes entropy to a $[0,1]$ scale, with entropy explicitly set to 0 when $F_{f_k,\text{sampled}} = N_{\text{sampled}}$.

The final coverage score for each fingerprint feature, $C_{final}(f_k)$, is computed as:

$$C_{\text{final}}(x_i) = \begin{cases} C_{\text{base}}(f_k)(2 - H(f_k \mid \text{sampled})) & \text{if } C_{\text{base}}(f_k) < 0 \\ & \cap \frac{F_{f_k,\text{sampled}}}{N_{\text{sampled}}} > \frac{1}{2} \\ C_{\text{base}}(f_k) H(f_k \mid \text{sampled}) & \text{otherwise} \end{cases} \tag{5}$$

The compound-level coverage score, $C_{comp}(x)$, is calculated as the sum of coverage scores of the features present in the compound's fingerprint:

$$C_{comp}(x) = \sum_{f_k} C_{final}(f_k) \tag{6}$$

Finally, the overall coverage score of the selected set $S_{selected}$ is obtained by summing the coverage scores of its constituent compounds:

$$C_{score}(S_{selected}) = \sum_{x \in S} C_{comp}(x) \tag{7}$$

The set of molecules selected is the one achieving the highest coverage score. The original paper utilizes the NSGA-II genetic algorithm[12] to identify this set. However, internal evaluations demonstrated that a greedy optimization strategy consistently yields comparable or better performance while offering superior computational efficiency and reproducibility. Consequently, we employ a greedy optimization approach in our implementation.

2.1.5 Spread Sampling

Spread sampling[8] is a sequential, greedy, space-filling approach designed to maximize diversity in the selected subset. At each iteration, the method selects the molecule that maximizes the minimum distance from its nearest neighbors within both the training set and previously selected molecules. Distances are calculated using the Tanimoto similarity metric with binary ECFP fingerprints (2048 bits, radius of 2). By emphasizing diversity, spread sampling ensures a representative coverage of the molecular feature space.

2.1.6 Similar-to-Known

The similar-to-known acquisition function prioritizes pool-set compounds that are highly similar to molecules already included in the training set, determined via Tanimoto similarity. Specifically, it selects batches exhibiting the highest average similarity to existing training compounds. However, this method does not consider diversity within the selected batch.

2.1.7 Dissimilar-to-Known

Conversely, the dissimilar-to-known acquisition function emphasizes selecting compounds least similar to the training set, again employing Tanimoto similarity. It selects batches with the lowest average similarity to known compounds, also neglecting internal batch diversity.

2.1.8 Similar-to-Good-Known

The similar-to-good-known acquisition function specifically selects pool-set compounds closely resembling "good" compounds in the training set, where "good" is defined by an experimental desirability score greater than 0.5. Similarity is assessed using Tanimoto similarity. If the training set lacks "good" compounds, this strategy defaults to the similar-to-known approach previously described.

2.1.9 Dissimilar-to-Good-Known

Similarly, the dissimilar-to-good-known acquisition function targets pool-set compounds least similar to the "good" compounds from the training set, with the same desirability threshold (>0.5). Tanimoto similarity again measures similarity. If the training set does not include any "good" compounds, the strategy defaults to the dissimilar-to-known approach described earlier.

2.2 QSAR/QSPR models

Before presenting the model-dependent acquisition function, we briefly review the QSAR/QSPR modeling methods available within our toolkit. Lead optimization typically operates in a low-data context, posing challenges to reliable predictive modeling. Several studies[13–16] have indicated that deep neural networks (DNNs), despite their higher complexity, do not consistently outperform classical machine learning approaches, such as Random Forests (RF), in QSAR tasks—particularly when data is limited. Internal evaluations using industrial datasets support these findings; thus, we have incorporated only fingerprint-based classical machine learning models in our prioritization toolkit.

2.2.1 Descriptors

By default, QSAR/QSPR models utilize a concatenated fingerprint comprising:

- ECFP (2048 bits, radius 2, incorporating chirality and count),
- MACCS keys (159 bits, with count).

This combination results in a 2207-bit fingerprint, thoroughly validated internally, demonstrating strong performance across various public and industrial datasets. Nevertheless, alternative descriptors can easily be used within the code.

2.2.2 Random Forest (RF)

Random Forest is an ensemble learning method that constructs multiple decision trees using bootstrapped subsets of the training data and randomly selected features. The final prediction is obtained by averaging the outputs of individual trees, which enhances predictive accuracy and robustness. It is a widely used technique in QSAR/QSPR modeling. Our framework supports two methods for estimating uncertainty in RF models:

- **Standard deviation**: The standard deviation of predictions across all trees in the ensemble is used as a measure of uncertainty. A higher standard deviation indicates greater model uncertainty. Predictions are made using the mean output of all trees.

- **Bootstrap**: Multiple Random Forest models are trained on independently resampled datasets. Uncertainty is estimated as the standard deviation of predictions across these models. The default number of bootstrap replicates is 50, and predictions are made by averaging across all models.

2.2.3 XGBoost (XGB)

XGBoost (eXtreme Gradient Boosting) is a high-performance gradient boosting algorithm designed for speed and scalability. It incorporates regularization, parallel computation, and efficient handling of sparse data, making it popular in many machine learning applications. In our framework, uncertainty in XGBoost models is estimated using:

- **Quantile regression**: The model predicts specific quantiles of the target distribution (e.g., the 2.5th and 97.5th percentiles by default). The interval between these quantiles reflects predictive uncertainty. The median (50th percentile) is used as the point estimate for prediction.

2.2.4 LightGBM (LGBM)

LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework that employs histogram-based algorithms and leaf-wise tree growth to achieve faster training and reduced memory consumption. As with XGBoost, our implementation provides uncertainty estimates via bootstrapping and quantile regression.

2.2.5 Gaussian Processes (GP)

Gaussian Processes (GPs) are Bayesian, non-parametric models that define a distribution over functions, governed by a kernel (covariance) function measuring the similarity between input points. This probabilistic formulation inherently provides uncertainty estimates through the variance of the posterior predictive distribution. In our implementation, we use the Tanimoto kernel[17], defined as:

$$k_{\text{Tanimoto}}(\mathbf{x}, \mathbf{x}') = \frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{|\mathbf{x}|^2 + |\mathbf{x}'|^2 - \langle \mathbf{x}, \mathbf{x}' \rangle} \tag{8}$$

Table 4 QSAR/QSPR models implemented in the framework and their associated uncertainty quantification methods.

Model	Uncertainty estimation	Prediction method
Random Forest	Standard deviation of trees	Mean of all trees
Random Forest	Bootstrap	Mean of all bootstrapped models
LightGBM	Quantile regression (0.025 and 0.975 percentiles)	Median
LightGBM	Bootstrap	Mean of all bootstrapped models
XGBoost	Quantile regression (0.025 and 0.975 percentiles)	Median
XGBoost	Bootstrap	Mean of all bootstrapped models
Gaussian Process	Variance	Mean

2.3 Model-dependent acquisition functions

2.3.1 Desirability

The desirability acquisition function relies on the QSAR/QSPR models trained at each iteration. Each molecule in the pool set receives predicted endpoint values, from which a predicted desirability score D_{score} is computed. Normalization and aggregation of predicted values mirror the methodology applied to experimental desirability scoring, using the utility functions and a weighted geometric mean. Molecules with the highest predicted desirability scores are subsequently selected.

2.3.2 Desirability (with applicability domain)

This variant extends the basic desirability approach by introducing an applicability domain (AD) filter, defined by molecular similarity. Molecules are considered within the AD if their maximum similarity to any training set molecule exceeds a specified threshold (default: 0.7). Prioritization occurs first among molecules within the AD based on predicted desirability scores. If fewer molecules fall within the domain than required for batch selection, remaining slots are filled by out-of-domain molecules according to their predicted desirability scores.

2.3.3 Uncertainty

The uncertainty acquisition function is widely employed in active learning to select samples with the highest predictive uncertainty. In multi-objective optimization, however, combining uncertainties across several endpoints can be challenging. To address this, we compute an uncertainty score U_{score} in a manner analogous to the desirability score. First, predicted uncertainties for all modeled

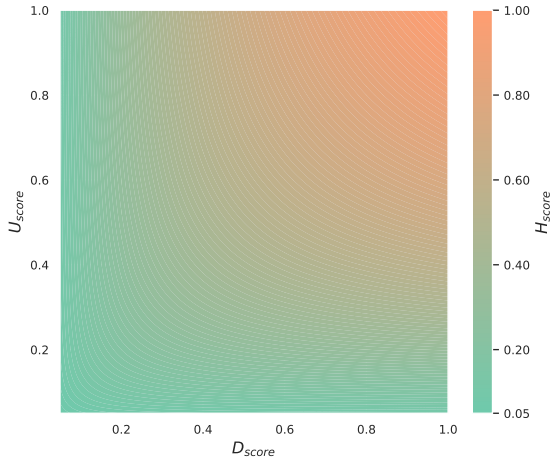

Fig. 1 U/D Harmonic score distribution based on U_{score} and D_{score}.

endpoints i are obtained. These uncertainties are then normalized using MinMax scaling with a lower threshold of 0.05 (Eq. 9). Finally, the normalized uncertainties are combined with a weighted geometric mean (Eq. 10) to yield a final uncertainty score for each compound.

$$u_i(x) = \max\left(\frac{x_i - \min(X_i)}{\max(X_i) - \min(X_i)}, 0.05\right) \tag{9}$$

where x_i is the predicted uncertainty on endpoint i for a compound x, X_i is the set of all predicted uncertainties on endpoint i for all molecules.

$$U_{score}(x) = \left(\prod_{i=1}^{n} u_i(x)^{w_i}\right)^{\frac{1}{\sum_{i=1}^{n}\omega_i}} \tag{10}$$

where n is the number of endpoints and ω_i the weight associated to endpoint i.

2.3.4 U/D Harmonic

The U/D Harmonic acquisition function combines the predicted desirability and uncertainty scores into a harmonic score (H_{score}) using the harmonic mean (Eq. 11). This approach balances exploitation and exploration by highlighting compounds that are both promising and uncertain.

$$H_{score}(x) = \frac{2 \times U_{score}(x) \times D_{score}(x)}{U_{score}(x) + D_{score}(x)} \tag{11}$$

where $U_{score}(x)$ and $D_{score}(x)$ are respectively the predicted uncertainty and predicted desirability scores of a molecule x.

2.3.5 U/D Ratio

The U/D Ratio acquisition function combines predicted desirability and prediction uncertainty scores into a ratio score (R_{score}) using an adjusted ratio (Eq. 12). Since the desirability lowest possible value is 0.95, the adjustment constant ε is set to 0.95 to have a maximum R_{score} value of 1. The ratio score balances exploitation and exploration by prioritizing promising uncertain molecules that are more or less desirable, depending on the ε value. Higher values of epsilon will promote highly desirable molecules.

$$R_{score}(x) = \frac{U_{score}(x)}{\varepsilon + D_{score}(x)} \tag{12}$$

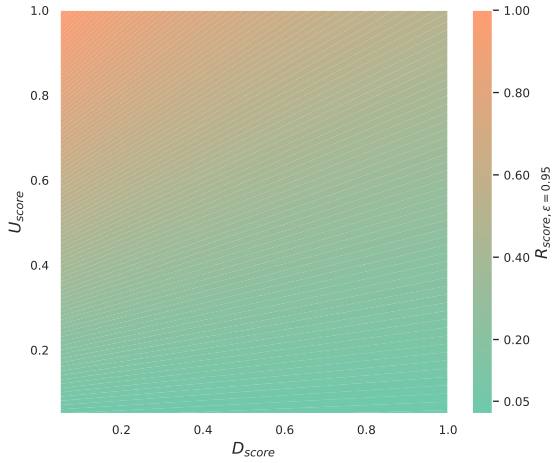

Fig. 2 U/D Ratio scores distribution based on U_{score} and D_{score} for $\varepsilon = 0.95$.

2.3.6 Desired Coverage

The desired coverage acquisition proposes a trade-off between exploration and exploitation by combining the predicted desirability score and the coverage score to create a desired coverage score (DC_{score}). Using a greedy approach, it optimizes a composite score for a subset of molecules that is its coverage score multiplied by its mean predicted desirability score (Eq. 13). This approach favors batches with high desirability and high coverage.

$$DC_{score}(S) = \sum_{x \in S} C_{comp}(x) \times \frac{1}{|S|} \sum_{x \in S} D_{score}(x) = C_{score}(S) \times D_{score}(S) \tag{13}$$

where S is the set of selected compounds, $C_{comp}(x)$ and $D_{score}(x)$ are respectively the compound-level coverage score and desirability score of compound $x \in S$. $C_{score}(S)$ is the coverage score of set S and $D_{score}(S)$ is the average predicted desirability score of compounds in S.

2.3.7 Desired Spread

The desired spread acquisition function proposes a trade-off between exploration and exploitation by combining the predicted desirability score and the spread strategy to create a desired spread score (Eq. 14). It is inspired by the demerit spread design strategy developed by Higgs *et al.* [18] using the desirability score as a scaling factor to the distance between the pool set and its nearest neighbors from the train set or previously selected compounds.

$$DS_{score}(x) = D_{score}(x) \times S_{score}(x) \tag{14}$$

where $D_{score}(x)$ and $S_{score}(x)$ are respectively the desirability score and spread score of compound x.

2.3.8 Greediverse

The greediverse acquisition function is based on the work of Langevin *et al.*. It optimizes the desirability scores within a batch, while yielding a penalty if the similarity of two molecules from the batch is above a defined threshold. The function is optimized using a greedy iterative approach as mentioned in the original paper.

2.3.9 TOPSIS

The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [9] is a widely used method in multiple-criteria decision analysis (MCDA). It ranks alternatives across multiple, often conflicting criteria by identifying solutions that are closest to an ideal option while remaining farthest from the least desirable one.

In its standard formulation, TOPSIS assumes that all criteria either increase or decrease monotonically, which justifies the common use of L2 normalization. However, in drug discovery, certain endpoints require maximization, others minimization, and some must fall within a specific range. This variability renders L2 normalization inappropriate. To address this limitation, we propose a modified TOPSIS approach that replaces L2 normalization with the utility functions previously described. These functions rescale endpoint values to a range between 0.05 and 1, ensuring a more suitable representation of diverse optimization objectives.

The first step involves normalizing the predicted values fore each endpoint i across all molecules j within the pool set using utility functions described in the main manuscript. A normalized decision matrix $m_{j,i}$ is constructed with these normalized values.

This matrix is then transformed into a weighted version, denoted as $m^*_{j,i}$:

$$m^*_{j,i} = m_{j,i} \times \frac{\omega_i}{\sum_{i=1}^{n} \omega_i} \tag{15}$$

where ω_i is the weight associated with endpoint i and n is the total number of endpoints.

Next, the positive ideal solution p_i^{ideal} and negative ideal solution n_i^{ideal} are determined for each endpoint by identifying the maximum and minium values in the weighted matrix, respectively: :

$$p_i^{ideal} = \max_j \left(m^*_{j,i}\right), \qquad n_i^{ideal} = \min_j \left(m^*_{j,i}\right) \tag{16}$$

The Euclidean distance of each molecule to these ideal solutions are then computed as follows:

$$dist_j^+ = \sqrt{\sum_{i=1}^{n} \left(m^*_{j,i} - p_i^{ideal}\right)^2} \qquad dist_j^- = \sqrt{\sum_{i=1}^{n} \left(m^*_{j,i} - n_i^{ideal}\right)^2} \tag{17}$$

where n is the number of endpoints.

Finally, the relative closeness of a molecule j to the ideal solution is calculcated:

$$c_j = \frac{dist_j^-}{dist_j^+ + dist_j^-} \tag{18}$$

For selection, compounds having the highest relative closeness are prioritized.

2.3.10 GRA

Grey Relational Analysis (GRA)[4] is a prominent Multi-Criteria Decision Analysis (MCDA) method developed within the framework of Grey System Theory. It is particularly effective in handling uncertain, incomplete, or imprecise information, hence the term "grey". Unlike conventional methods that rely on fully defined datasets, GRA can accommodate varying data quality, making it especially useful in real-world decision-making scenarios. Similar to the TOPSIS algorithm, GRA evaluates alternatives based on their similarity to an ideal solution.

Originally, GRA employs Min-Max normalization for attributes that should be maximized and Max-Min normalization for those that should be minimized. However, like TOPSIS, it does not inherently account for criteria that should be maintained within a specific range. To address this limitation, we define a normalized decision matrix $m_{j,i}$ similarly to the TOPSIS method, using utility functions.

Next, the deviation sequence $\Delta_{j,i}$ is computed:

$$\Delta_{j,i} = 1 - m_{j,i} \tag{19}$$

Then, the grey relational coefficient $\varphi_{j,i}$ is calculated for each data point using an ε parameter, typically set to 0.5:

$$\varphi_{j,i} = \frac{\varepsilon}{\Delta_{j,i} + \varepsilon} \tag{20}$$

where $\Delta_{j,i}$ is the deviation sequence of molecule j on endpoint i.

Finally, the grey relational coefficients for different endpoints are aggregated into a grey relational grade using a weighted arithmetic mean:

$$GRG_j = \frac{1}{n} \sum_{i=1}^{n} \varphi_{j,i} \cdot \omega_i \tag{21}$$

where n is the number of endpoints, ω_i the weight associated to endpoint i and $\varphi_{j,i}$ the grey relational coefficient for molecule j on endpoint i.

Compounds with the highest grey relational grades are prioritized for selection.

2.4 Oracle acquisition function

The oracle acquisition function operates as an omniscient strategy, having access to the complete set of experimental data at each stage of the selection process. Consequently, it identifies and selects the compounds with the highest experimental desirability scores from the pool set at every iteration. This strategy leads to optimal exploitation to which other strategies can be compared.

3 FXa

This section display additional figures for the FXa dataset:

- Impact of the memory effect
- Impact of the QSAR model type (Desirability strategy)
- Impact of the QSAR model type (Uncertainty strategy)

3.1 Impact of the memory effect

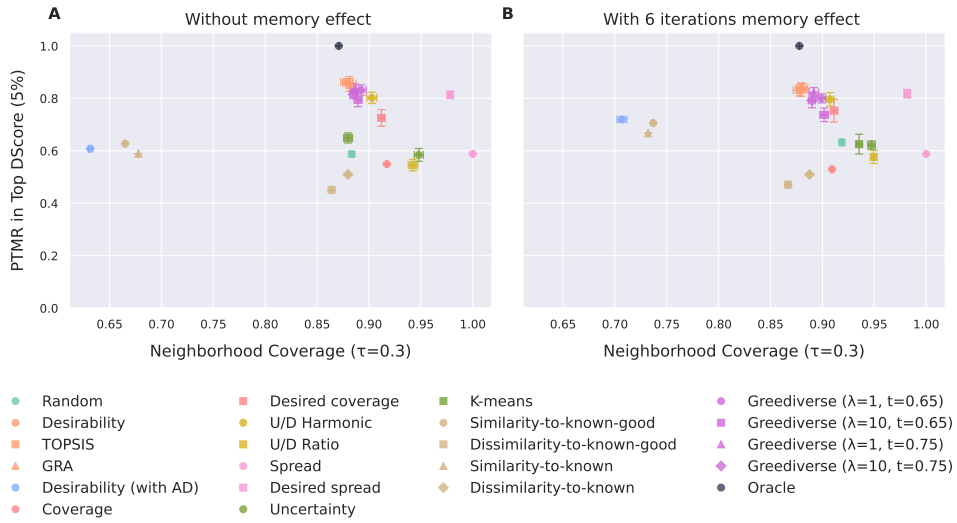

Fig. 3 Final overview of Exploitation vs. Exploration with and without memory effect (FXa dataset). The panels display results without memory effect (A) and with a 6 iterations memory effect (B). The 95% confidence intervals are displayed, calculated from 100 replicates for the Random baseline and 10 replicates for all other acquisition strategies.

3.2 Impact of the QSAR model type (Desirability strategy

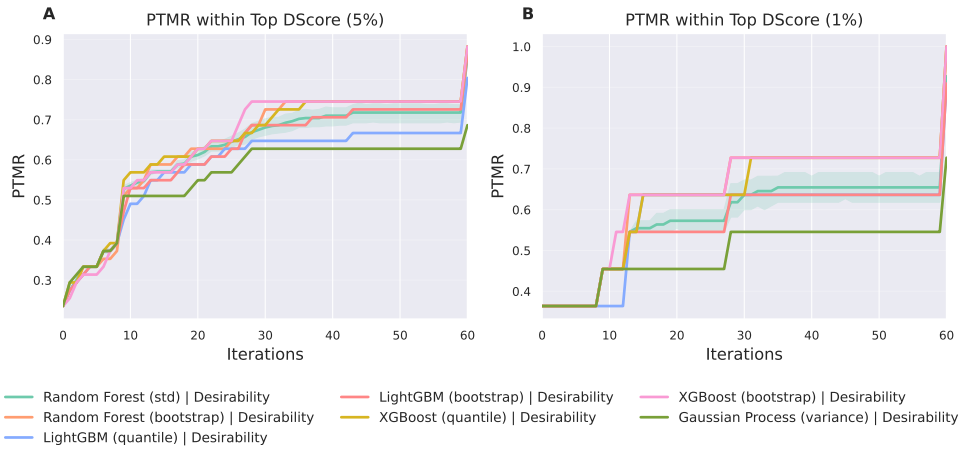

Fig. 4 Evolution of Neighborhood Coverage with $\tau = 0.3$ (A), Neighborhood Coverage AUC (B), Internal Diversity (C) and #Circles (D) exploration metrics over iterations for the Desirability acquisition function on the FXa dataset. Simulations were done using various QSAR/QSPR models, each with different uncertainty estimation and prediction methods: Random Forest (with either standard deviation of trees or bootstrap), LightGBM (with either quantile regression or bootstrap) and Gaussian Process (posterior variance).

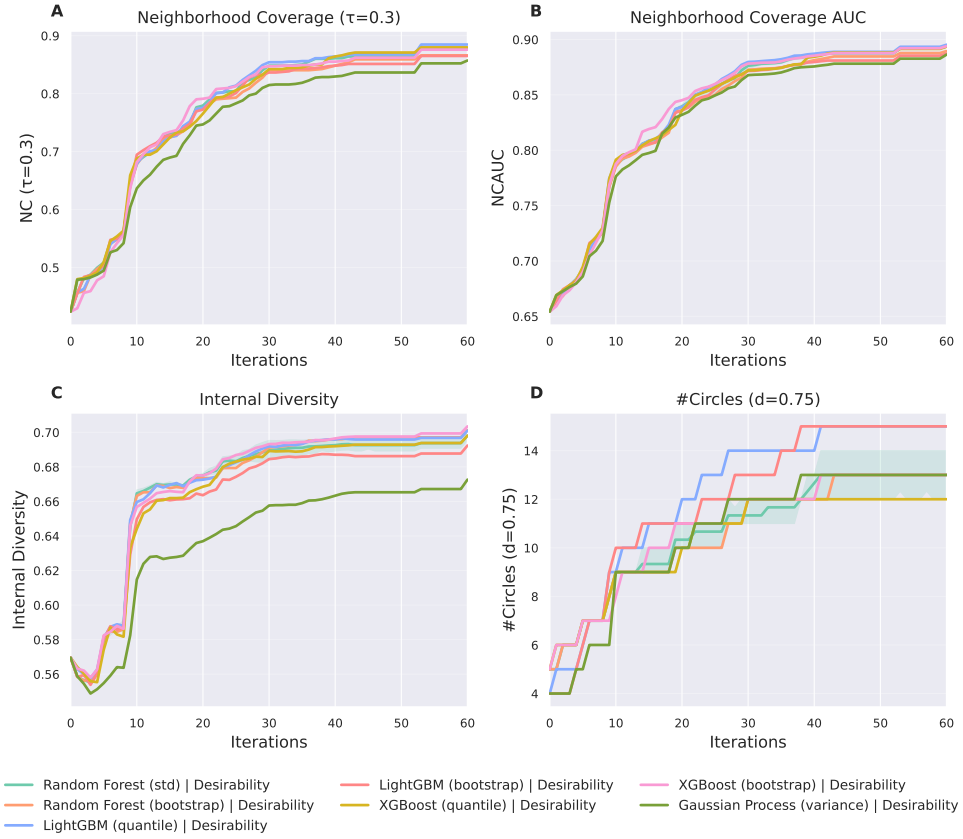

Fig. 5 Evolution of Proportion of Top Molecules Retrieved (PTMR) monitoring Top DScore 5% (A) and Top DScore 1% (B) categories for the Desirability acquisition function on the FXa dataset. Simulations were done using various QSAR/QSPR models, each with different uncertainty estimation and prediction methods: Random Forest (with either standard deviation of trees or bootstrap), LightGBM (with either quantile regression or bootstrap) and Gaussian Process (posterior variance).

3.3 Impact of the QSAR model type (Uncertainty strategy

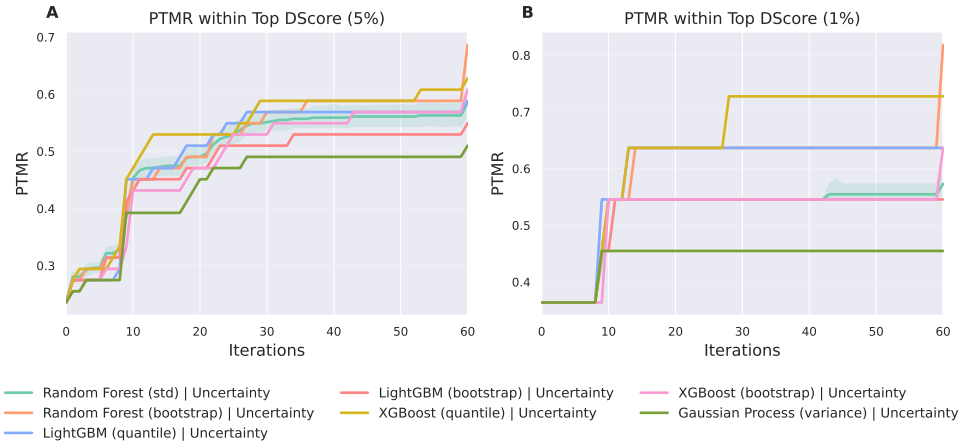

Fig. 6 Evolution of Neighborhood Coverage with $\tau = 0.3$ (A), Neighborhood Coverage AUC (B), Internal Diversity (C) and #Circles (D) exploration metrics over iterations for the Uncertainty acquisition function on the FXa dataset. Simulations were done using various QSAR/QSPR models, each with different uncertainty estimation and prediction methods: Random Forest (with either standard deviation of trees or bootstrap), LightGBM (with either quantile regression or bootstrap) and Gaussian Process (posterior variance).

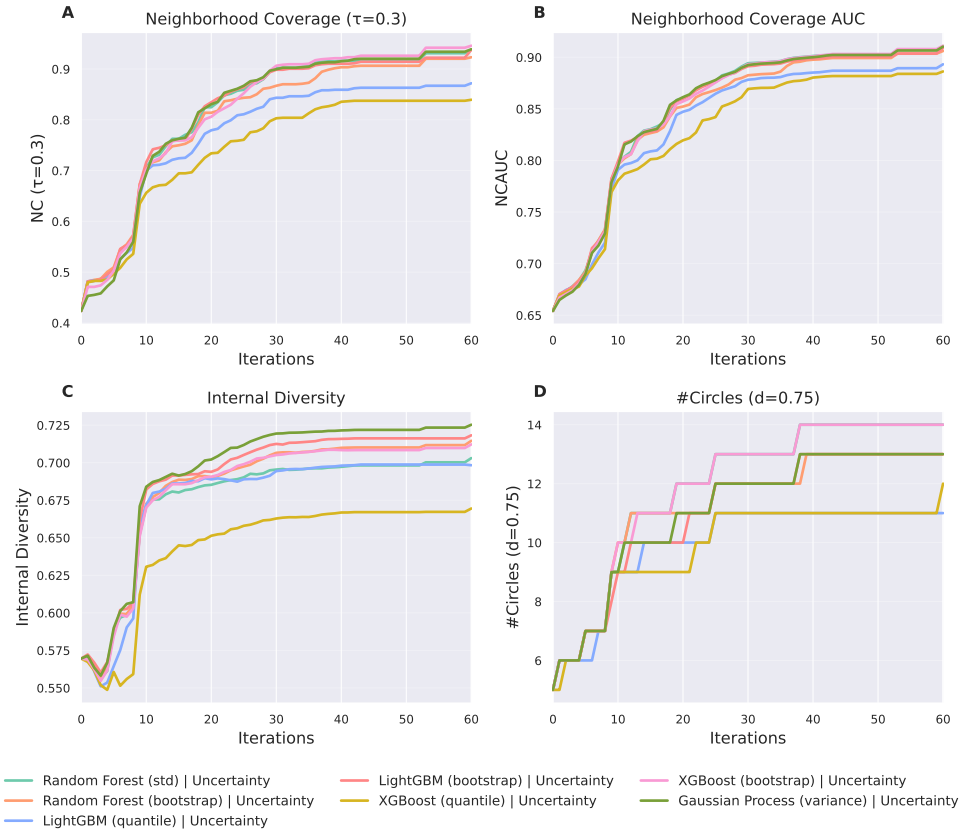

Fig. 7 Evolution of Proportion of Top Molecules Retrieved (PTMR) monitoring Top DScore 5% (A) and Top DScore 1% (B) categories for the Uncertainty acquisition function on the FXa dataset. Simulations were done using various QSAR/QSPR models, each with different uncertainty estimation and prediction methods: Random Forest (with either standard deviation of trees or bootstrap), LightGBM (with either quantile regression or bootstrap) and Gaussian Process (posterior variance).

4 Renin

This section display additional figures for the Renin dataset:

- Top-tier categories
- Timeline
- Initial compounds and chemical series KDE TMAP
- Final overview KDE TMAP
- Exploration
- Exploitation
- Exploitation / Exploration trade-off
- Impact of the memory effect

4.1 Top-tier categories

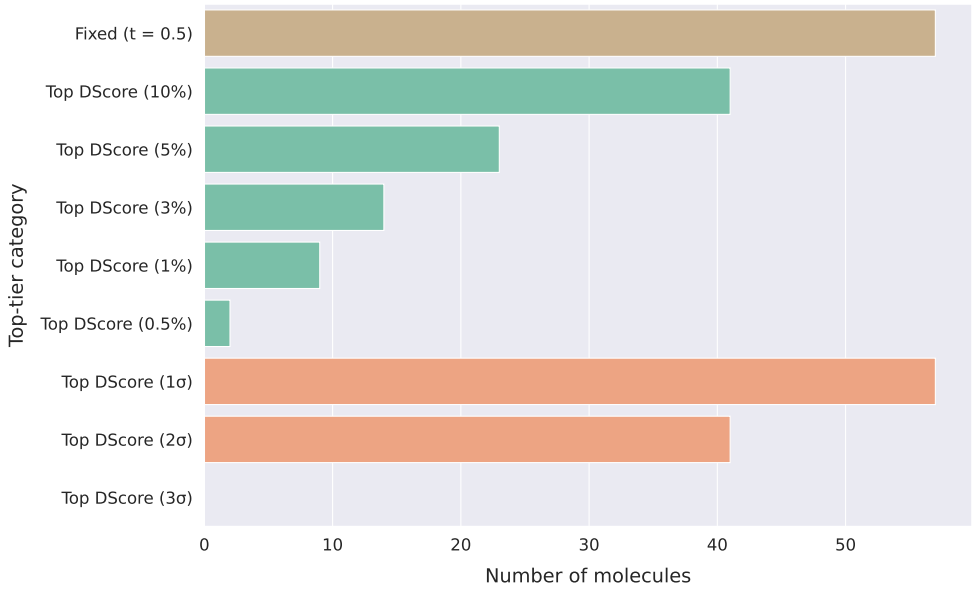

Fig. 8 Number of molecules classified as "top-tier" across different threshold definitions (Renin dataset). The definitions are grouped by methodology: brown indicates fixed desirability thresholds, orange indicates percentile-based thresholds, and green indicates standard deviation-based thresholds.

4.2 Timeline

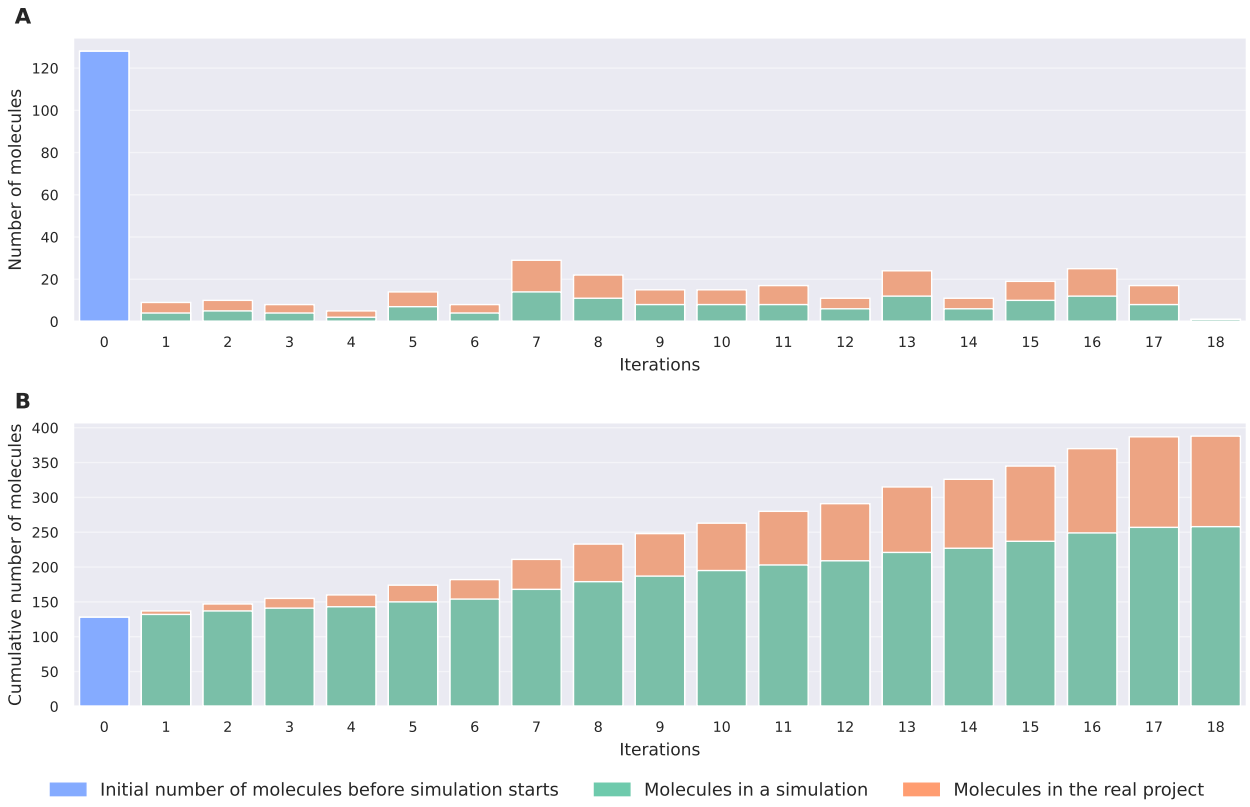

Fig. 9 Comparison of compound selection between the real project and a simulation (Renin dataset). (A) Number of compounds selected per iteration. (B) Cumulative number of compounds selected over time.

4.3 Initial compounds and chemical series KDE TMAP

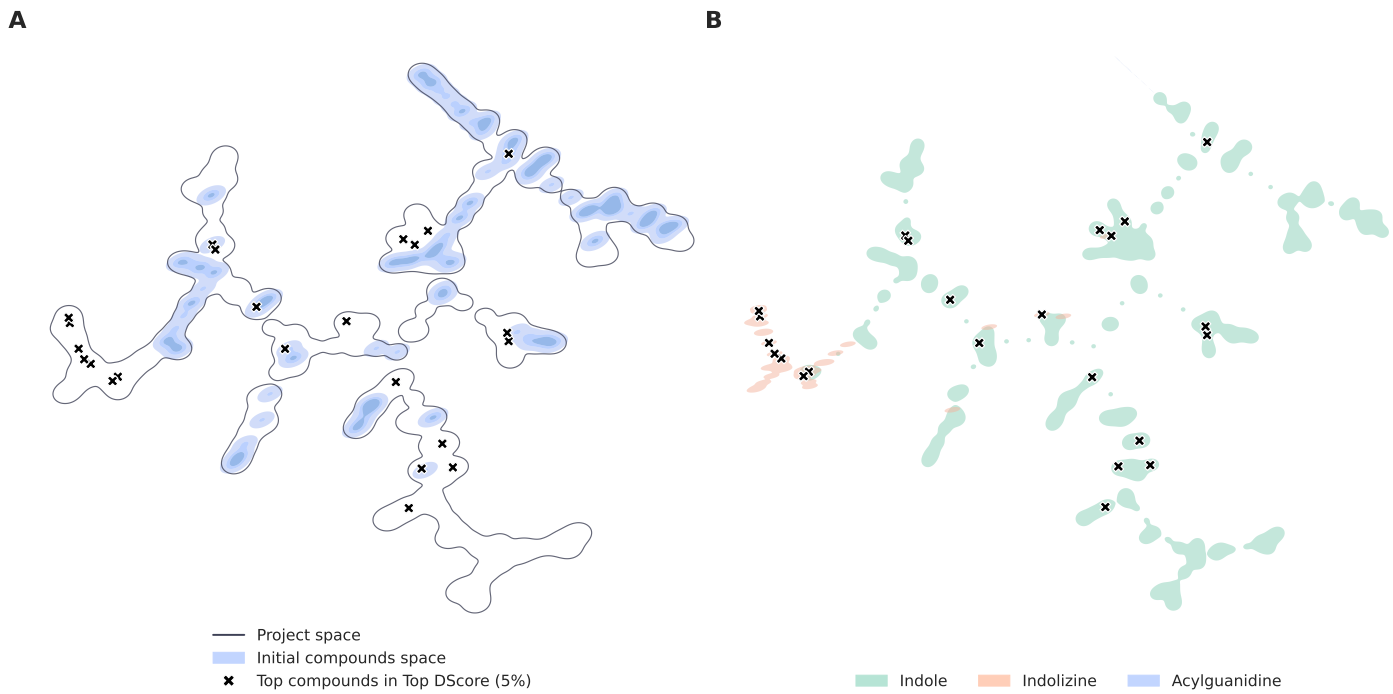

Fig. 10 Overview of the Renin dataset chemical space. (A) Global view of the chemical space at the start of the simulation. (B) Distributions of the major chemical series. (C) Representative chemical structures of top-tier compounds (top 5% DScore) from various series. The box colors in (C) correspond to the chemical series colors used in (B).

4.4 Final overview KDE TMAP

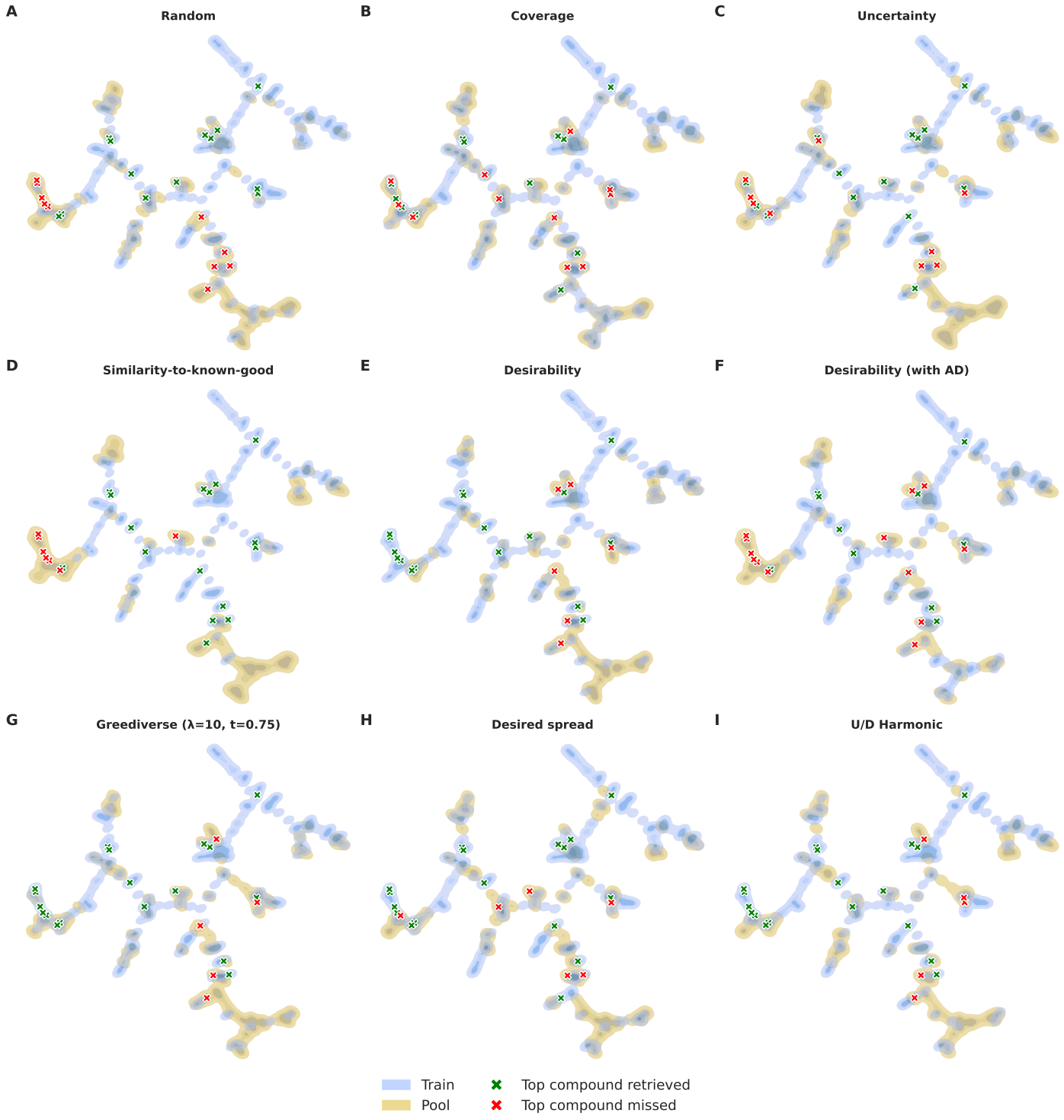

Fig. 11 TMAP visualizations of the final chemical space across nine acquisition functions (Renin dataset). Regions: Blue areas represent the sampled subspace (selected molecules), while yellow areas indicate the remaining unselected pool. Markers: Crosses identify top-tier compounds (top 5% DScore). Green crosses denote top-tier molecules that were successfully retrieved, whereas red crosses denote those that were missed.

4.5 Exploration

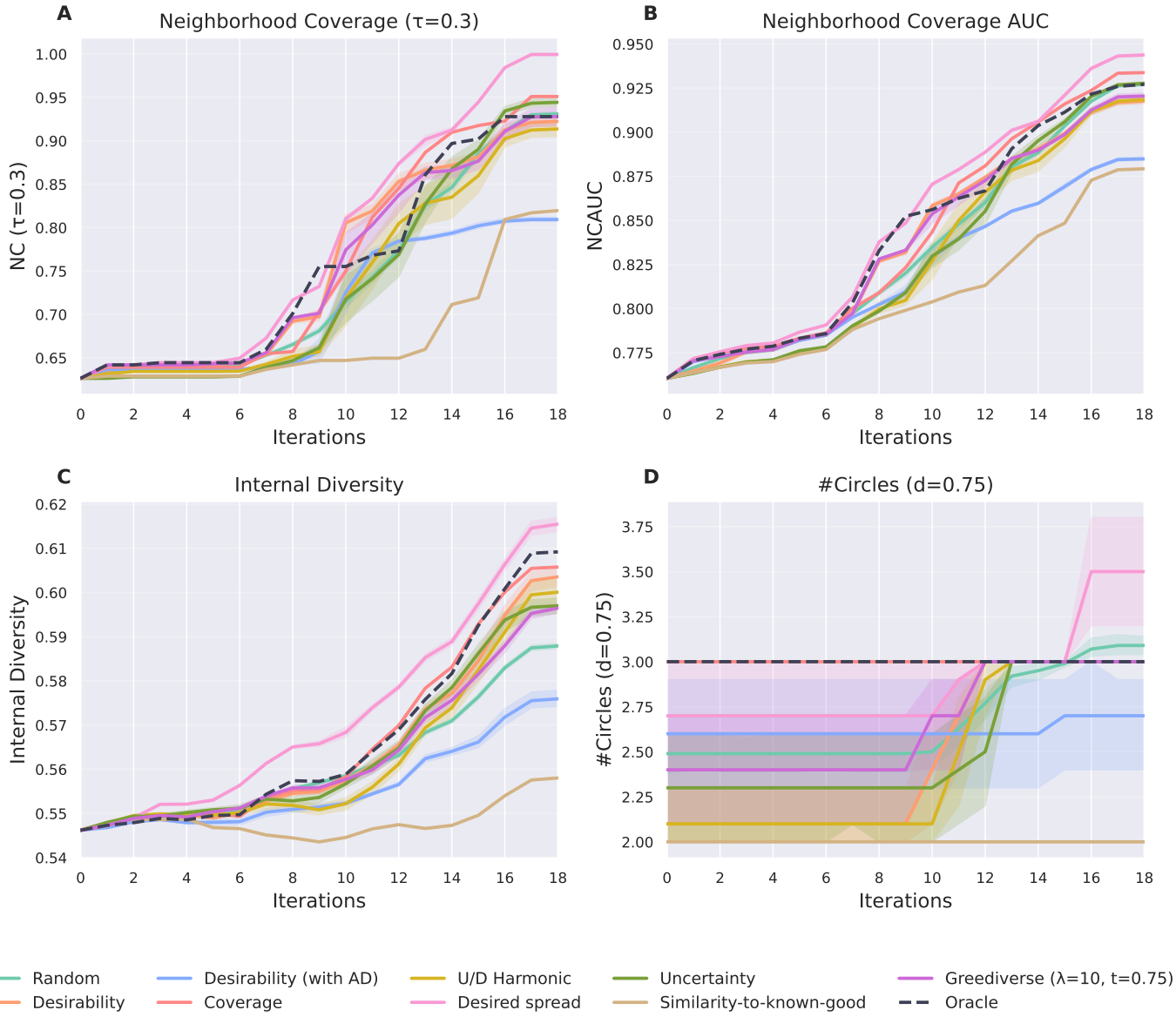

Fig. 12 Exploration metrics over iterations for selected acquisition functions (Renin dataset). (A) Evolution of Neighborhood Coverage ($\tau = 0.3$). (B) Neighborhood Coverage AUC. (C) Internal Diversity. (D) Number of Circles (#Circles). Shaded regions indicate the 95% confidence interval. These intervals are calculated from 100 replicates for the Random strategy and 10 replicates for all other strategies.

4.6 Exploitation

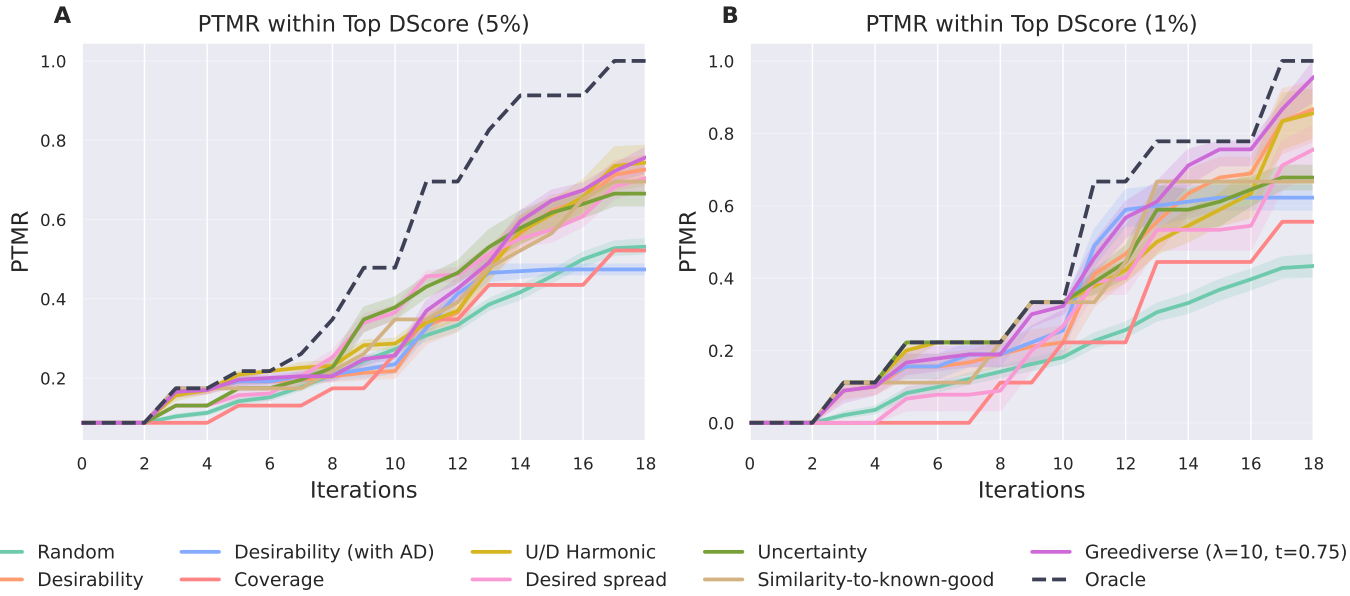

Fig. 13 Exploitation performance of various acquisition functions (Renin dataset). The plots track the retrieval of top-tier compounds defined by two stringency levels: (A) the top 5% DScore and (B) the top 1% DScore categories. Shaded regions indicate the 95% confidence interval. These intervals are calculated from 100 replicates for the Random strategy and 10 replicates for all other strategies.

4.7 Exploitation / Exploration trade-off

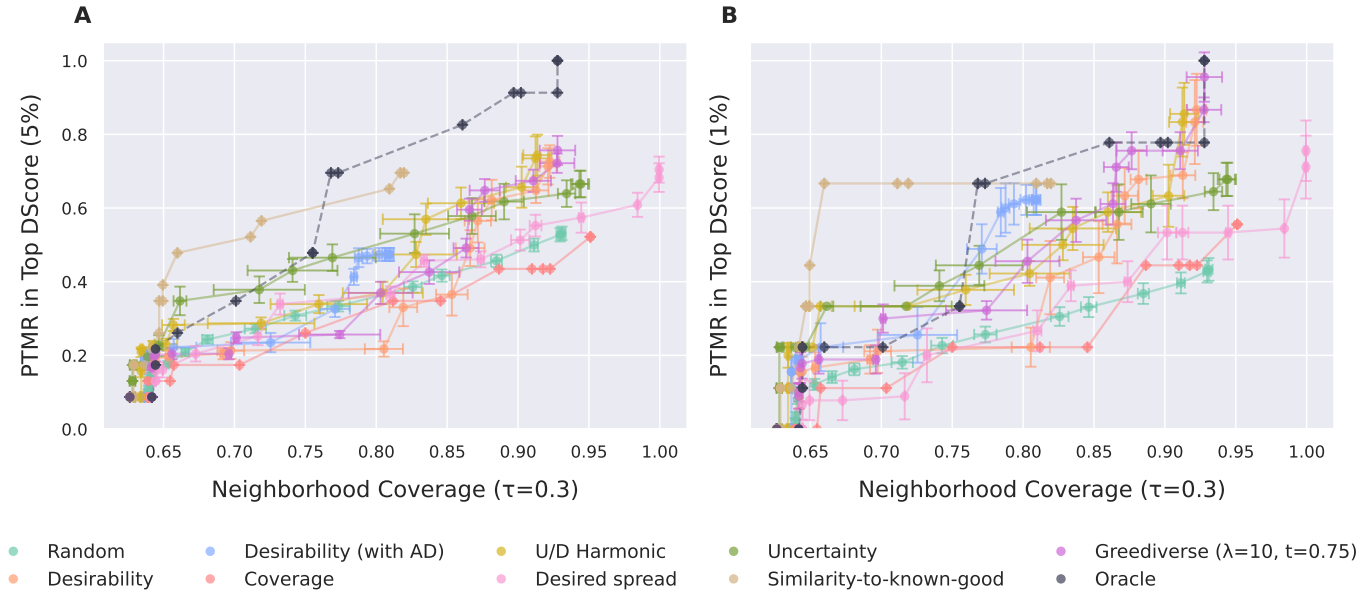

Fig. 14 Trajectories of Exploitation vs. Exploration (Renin dataset). Performance is mapped by plotting Exploitation (monitored via PTMR within the top 5% DScore and 1% DScore) against Exploration (monitored via Neighborhood Coverage at $\tau = 0.3$). Shaded regions indicate the 95% confidence interval, calculated from 100 replicates for the Random baseline and 10 replicates for all other acquisition strategies.

4.8 Impact of the memory effect

Fig. 15 Final overview of Exploitation vs. Exploration with and without memory effect (Renin dataset). The panels display results without memory effect (A) and with a 6 iterations memory effect (B). The 95% confidence intervals are displayed, calculated from 100 replicates for the Random baseline and 10 replicates for all other acquisition strategies.

5 PPARδ

This section display additional figures for the PPARδ dataset:

- Top-tier categories
- Timeline
- Initial compounds and chemical series KDE TMAP
- Final overview KDE TMAP
- Exploration
- Exploitation
- Exploitation / Exploration trade-off
- Impact of the memory effect

5.1 Top-tier categories

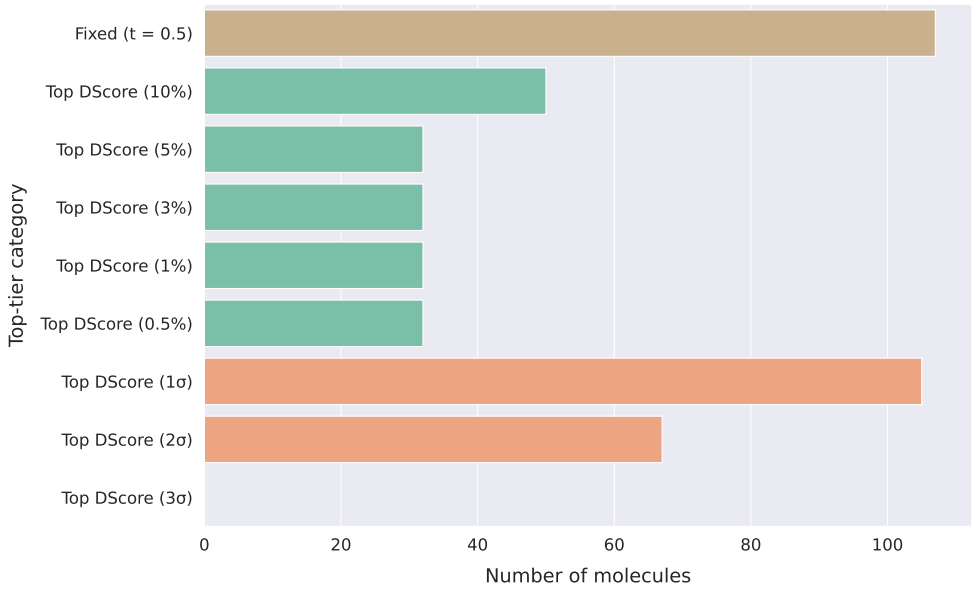

Fig. 16 Number of molecules classified as "top-tier" across different threshold definitions (PPARδ dataset). The definitions are grouped by methodology: brown indicates fixed desirability thresholds, orange indicates percentile-based thresholds, and green indicates standard deviation-based thresholds.

5.2 Timeline

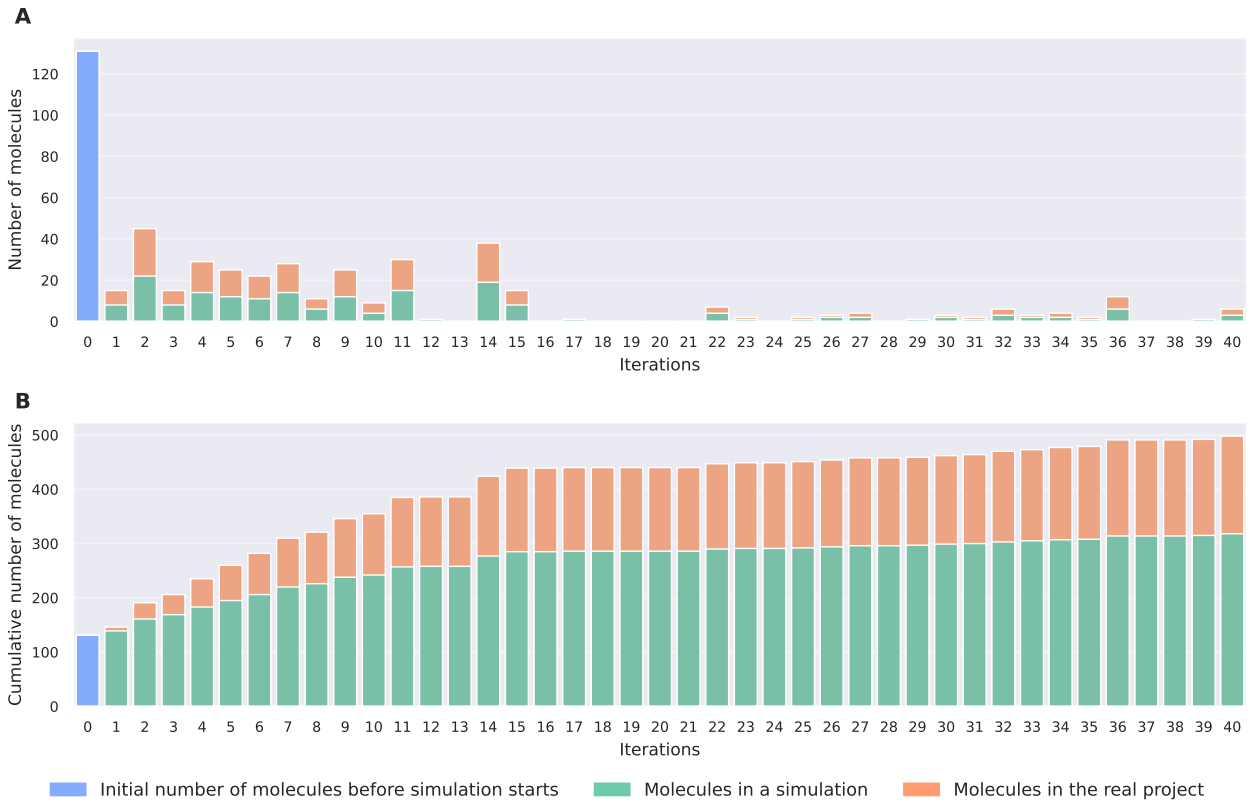

Fig. 17 Comparison of compound selection between the real project and a simulation (PPARδ dataset). (A) Number of compounds selected per iteration. (B) Cumulative number of compounds selected over time.

5.3 Initial compounds and chemical series KDE TMAP

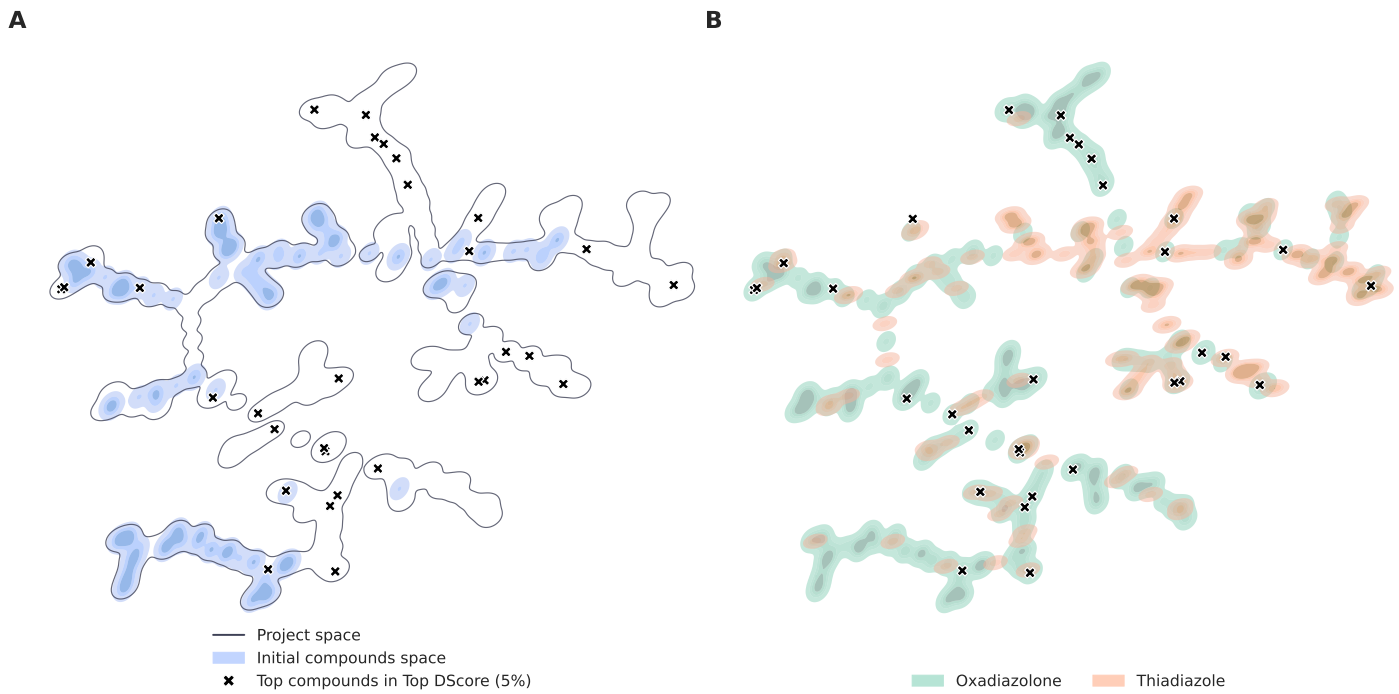

Fig. 18 Overview of the PPARδ dataset chemical space. (A) Global view of the chemical space at the start of the simulation. (B) Distributions of the major chemical series. (C) Representative chemical structures of top-tier compounds (top 5% DScore) from various series. The box colors in (C) correspond to the chemical series colors used in (B).

5.4 Final overview KDE TMAP

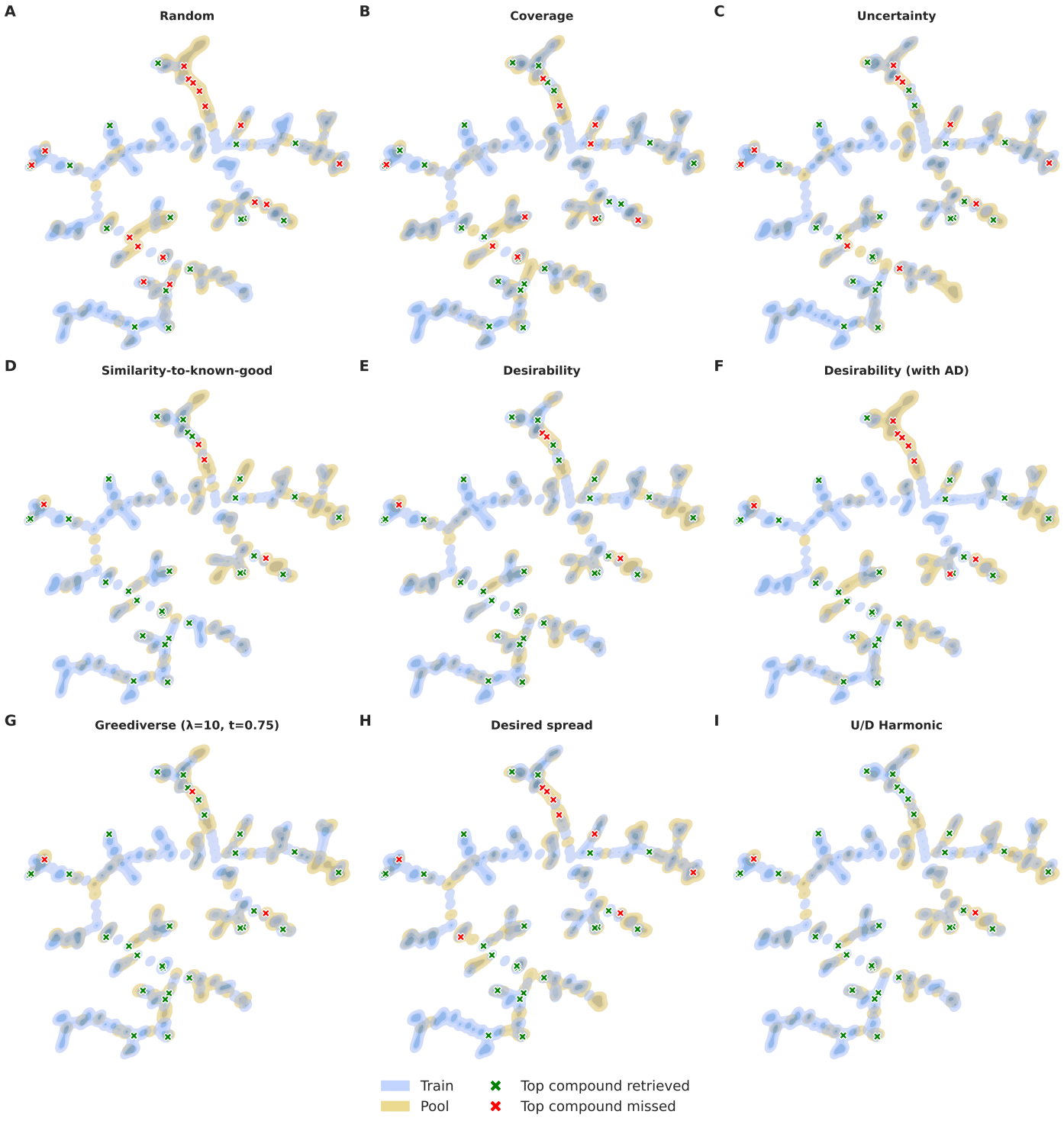

Fig. 19 TMAP visualizations of the final chemical space across nine acquisition functions (PPARδ dataset). Regions: Blue areas represent the sampled subspace (selected molecules), while yellow areas indicate the remaining unselected pool. Markers: Crosses identify top-tier compounds (top 5% DScore). Green crosses denote top-tier molecules that were successfully retrieved, whereas red crosses denote those that were missed.

5.5 Exploration

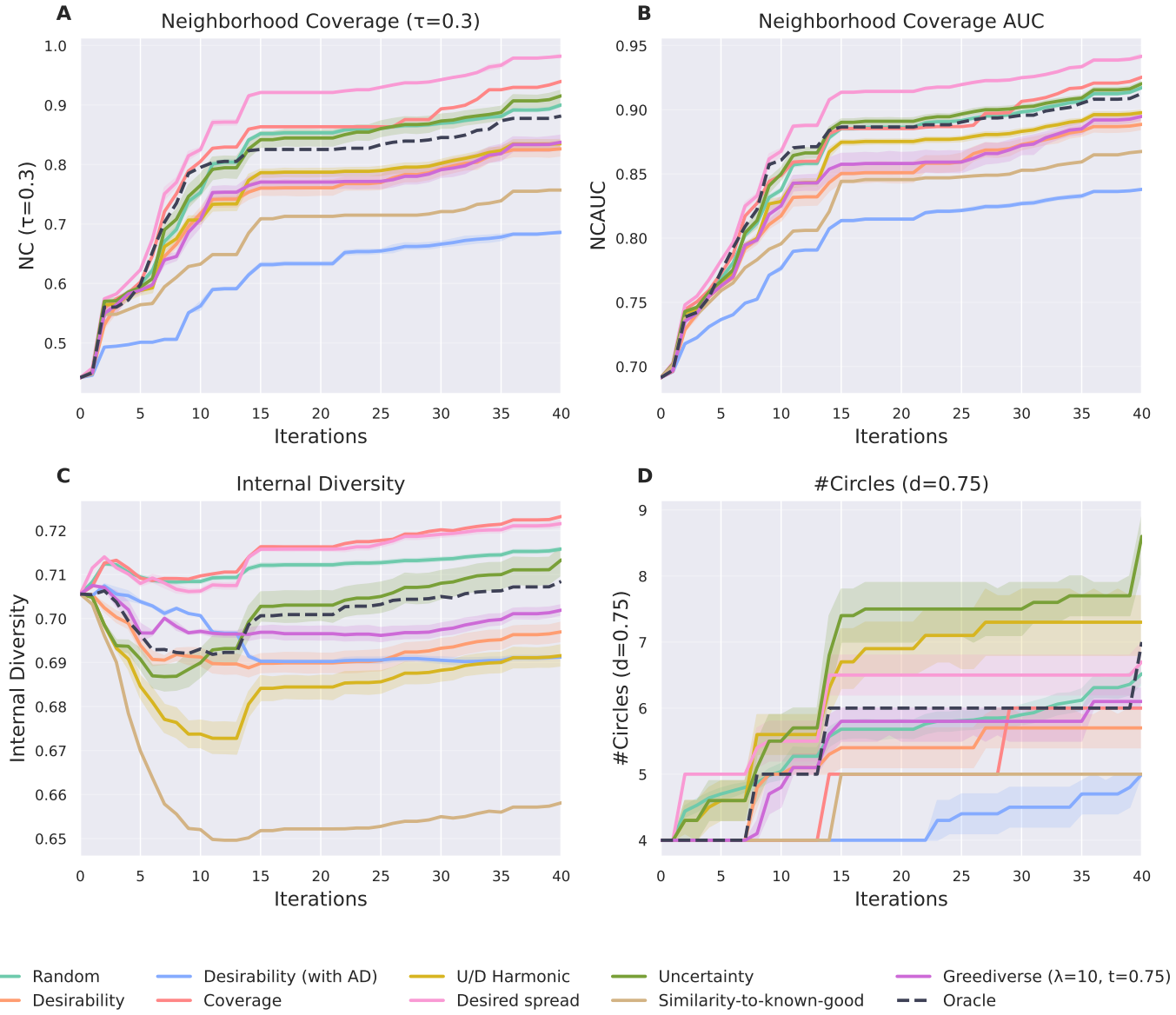

Fig. 20 Exploration metrics over iterations for selected acquisition functions (PPARδ dataset). (A) Evolution of Neighborhood Coverage ($\tau = 0.3$). (B) Neighborhood Coverage AUC. (C) Internal Diversity. (D) Number of Circles (#Circles). Shaded regions indicate the 95% confidence interval. These intervals are calculated from 100 replicates for the Random strategy and 10 replicates for all other strategies.

5.6 Exploitation

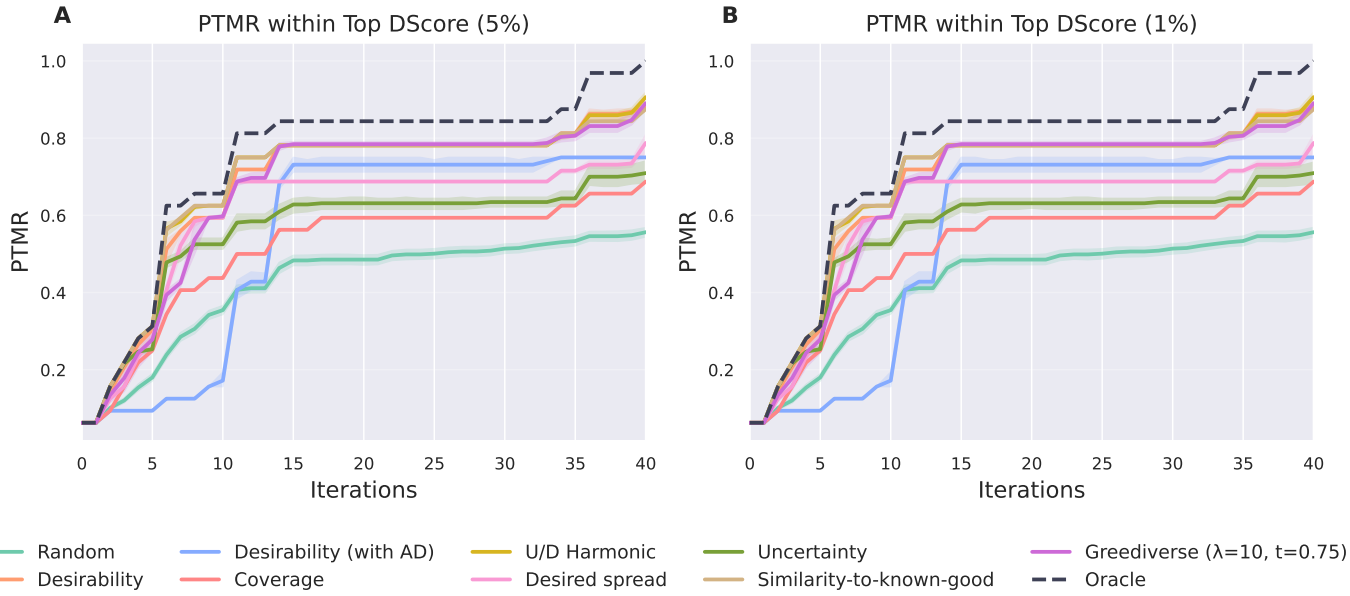

Fig. 21 Exploitation performance of various acquisition functions (PPARδ dataset). The plots track the retrieval of top-tier compounds defined by two stringency levels: (A) the top 5% DScore and (B) the top 1% DScore categories. Shaded regions indicate the 95% confidence interval. These intervals are calculated from 100 replicates for the Random strategy and 10 replicates for all other strategies.

5.7 Exploitation / Exploration trade-off

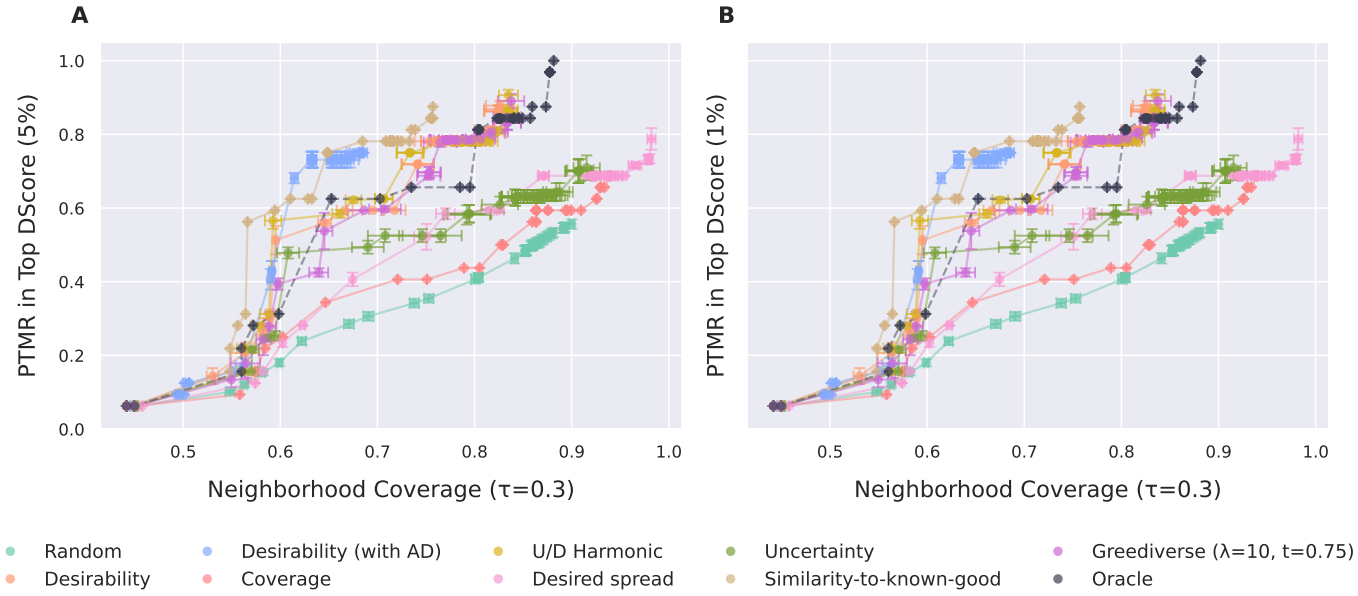

Fig. 22 Trajectories of Exploitation vs. Exploration (PPARδ dataset). Performance is mapped by plotting Exploitation (monitored via PTMR within the top 5% DScore and 1% DScore) against Exploration (monitored via Neighborhood Coverage at $\tau = 0.3$). Shaded regions indicate the 95% confidence interval, calculated from 100 replicates for the Random baseline and 10 replicates for all other acquisition strategies.

5.8 Impact of the memory effect

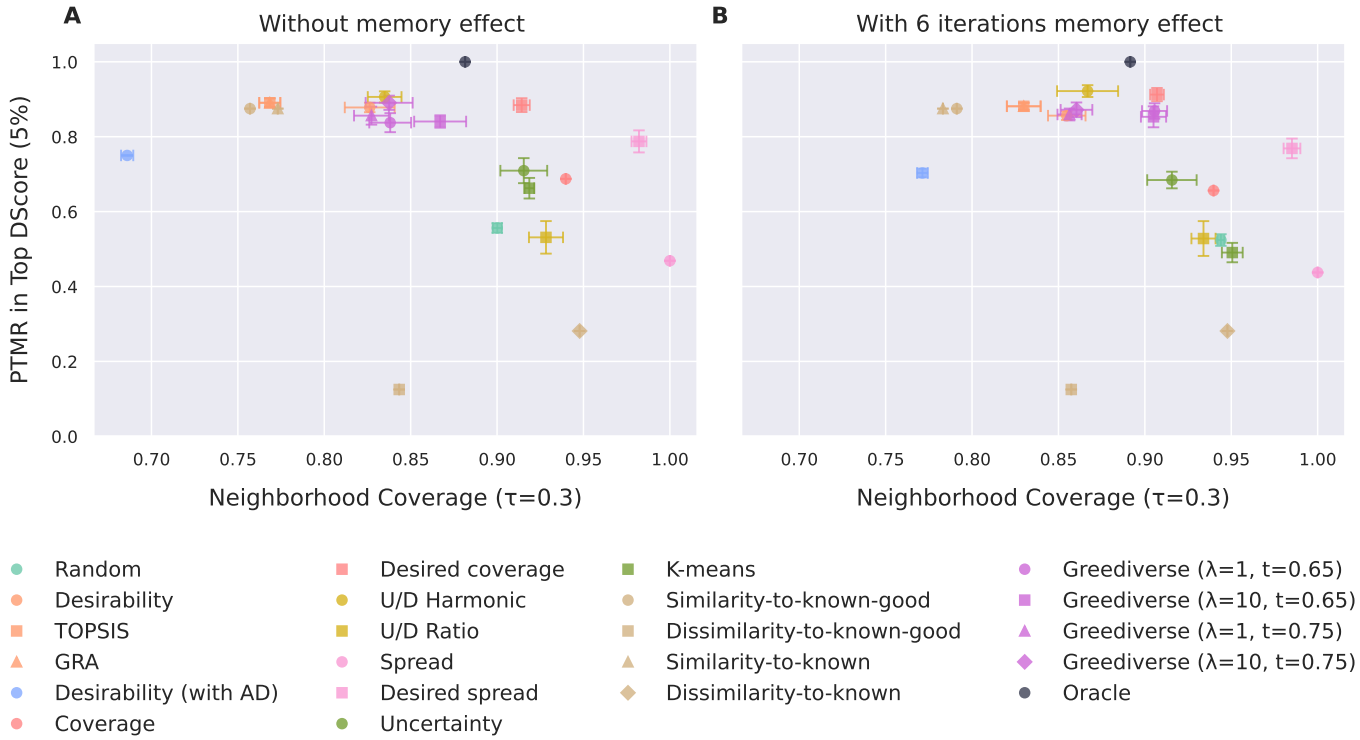

Fig. 23 Final overview of Exploitation vs. Exploration with and without memory effect (PPARδ dataset). The panels display results without memory effect (A) and with a 6 iterations memory effect (B). The 95% confidence intervals are displayed, calculated from 100 replicates for the Random baseline and 10 replicates for all other acquisition strategies.

6 MMP-8

This section display additional figures for the MMP-8 dataset:

- Top-tier categories
- Timeline
- Initial compounds and chemical series KDE TMAP
- Final overview KDE TMAP
- Exploration
- Exploitation
- Exploitation / Exploration trade-off
- Impact of the memory effect

6.1 Top-tier categories

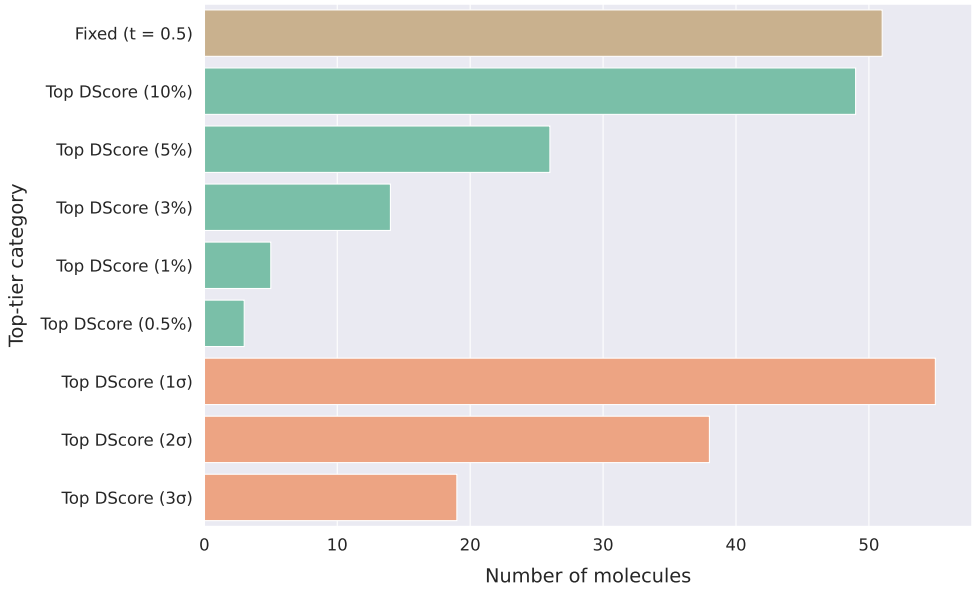

Fig. 24 Number of molecules classified as "top-tier" across different threshold definitions (MMP-8 dataset). The definitions are grouped by methodology: brown indicates fixed desirability thresholds, orange indicates percentile-based thresholds, and green indicates standard deviation-based thresholds.

6.2 Timeline

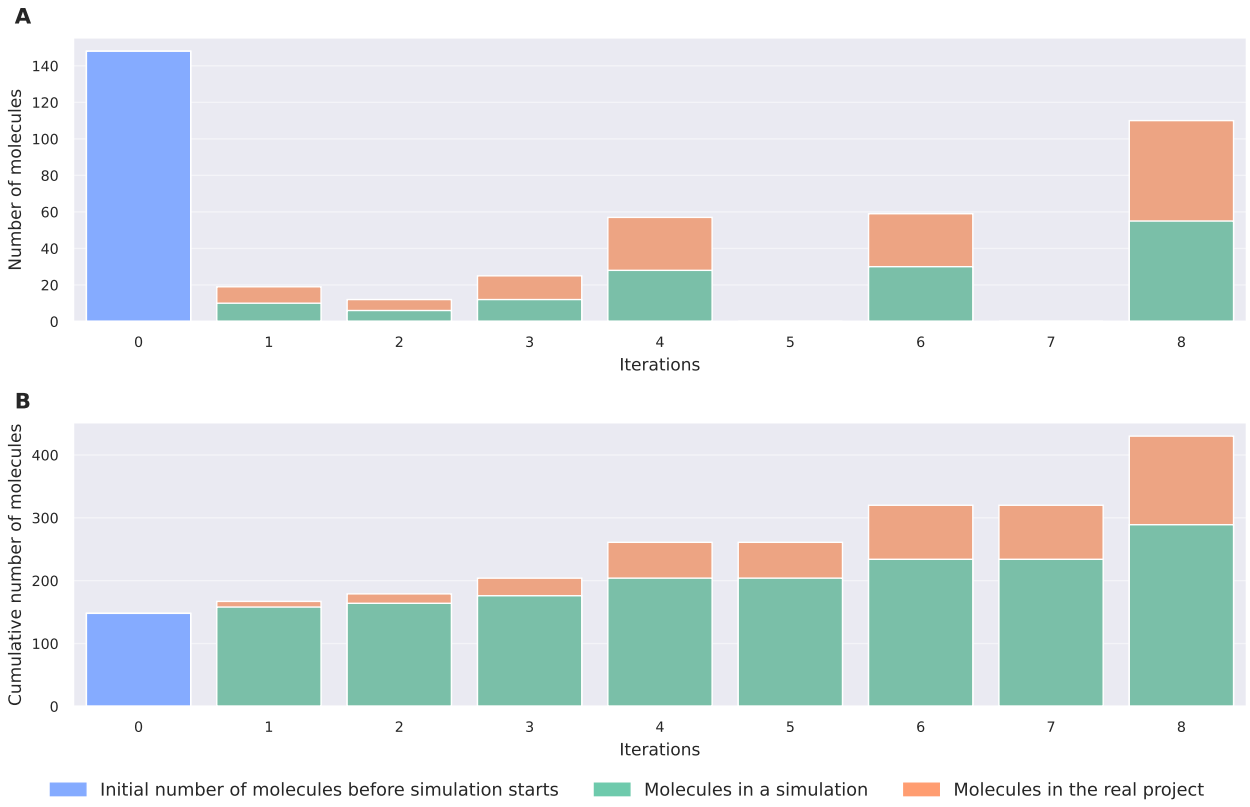

Fig. 25 Comparison of compound selection between the real project and a simulation (MMP-8 dataset). (A) Number of compounds selected per iteration. (B) Cumulative number of compounds selected over time.

6.3 Initial compounds and chemical series KDE TMAP

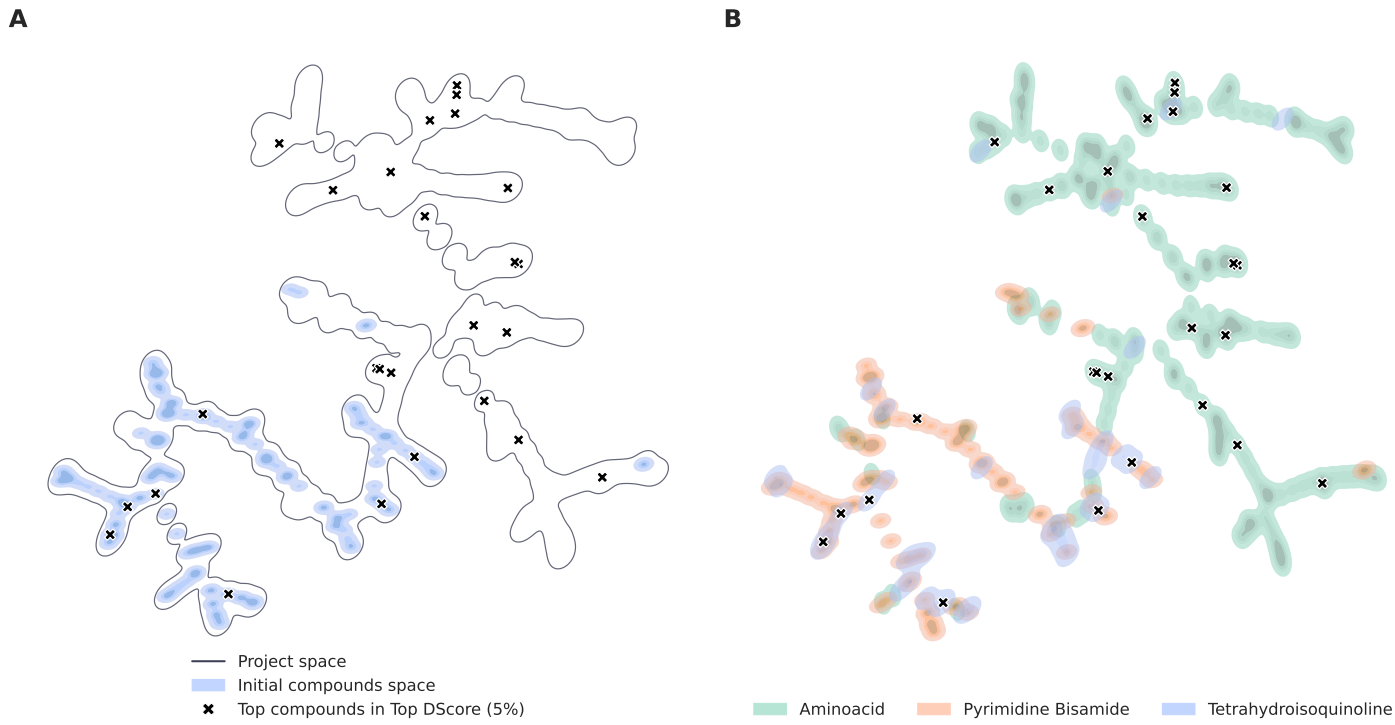

Fig. 26 Overview of the MMP-8 dataset chemical space. (A) Global view of the chemical space at the start of the simulation. (B) Distributions of the major chemical series. (C) Representative chemical structures of top-tier compounds (top 5% DScore) from various series. The box colors in (C) correspond to the chemical series colors used in (B).

6.4 Final overview KDE TMAP

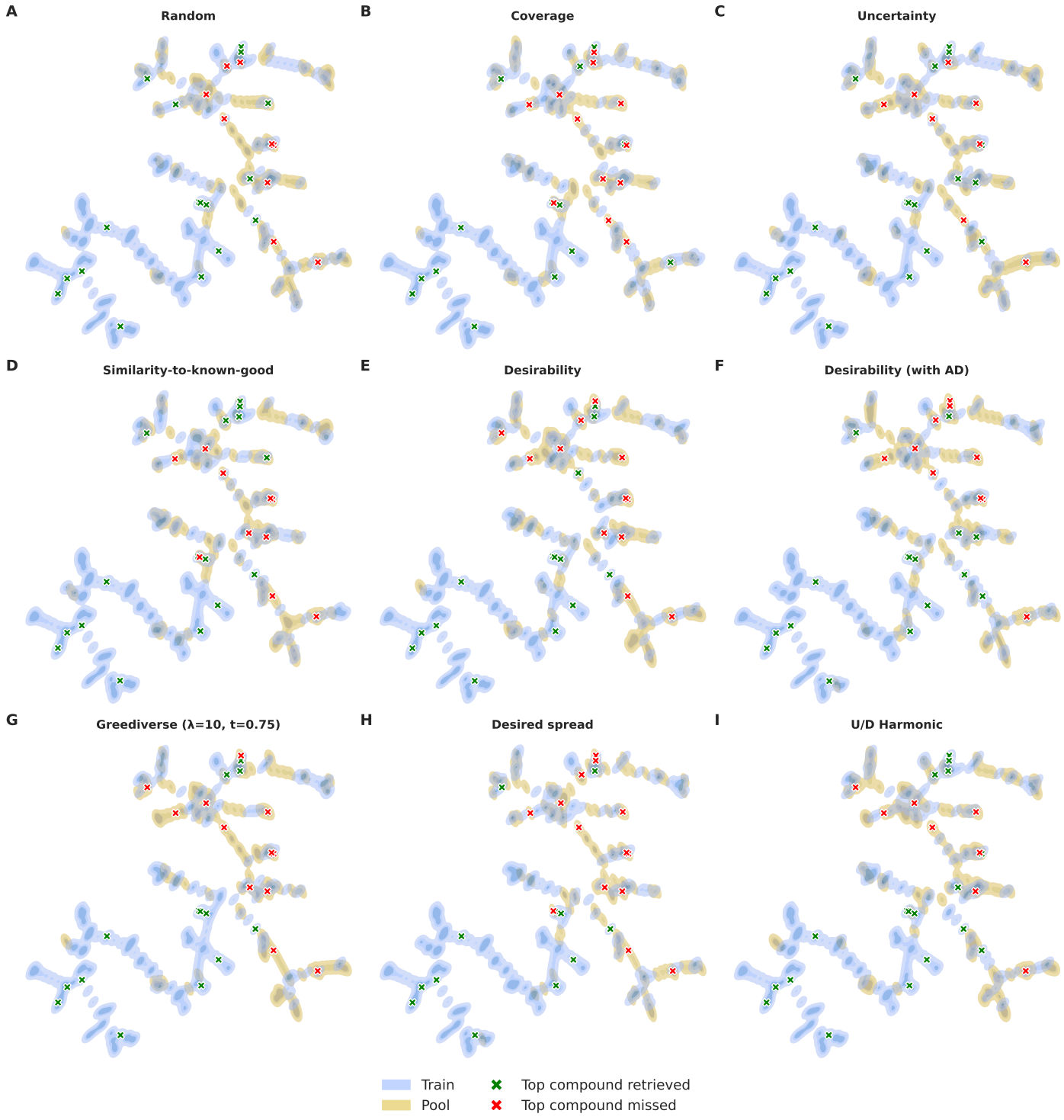

Fig. 27 TMAP visualizations of the final chemical space across nine acquisition functions (MMP-8 dataset). Regions: Blue areas represent the sampled subspace (selected molecules), while yellow areas indicate the remaining unselected pool. Markers: Crosses identify top-tier compounds (top 5% DScore). Green crosses denote top-tier molecules that were successfully retrieved, whereas red crosses denote those that were missed.

6.5 Exploration

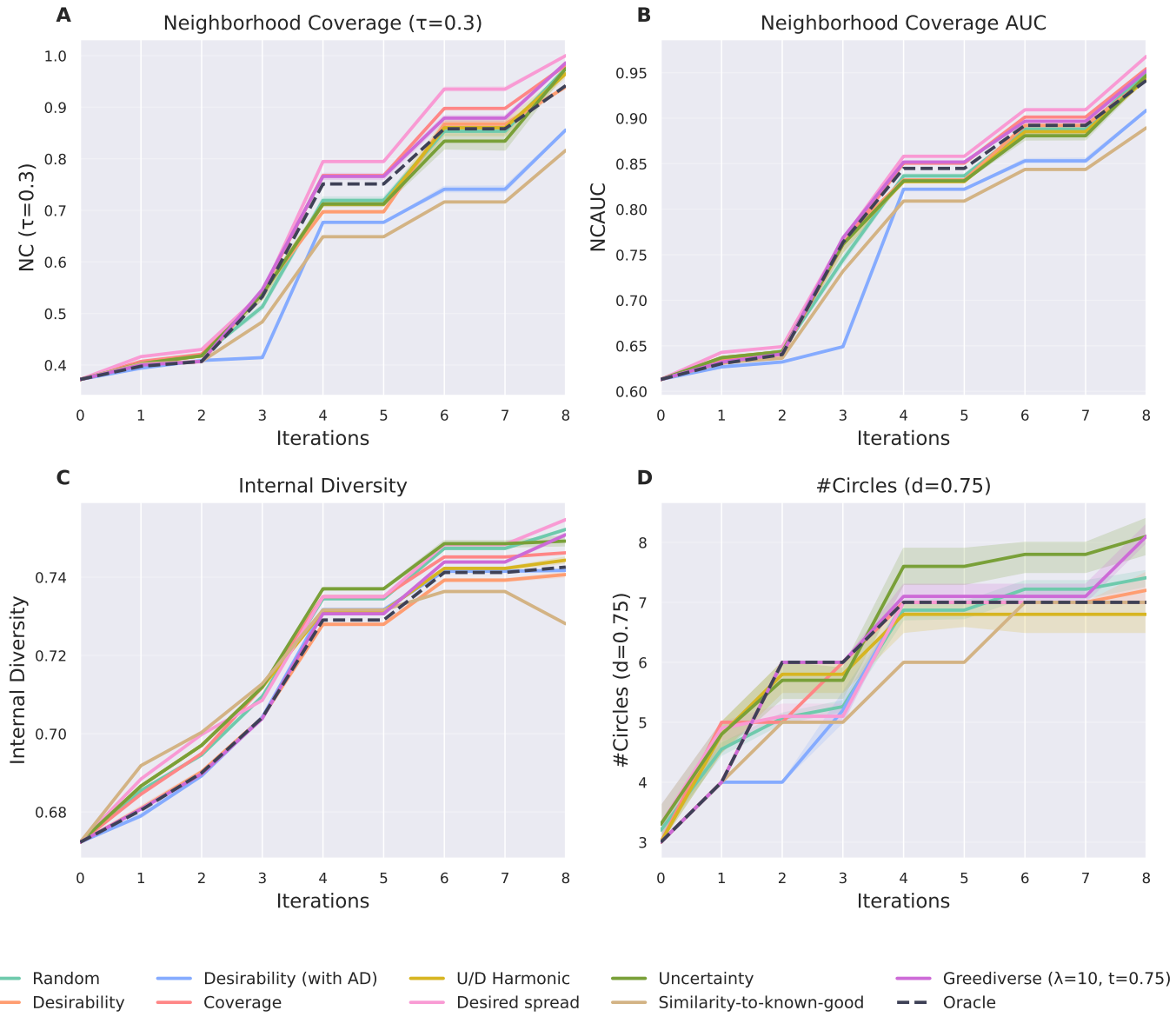

Fig. 28 Exploration metrics over iterations for selected acquisition functions (MMP-8 dataset). (A) Evolution of Neighborhood Coverage ($\tau = 0.3$). (B) Neighborhood Coverage AUC. (C) Internal Diversity. (D) Number of Circles (#Circles). Shaded regions indicate the 95% confidence interval. These intervals are calculated from 100 replicates for the Random strategy and 10 replicates for all other strategies.

6.6 Exploitation

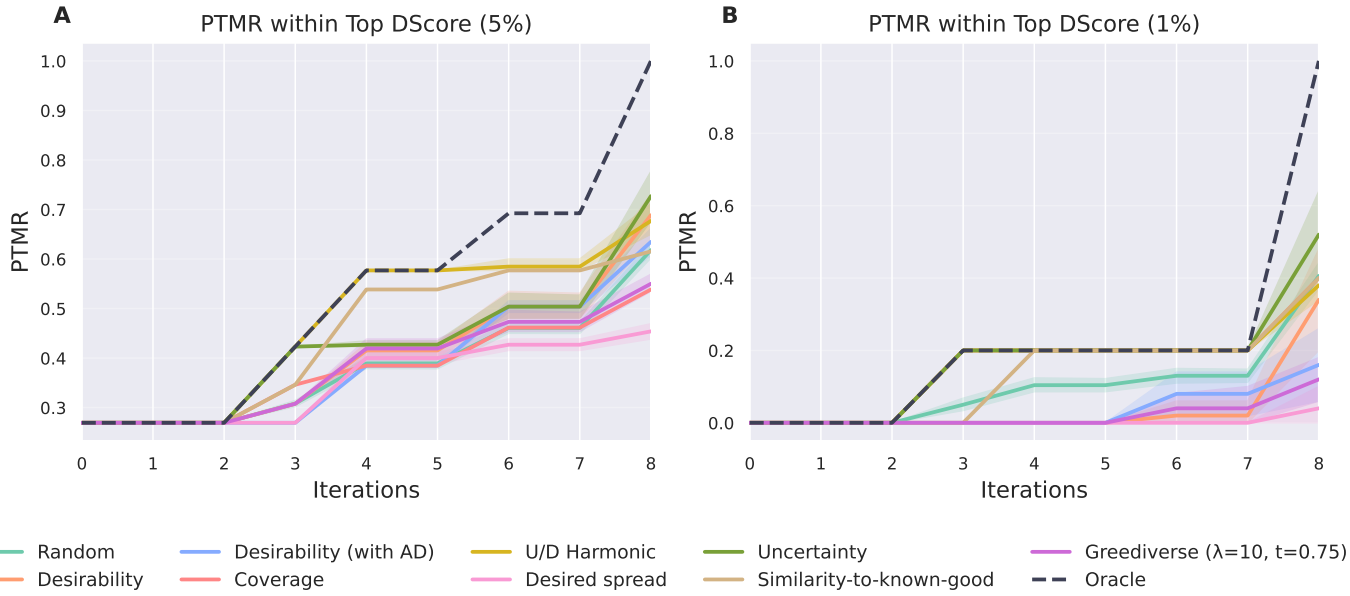

Fig. 29 Exploitation performance of various acquisition functions (MMP-8 dataset). The plots track the retrieval of top-tier compounds defined by two stringency levels: (A) the top 5% DScore and (B) the top 1% DScore categories. Shaded regions indicate the 95% confidence interval. These intervals are calculated from 100 replicates for the Random strategy and 10 replicates for all other strategies.

6.7 Exploitation / Exploration trade-off

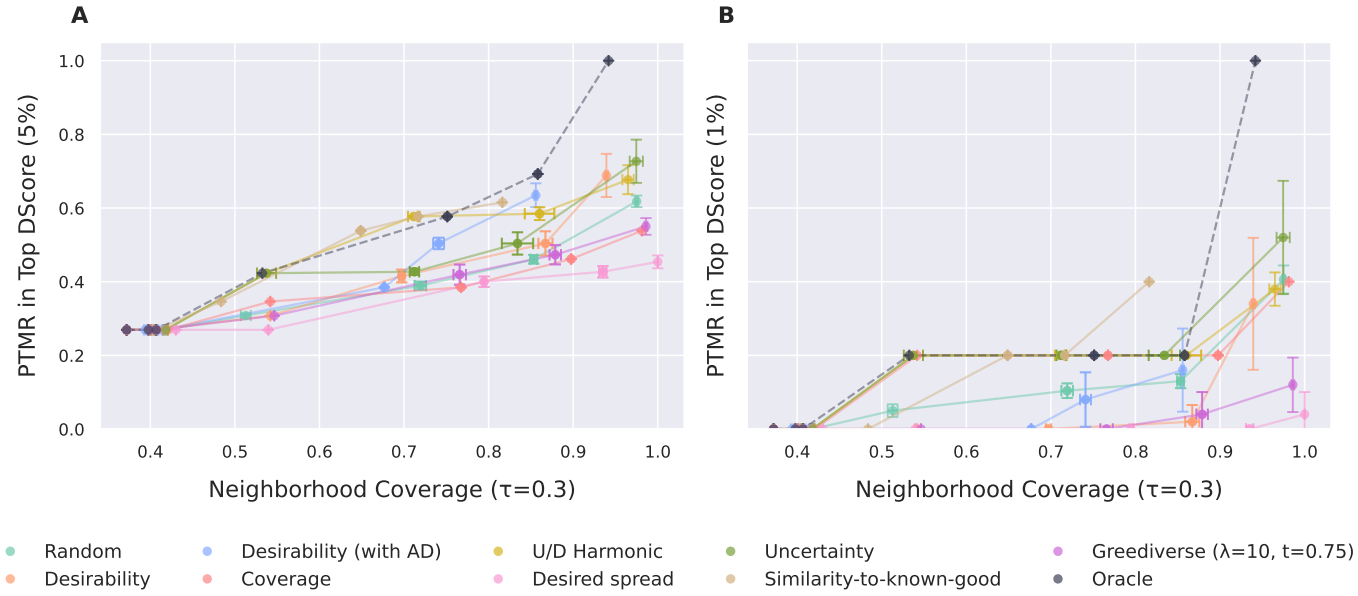

Fig. 30 Trajectories of Exploitation vs. Exploration (MMP-8 dataset). Performance is mapped by plotting Exploitation (monitored via PTMR within the top 5% DScore and 1% DScore) against Exploration (monitored via Neighborhood Coverage at $\tau = 0.3$). Shaded regions indicate the 95% confidence interval, calculated from 100 replicates for the Random baseline and 10 replicates for all other acquisition strategies.

6.8 Impact of the memory effect

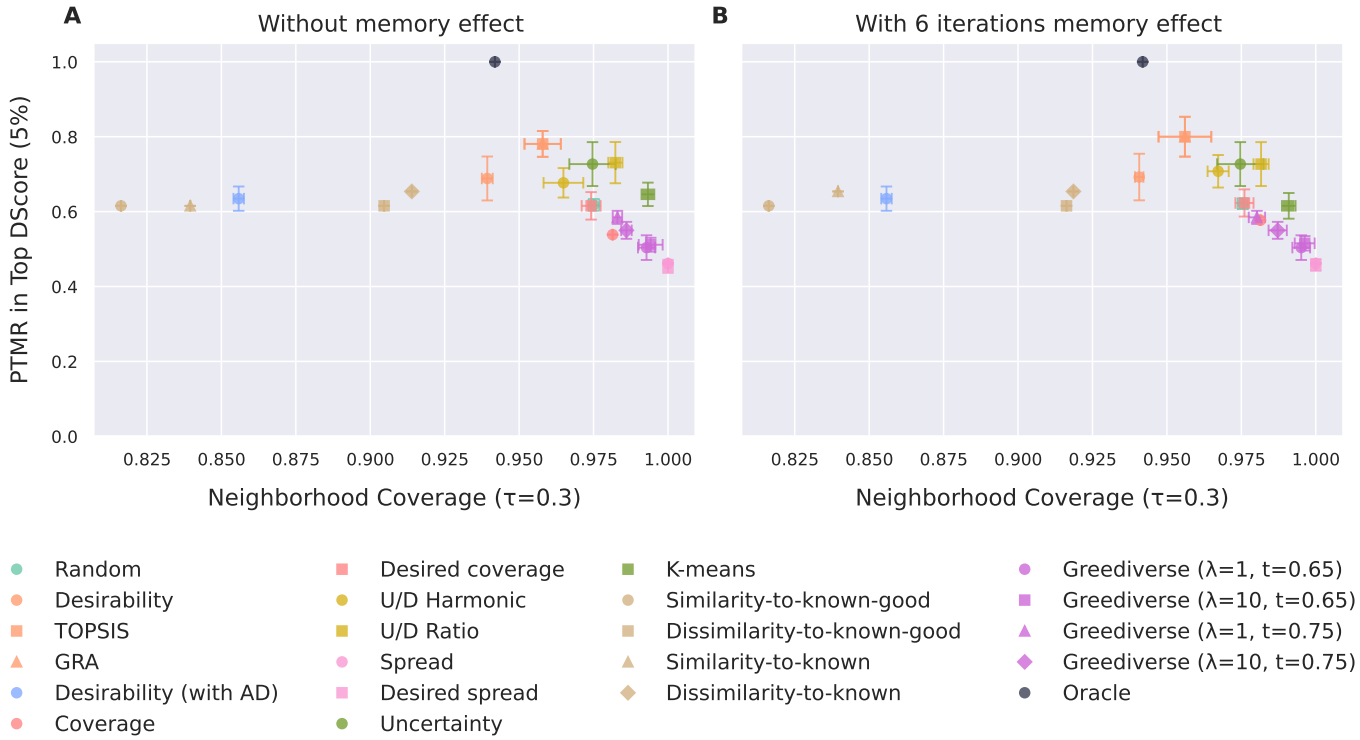

Fig. 31 Final overview of Exploitation vs. Exploration with and without memory effect (MMP-8 dataset). The panels display results without memory effect (A) and with a 6 iterations memory effect (B). The 95% confidence intervals are displayed, calculated from 100 replicates for the Random baseline and 10 replicates for all other acquisition strategies.

Notes and references

1 D. J. Woodward, A. R. Bradley and W. P. v. Hoorn, *Journal of Chemical Information and Modeling*, 2022.

2 D. J. Cummins and M. A. Bell, *Journal of Medicinal Chemistry*, 2016, **59**, 6999–7010.

3 R. P. Sheridan, B. P. Feuston, V. N. Maiorov and S. K. Kearsley, *Journal of Chemical Information and Computer Sciences*, 2004, **44**, 1912–1928.

4 D. Ju-Long, *Systems & Control Letters*, 1982, **1**, 288–294.

5 M. Langevin, M. Bianciotto and R. Vuilleumier, *Digital Discovery*, 2024.

6 J. MacQueen *et al.*, Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1967, pp. 281–297.

7 L. Kaufman and P. J. Rousseeuw, *Wiley Series in Probability and Statistics*, 2010, 68–125.

8 R. W. Kennard and L. A. Stone, *Technometrics*, 1969, **11**, 137–148.

9 C.-L. Hwang and K. Yoon, *Lecture Notes in Economics and Mathematical Systems*, 1981.

10 D. Rogers and M. Hahn, *Journal of chemical information and modeling*, 2010, **50**, 742–754.

11 J. Adamczyk and P. Ludynia, *SoftwareX*, 2024, **28**, 101944.

12 K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, *IEEE Transactions on Evolutionary Computation*, 2002, **6**, 182.

13 J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl and V. Svetnik, *Journal of Chemical Information and Modeling*, 2015, **55**, 263–274.

14 Z. Wu, M. Zhu, Y. Kang, E. L.-H. Leung, T. Lei, C. Shen, D. Jiang, Z. Wang, D. Cao and T. Hou, *Briefings in Bioinformatics*, 2020, **22**, bbaa321.

15 D. v. Tilborg, H. Brinkmann, E. Criscuolo, L. Rossen, R. Özçelik and F. Grisoni, 2024.

16 N. C. Frey, R. Soklaski, S. Axelrod, S. Samsi, R. Gómez-Bombarelli, C. W. Coley and V. Gadepally, *Nature Machine Intelligence*, 2023, **5**, 1297–1305.

17 R.-R. Griffiths, L. Klarner, H. Moss, A. Ravuri, S. Truong, Y. Du, S. Stanton, G. Tom, B. Rankovic, A. Jamasb *et al.*, *Advances in Neural Information Processing Systems*, 2024, **36**, year.

18 R. E. Higgs, K. G. Bemis, I. A. Watson and J. H. Wikel, *Journal of Chemical Information and Computer Sciences*, 1997, **37**, 861–870.