SynthonBench: Benchmarking Sample-Efficient Optimization in Combinatorial Chemical Spaces

Miroslav Lžičař [1]

Abstract

Synthesis-on-demand libraries define chemical spaces by reaction templates and compatible synthons. We introduce SynthonBench, a benchmark for sample-efficient search and optimization over combinatorial (synthon/make-on-demand/fragment-space) libraries: methods query reaction/synthon tuples to assemble valid products maximizing oracle reward—docking, an ML surrogate, or property objectives—under a shared budget, and are graded on exact metrics including top-k recall, regret, and the docking-call reduction factor (DCRF), which answers the practical question of how many fewer oracle calls a method needs to recover the same top-k yield as brute-force virtual screening.

The benchmark is complete: every oracle value is precomputed once over the full product universe and served from fixed lookup tables, so scoring a new method is exact, cheap, requiring no docking software and ensuring reproducibility. The release is built from a subset of the Freedom Space 5.0 make-on-demand library and includes three 1M Glide Docking HTVS tables (KIF11, PYRD, TGFR1), matched docking-surrogate tasks at 1M, 10M, and 100M products, and three non-docking suites (Synthetic Properties, Lead Rediscovery, and Lead Optimization).

At the default 10,000-call docking budget and across the synthetic objectives, genetic algorithms lead—our typed GA reaches mean top-1000 DCRF 21.5 (vs. 0.86 for random and a 99.06 exact-oracle ceiling), with SpaceGA close behind. When calls are scarce ($\leq$ 1,000), factored bandits take over, and SALSA-inspired TS tops the 1,000-call leaderboard at 39.6×. This GA advantage is not docking-specific: across the non-docking suites (lead rediscovery, lead optimization, druglike MPO, and synthetic-property landscapes) the typed GA again leads—the unique top significance group, closing 76% of the random-to-optimum utility gap—whereas factored bandits post competitive DCRF but fail to close it (negative on average), recovering locally enriched rather than globally optimal sets. Active learning (HASTEN) protocol posts the highest reduction factors of any method—several-fold above every synthon-space search at the larger scales—but only by enumerating the entire library first. Efficiency grows monotonically with library size, with SpaceHASTEN the strongest scaler among iterative search methods on the surrogate suites (top-0.1% DCRF 17 $\rightarrow$ 109 $\rightarrow$ 163 from 1M to 100M), and method rankings transfer strongly across oracles and scales (Spearman ρ 0.86–0.95). Hundred-to thousand-fold cost reductions often reported in the industry by proprietary active-learning pipelines on ultra-large libraries are therefore not a pure hype and scaling laws emerging from our audit make them plausible—but they have lacked a common, exact benchmark to verify them, which is precisely the gap SynthonBench fills.

1. Introduction

Ultra-large synthesis-on-demand libraries are now standard inputs to early hit discovery. Their size is largely combinatorial: purchasable products are assembled from validated reaction schemes and in-stock building blocks, and are increasingly accessed as synthon-based spaces rather than as explicitly enumerated molecule lists (Grygorenko et al., 2020; Lyu et al., 2019; Gorgulla et al., 2020; Warr et al., 2022; Sadybekov et al., 2022; Kalliokoski et al., 2025). Because larger libraries and broader testing empirically yield more hits and more diverse chemotypes (Liu et al., 2025), searching these spaces efficiently under a fixed evaluation budget is directly valuable. This representation also changes the algorithmic problem: screening is no longer the ranking of a fixed file of SMILES, but a decision about which reaction/synthon tuples to score and how to spend a limited

[1]Deep MedChem. Correspondence to: Miroslav Lžičař <miroslav.lzicar@deepmedchem.com>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

Table 1. Protocol-family summary for SynthonBench. All rows share the synthon-space interface and a 10,000-call submitted-oracle budget per task; *Size* is the product-universe tier (exact deduplicated counts are given in the dataset text). The non-docking suites split by objective into Synthetic Properties (additive, pairwise, sparse3, and constrained analytic landscapes over synthon properties), Lead Rediscovery (ECFP4 and FCFP4 aripiprazole similarity, a thresholded similarity variant, and a Murcko scaffold-hop), and Lead Optimization (a strict drug-likeness MPO and a Lead+MPO composite that couples similarity with physicochemical descriptors). The MPO rows are RDKit physicochemical objectives, not ADMET or learned property prediction. The *Seeds* column gives the submission-protocol minimum (3) with the released-baseline count (5) in parentheses, so compute-intensive methods are not gatekept; the seed policy is detailed in the Experimental Protocol.

Protocol	Type	Size	Tasks	Seeds
Glide HTVS	Docking	1M	3 targets	3 (5)
Docking Prediction	ML Surrogate	1M	3 targets	3 (5)
Docking Prediction	ML Surrogate	10M	3 targets	3 (5)
Docking Prediction	ML Surrogate	100M	3 targets	3 (5)
Synthetic Properties	Analytic	1M	4 tasks	3 (5)
Lead Rediscovery	FP Similarity	1M	4 tasks	3 (5)
Lead Optimization	MPO	1M	2 tasks	3 (5)

budget across reactions and building-block positions.

Most de-novo molecular-generation benchmarks ask whether a model can produce valid, high-scoring molecules. That question is useful, but it is distinct from search in a synthesis-on-demand library, where candidates are constrained to molecules vendors actually stock and that are reachable in a few reliable reaction steps—a simpler grounding than the multi-step routes targeted by retrosynthesis benchmarks. SynthonBench[1] instead measures a *search* objective: which reactions and synthons to sample, under a fixed budget, to recover the best products (by docking score, a property objective, etc.). Its core design choice is a separation between *search* and *audit*. During search, a method interacts only with an API and one shared oracle budget; the API returns valid synthon ids per reaction slot and scores proposed tuples on demand. During audit, the queried product identifiers are joined to precomputed reference score tables to give exact metrics—top-1000 recall, DCRF, top-0.1% DCRF, regret, and others—without requiring any optimizer to enumerate or store the full product space. To our knowledge, SynthonBench is the first systematic, reproducible benchmark to isolate the acquisition decision for synthon-based (fragment-space) combinatorial libraries with exact, license-free, and fast auditing.

The specific oracles in SynthonBench (Table 1) are illustrative, not essential. Each task is exposed to the optimizer as a black box returning a scalar utility, so the benchmark measures an acquisition policy's ability to recover elite products under a fixed budget, independently of what the oracle computes. This abstraction holds empirically: method rankings transfer strongly across the synthetic, surrogate, and measured Glide Docking oracles and across the 1M, 10M, and 100M scales (cross-run Spearman ρ from 0.86 to 0.95; Figure 3). A policy that does well here is therefore a strong candidate for any budgeted screen over the same synthon-space interface—whether the oracle is a cheap computed property, a measured docking table, a wet-lab assay in a lab-in-the-loop campaign (Coley et al., 2020; Frey et al., 2025), or a costlier structure- or physics-based method such as AlphaFold3, Boltz, or free-energy perturbation (Jumper et al., 2021; Abramson et al., 2024; Passaro et al., 2025; Hollingsworth & Dror, 2018; Kohn & Sham, 1965; Wang et al., 2015). SynthonBench is the cheap, exact stand-in for comparing such policies before they are run against those costlier oracles.

SynthonBench complements rather than replaces molecular-generation benchmarks such as GuacaMol (Brown et al., 2019), MOSES (Polykovskiy et al., 2020), PMO (Gao et al., 2022a), and Therapeutics Data Commons (Huang et al., 2021), and docking-oriented suites such as DOCKSTRING (Garcia-Ortegon et al., 2022). Those resources standardize distribution learning, de novo design, molecular optimization, property prediction, and docking-score objectives; SynthonBench asks instead which search policy most efficiently recovers the top-ranked products of a fixed reaction-defined space under a fixed budget.

Prior virtual-screening systems search non-enumerated synthon spaces, but each measures something other than budgeted acquisition. REAL-style make-on-demand libraries made billion-compound prospective docking practical, and VirtualFlow and V-SYNTHES demonstrated ultra-large and synthon-based docking (Grygorenko et al., 2020; Lyu et al., 2019; Stein et al., 2020; Kaplan et al., 2022; Gorgulla et al., 2020; Sadybekov et al., 2022). Chemical-space docking, SpaceLight/FTrees similarity search, SASS shape-aware search, and SpaceProp/SpaceProp2 property distributions all operate directly on combinatorial libraries (Beroza et al., 2022; Bellmann et al., 2021; Rarey & Stahl, 2001; Cheng & Beroza, 2024; Bellmann et al., 2022; Lübbers et al., 2024). ML-accelerated docking (Deep Docking, HASTEN, large-scale ML-guided docking) scores a subset, fits a surrogate, and triages an enumerated pool (Gentile et al., 2020; Kalliokoski, 2021; Sivula et al., 2023; Yang et al., 2021; Luttens et al., 2025), and SpaceHASTEN extends this to non-enumerated libraries via similarity expansion and learned docking-score prediction (Kalliokoski et al., 2025). Thompson sampling and SALSA target budgeted acquisition in synthon-factored spaces (Klarich et al., 2024; Grigg et al., 2025), and ROCS X materials describe reaction-aware syn-

[1]Code: `https://github.com/mireklzicar/synthonbench`; project page: `https://synthonbench.mireklzicar.com/`.

thon libraries, 3D similarity scoring, and Bayesian-bandit active learning at trillion scale (Muchmore et al., 2008; OpenEye, Cadence Molecular Sciences, 2026; Geballe & Lee, 2026; Cadence Design Systems, 2025; OpenEye, Cadence Molecular Sciences, 2025). SynthonBench is the audit layer beneath all of these: every method receives the same synthon-space API, oracle budget, duplicate accounting, and exact product-id evaluation.

Synthesis-aware generators—SynNet, Reaction-GFlowNet, SynFormer, SyntheMol, RosettaAMRLD, SpaceGA, and REvoLd—optimize *what to make* via reaction templates, synthetic trees, reaction-space policies, or evolutionary docking over make-on-demand libraries (Gao et al., 2022b; Koziarski et al., 2024; Gao et al., 2025; Swanson et al., 2024; Tang et al., 2025; Moesgaard & Kongsted, 2024; Eisenhuth et al., 2025). SynthonBench instead fixes the chemistry and budget and measures sample-efficient recovery of elite products.

In ML terms the setting is fixed-budget pure exploration and active search rather than cumulative-reward bandit learning: after paying for B oracle calls, a run is judged by how many elite products it found (Garnett et al., 2012; Bubeck et al., 2009; Chen et al., 2014). This is why SynthonBench reports DCRF in addition to best score.

2. Benchmark Design

Formal benchmark statement. Each scalar task is a fixed-budget, deterministic black-box optimization over a reaction-defined product set. The primary (*global*) protocol is *multi-reaction*: a task exposes reaction templates $\mathcal{R}$ and spends one shared budget across all of them, over the searchable universe

$$\mathcal{P}_{\mathcal{R}} = \bigcup_{r \in \mathcal{R}} \{\mathrm{Product}_r(s_1, \ldots, s_m) : s_i \in S_{r,i}\},$$

where $S_{r,i}$ is the typed synthon set for slot i. An optimizer proposes valid tuples $(r, s_1, \ldots, s_m)$, receives an oracle value per call, and stops after a shared budget B of charged calls. Every reported result uses this setting, so below we name protocols only by their oracle (Glide Docking, docking-surrogate, or the non-docking suites). The 1M, 10M, and 100M docking-surrogate protocols use $B = 10{,}000$ with matched exact audits to probe scaling and systems behavior; they complement, but do not replace, the Glide Docking endpoint.

Each run records the queried product identifiers, oracle outcomes, best-so-far curves, and the final selected set. Stable product identifiers are central: post-hoc metrics join queried ids to reference score tables rather than relying on canonical-SMILES matching. Methods are scored as pure-exploration policies by simple regret and elite-set recovery, not cumulative reward (Bubeck et al., 2009)—matching virtual screening, where the query cost is paid during search and the benchmark asks how much of the best reference set survived the budget.

Shared-budget tasks. Realistic campaigns allocate one finite budget across many reactions, so the shared-budget task is the main SynthonBench unit: the 1M protocol uses all 42 canonical reactions and one 10,000-call budget per target. It inherits the stable product ids and failure penalties of the single-reaction Glide Docking track but shifts the question from per-reaction proposer quality to shared-budget reaction allocation.

Task families. The primary 1M tier comprises 990,610 unique products (after product-id deduplication) enumerated from the synthons and reaction templates of the Freedom Space make-on-demand library. The Glide Docking track uses fixed Glide HTVS tables for KIF11, PYRD, and TGFR1—actual Glide results consumed as lookup feedback, not ML predictions; since lower raw docking score is better, utilities are negated scores. The docking-surrogate suite instead uses LightGBM surrogates for the same three targets, trained on 0.85–1.00M docked rows per target (0.60–0.70M in the 70% training split), generated with Phase/LigPrep and Glide from Schrödinger Suite 2023-4 following SpaceHASTEN (Kalliokoski et al., 2025; Friesner et al., 2004). Held-out Spearman correlation with the source Glide labels is 0.71–0.79 and lowest-0.1% tail recall is 0.153–0.279 (Appendix Table 8), so we treat these as fixed surrogate-landscape search tasks rather than measured docking. The 70/15/15 split is random over product rows: rows are disjoint, but scaffolds, synthons, and reaction templates can recur across the split, so the validation measures interpolation within the source distribution, not transfer to new chemistry or measured binding.

The non-docking suites add objectives over the same 1M universe in three families (Table 1): *Synthetic Properties* (analytic landscapes over synthon properties), *Lead Rediscovery* (ECFP4/FCFP4 similarity and a Murcko scaffold-hop (Bemis & Murcko, 1996)), and *Lead Optimization* (a strict drug-likeness MPO and a Lead+MPO composite). Exact objective definitions live in the release metadata; the full protocol and results are in Appendix Section A.13. The 1M, 10M, and 100M surrogate tiers stay in the benchmark to probe scaling, stability, and exact-audit feasibility, and are reported as scale summaries rather than biological evidence. Table 1 summarizes the protocol families.

Exact metrics. For a score-table task, the reference for objective column c is the full audited universe $\mathcal{P}_{\mathcal{R}}$ sorted by utility; for a queried product-id set Q, top-k recall is $|Q \cap \mathrm{Top}_k(c)|/|\mathrm{Top}_k(c)|$. Our primary efficiency metric is

the oracle-call reduction factor

$$\mathrm{DCRF}_k = \frac{top - krecall}{B/N},$$

the enrichment of the charged query set over uniform sampling (universe size N, budget B). It has random expectation 1 and reads operationally as the factor by which uniform sampling would need more calls to match the same top-k yield. The acronym is from the docking case; the identical quantity, stored as `oracle_call_reduction_factor`, is used for every oracle, so we write DCRF throughout, including the non-docking tasks. We also report the top-0.1% threshold-hit-rate DCRF, exact regret $\max(0, u^\star - \max_{p \in Q} u(p))$, and the rank of the best queried product. Against the exact-oracle ceiling $\mathrm{DCRF}_k^\star$ (the best DCRF achievable at that budget), the 1M benchmark ($N =$ 990,610, $B =$ 10,000, $k =$ 1000) gives $\mathrm{DCRF}_{1000}^\star =$ 99.1×, so a top-1000 DCRF of 20× is both a 20-fold random-equivalent saving and ≈20% of the available exact-oracle efficiency. Appendix Section A.4 derives the random expectation, the random-equivalent-oracle-call (REOC) and enrichment-factor equivalences, and the ceiling formula.

3. Methods

We evaluate a compact set of baselines that all interact through the synthon-space API—proposing reaction/synthon tuples under the shared budget rather than emitting arbitrary SMILES—with random search as the lower bound. Methods marked *(Ours)* are search policies we implemented for synthon spaces, where to our knowledge no prior published system applies the same acquisition idea to a non-enumerated synthon library; we cite related work for lineage, but the adaptation, implementation, and runs are ours. The remaining rows are clearly labeled analogues or reimplementations of named prior systems (Thompson sampling, SALSA, ROCS X, SpaceHASTEN), used where the original is closed-source, depends on a proprietary engine, or assumes full enumeration. Appendix Section A.8 records which rows are in-house analogues and which are exact upstream comparisons, and Appendix Table 7 lists every per-method hyperparameter. The same names are used throughout, and every search method is rerun on every protocol (Glide Docking and the 1M/10M/100M surrogate scales).

Genetic Algorithm (GA, Ours). A reaction-aware typed GA over valid synthon tuples: population 96 under elitist (top-half) selection, crossover between sampled elite parents, per-slot synthon mutation (0.35), reaction re-typing (0.15), 10% random immigrants, and unseen-child retries. Its closest precedent is the reagent-index GA in the Klarich et al. (2024) release, which likewise recombines per-component reagent indices on a synthesis-on-demand library; related search appears in SpaceGA (Moesgaard & Kongsted, 2024) and graph-based molecular GAs (Jensen, 2019). Our variant adds reaction identity as a typed gene for native global search across reactions; none of these is reproduced here.

Iterative Surrogate Expansion (ISE, Ours). An iterative surrogate search in the spirit of the SpaceHASTEN screening loop (Kalliokoski et al., 2025) but without its proprietary backends: from 32 random queries it periodically refits an ExtraTrees regressor (Geurts et al., 2006) on product ECFP4 fingerprints (Rogers & Hahn, 2010; Landrum, 2025), mutates elite queried and predicted tuples to expand local synthon neighborhoods, and greedily queries the best-predicted candidates. The SpaceHASTEN name is reserved for the licensed FTrees/SpaceLight rows (SH, SHC); ISE is the in-house analogue.

Random Forest (RF, Ours). A pooled random-forest surrogate (16 trees) over Blake2b-hashed reaction- and synthon-indicator features (Breiman, 2001); it proposes the best tuple from a sampled pool and refits every 1,000 charged calls after a 128-query warmup.

Monte Carlo Tree Search (MCTS, Ours). UCT over the synthon-slot assignments (exploration constant 1.4) with progressive widening and short stochastic rollouts (Browne et al., 2012).

Hill Climb (HC, Ours). A GuacaMol-style hill climber (Brown et al., 2019) restricted to synthon-slot mutations so every proposal stays inside the reaction space: a rank-weighted elite buffer (size 16), mutation of a sampled elite parent, and restart from a random tuple after 64 non-improving proposals.

Thompson Sampling (TS). A reagent-prior Thompson sampler reimplementing the warmup-prior pattern of Klarich et al. (Klarich et al., 2024): after a per-synthon warmup (three trials each) it estimates a shared prior and updates per-synthon Gaussian posteriors with a normal–normal conjugate rule.

SALSA-inspired tabular TS (SAL). A tabular synthon-acquisition Thompson loop inspired by SALSA (Grigg et al., 2025): empirical per-synthon mean/variance posteriors, assembling one Thompson-sampled synthon per slot. It is an analysis-oriented reimplementation, not the MPNN SALSA system.

ROCS-X-inspired Bandit (BB). A ROCS X-inspired Bayesian belief-fusion bandit—a public-description analogue, not a reproduction. Public OpenEye/Cadence materials describe ROCS X as using reaction-aware synthon

search, top-hit feedback, and Bayesian bandits, but the implementation is closed-source (OpenEye, Cadence Molecular Sciences, 2025). Our bandit maps observed utility to an empirical similarity percentile, applies the Muchmore et al. ROCS probability-assignment curve ($F_{\max} = 0.46$, $\mathrm{SC}_{50} = 0.86$, slope 6.8), updates Beta posteriors over synthons and observed pairs, and ranks candidate tuples by Hooper/Dempster–Shafer joint belief $1 - \prod_i (1 - b_i)$ (Muchmore et al., 2008; OpenEye, Cadence Molecular Sciences, 2026; Geballe & Lee, 2026; Cadence Design Systems, 2025). We never used the ROCS X product or code, so this row is not ROCS X, ROCS, OMEGA, or FastROCS.

Random anchors (RPW, RRW). Two uniform-random baselines differing only in how the shared budget splits across reactions: product-weighted (RPW), proportional to per-reaction product-space size, and reaction-weighted (RRW), uniform across reactions. Together they isolate reaction allocation from the within-reaction proposer.

SpaceHASTEN XGB (SH). The SpaceHASTEN FTrees/SpaceLight backend run directly on the synthon space (Kalliokoski et al., 2025), with a cycle-local XGBoost surrogate (Chen & Guestrin, 2016) substituted for the upstream Chemprop model purely for speed and parity with the other surrogate baselines. Since SpaceHASTEN's default backend is FTrees/SpaceLight, we name this variant simply *SpaceHASTEN XGB*; the surrogate-free variant (SHC) is not a standard configuration and appears only as an appendix ablation (Appendix A.10). Being non-enumerated, SH is compared directly on the Glide Docking endpoint (Table 4); software versions and the CoLibri space construction are in Appendix Section A.10. These are the only rows that may carry the SpaceHASTEN name.

SpaceGA (SGA). The official SpaceGA repository (Moesgaard & Kongsted, 2024) run directly on the same CoLibri synthon space: a graph genetic algorithm whose mutation operator is a SpaceLight similarity search, so every offspring is an analog that already exists in the library. We run it as published (default `ECFP4` fingerprints, crossover rate 0.2, population 100) under the shared 5×2000 charged-call budget, scoring proposals against the same score-table oracle as every other method. Two minimal, behaviour-preserving adapter changes were required (Appendix A.8): the upstream tool exposes no random seed, and its SpaceLight-result parser is guarded against the empty/malformed outputs that otherwise abort a generation under parallel search. Like SH/SHC it uses the licensed FTrees/SpaceLight backend, but it is an external published genetic search rather than a SpaceHASTEN configuration.

From per-reaction proposers to multi-reaction search. A per-reaction proposer optimizes within one reaction; the shared-budget protocol spends one budget B across all 42. A reaction allocator chooses which reaction to query at each charged call and delegates the within-reaction proposal. GA and RF search all 42 directly and use no allocator. TS, SAL, and BB use an adaptive per-reaction Gaussian-Thompson allocator—reaction selection is integral to these bandits. ISE uses a softmax-over-mean-reward allocator (temperature 0.25, with a 5% uniform-exploration mixture), and MCTS and HC are routed by the same static product-weighted allocator as RPW. Definitions and parameters are in Appendix Table 7.

For external context we also run enumerate-then-learn HASTEN and MolPAL under the same 10,000-call, five-batch curriculum: an initial 2,000-compound batch, then XGB trained on observed fixed-table scores, predicting the full precomputed 1M fingerprint matrix and charging only the selected 2,000 fixed-table lookups per batch. This is fair in oracle budget but requires the full library to be enumerated first, so these rows are appendix baselines (Table 3) rather than among the main non-enumerated methods. Where full enumeration is feasible they can match or exceed the synthon-space methods, which helps delimit when reaction-native search is most useful.

All methods are wrapped with a unique-query guard: methods may occasionally propose the same tuple more than once, but duplicate oracle calls are rejected before scoring.

4. Experimental Protocol

The reported baselines use five seeds $0, \ldots, 4$ (15 matched runs per method across the three targets) to tighten confidence intervals; the submission protocol requires only the three-seed minimum (seeds $0, 1, 2$), and restricting the Glide Docking endpoint to it leaves the Genetic Algorithm the unique top-band method ($p < 0.05$), reshuffling only the statistically tied elite cluster (Table 2, groups b–d). Each method follows a frozen execution protocol with shared task definitions, score-table checksums, duplicate-query guards, and reproducible command records; diversity metrics for large scalar sweeps are computed offline, keeping the online oracle interaction identical across methods. The result package contains manifests, environment captures, exact metric tables, and plot data.

The primary Glide Docking track is fixed-table measured-score feedback under a medchem-aware utility: property and substructure filters are applied, each finite successful Glide row is converted to the lipophilicity-penalized score $-\mathrm{Glide} - \mathrm{cLogP}$, and Glide rejections, NaNs, and filter failures receive zero utility. Every optimizer thus sees the same filtered objective in-loop rather than post-hoc credit

under a pure docking-score oracle; earlier raw-score and external-baseline checks remain in the supplementary files but are not primary figures or leaderboards.

5. Results

Across three measured Glide targets and surrogate suites from 1M to 100M products, simple adaptive policies recover elite products an order of magnitude more efficiently than random sampling—but the best policy depends on the budget and the scale, exactly the sensitivity SynthonBench is built to measure.

Table 2 is the measured-score anchor; Appendix Figure 2 shows it target-resolved with bootstrap confidence intervals, and Appendix Table 4 gives the Glide validity, best-score, and regret diagnostics. The medchem filter leaves 12.8% of the 990,610 unique products, and after target-specific Glide failures the valid oracle universe is 10.5–12.7%. Among the non-enumerated synthon-space methods the reaction-typed Genetic Algorithm (GA) is strongest: mean top-1000 DCRF 21.51 over the 15-run matrix, versus 0.86 for the product-weighted random anchor (RPW) and a 99.06 exact-oracle ceiling—21.7% of the efficiency an omniscient selector could reach at this budget. ISE, SALSA, the ROCS-X-inspired bandit, reagent-prior Thompson sampling, and the SpaceHASTEN XGB backend (SH) follow; the external SpaceGA tool lands at 17.8, statistically tied with the ISE/SALSA cluster (group b) and itself significantly behind GA. The head ordering is statistically robust: GA occupies its own band (group a), significantly ahead of every other method under Holm-corrected paired t-tests blocked by target and seed (versus the next-ranked ISE, $p < 0.001$; Wilcoxon signed-rank concurs), while the methods below it form one indistinguishable cluster (groups b–d). Because SpaceHASTEN requires no full-library enumeration, its backend is run directly on the same matrix: over the merged CoLibri space (38/42 supported reactions) SH reaches 13.68 mean top-1000 DCRF, and an ablation shows the surrogate is essential—removing it drops the backend to 5.16, a 2.7× loss (Appendix A.10). The enumerate-then-learn HASTEN and MolPAL baselines (Table 3) require full 1M enumeration but, where feasible, can match or exceed the synthon-space methods.

A nuanced, budget-dependent efficiency landscape emerges (Figure 1), which sweeps nine budgets from 250 to 100,000 calls on the same medchem-filtered Glide Docking run (the largest is ≈1/10 of the 990,610-product library). When calls are scarce (≤1,000) the factored SALSA bandit leads on all three targets (30–52× at 500–1,000 calls), with the ROCS-X-inspired bandit and ISE close behind and the Genetic Algorithm only mid-pack. By 2,500–10,000 calls the ordering inverts and the Genetic Algorithm leads, before SALSA, the bandit, the random forest, and (on KIF11) ISE overtake it near 100,000 calls as the exact-oracle ceiling collapses from 991× to 9.9×. Both regimes are large and robust across seeds—SALSA's small-budget lead is several-fold, not marginal—and only the precise crossover budget is seed-sensitive. The mechanism is the classic low-data trade-off: a factored or surrogate model updates beliefs about many unseen candidates from each call, whereas the population-based Genetic Algorithm must first fill and mature its 96-member population before recombination pays off. The external SpaceGA tool—an independent graph-GA over the same space—traces the same signature under budget-matched curricula (mid-budget peak of 20.4× near 2,500 calls, weakest below 1,000), confirming this is a property of population search rather than our particular implementation. Nor is it a tuning artifact: retuned for a 1,000-call budget, the Genetic Algorithm improves from 18.0 to 27.0× but still trails SALSA's 39.6× (Appendix A.2). We therefore read the 10,000-call top-1000 DCRF as one operating point for the expensive-oracle regime this benchmark targets, not a budget-invariant ranking; the elite adaptive methods stay well above random throughout, while MCTS, hill climbing, and the reaction-weighted random anchor sit only modestly above the DCRF= 1 line. The enumerate-then-learn HASTEN reference, overlaid for context (dashed), tracks above the synthon-space methods through the 1,000–10,000-call band—peaking at 45.6× near 2,500 calls, above every synthon-space peak—reflecting its full-enumeration head start rather than a like-for-like search advantage.

The surrogate scales of Table 2 (lower three rows of Appendix Figure 2) place the 1M, 10M, and 100M docking-surrogate suites in the same target-resolved view, each audited at a scale-normalized top-0.1% elite set. At every scale the in-loop oracle is the same precomputed surrogate table the audit grades against (served at 10M/100M through a memory-mapped product-id index), so optimization and scoring share one landscape and the search methods separate cleanly from the random anchors in every panel. Their elite-set advantage grows with the universe: in the aggregate column the typed GA improves from 16.2 top-0.1% DCRF at 1M to 35.0 at 10M and 49.0 at 100M, and the ROCS-X-inspired bandit from 11.4 to 41.7, while product- and reaction-weighted random stay near their unit baselines; per target the strongest runs are higher still (GA 62.9 and the bandit 64.7 on KIF11 at 100M). The strongest scaler is the non-enumerated SpaceHASTEN similarity backend (SH; top-0.1% DCRF 17.1/109.1/163.3 across the three scales), second only to the enumerate-then-learn HASTEN reference, with the external SpaceGA tool tracking the typed GA (17.7/31.4/47.5). SH's lead is surrogate-specific, though: on the *measured* Glide endpoint it is only mid-pack (top-1000 DCRF 13.7), so part of the gain may reflect alignment between fingerprint-similarity search and a fingerprint-trained surrogate rather than recovery of measured binders. The

Table 2. Top-0.1% DCRF, the call-reduction factor for recovering the top-0.1% elite set (top-1000 at 1M, top-10,000 at 10M, top-100,000 at 100M; at the 990,610-product 1M scale the top-0.1% and top-1000 sets coincide). *Glide 1M @10k*/*@1k* report it on the measured Glide Docking run at the 10,000- and 1,000-call budgets (the two vertical guides in Figure 1); *Surr. 1M/10M/100M* report it at the 10,000-call budget on the three docking-surrogate scales; and the final three columns summarize the 250–10,000-call budget-response sweep on the same Glide Docking 1M run—its peak value (*Peak DCRF*), the budget at which the peak occurs (*Peak budget*), and its mean over the regime (*Avg. DCRF*). Rows are ordered by *Glide 1M @10k*; random expectation is DCRF $= 1$ (Appendix Figure 2). Surrogate-scale columns measure recovery of predicted docking scores, which correlate but are not entirely comparable to the measured-Glide scores. *Sig.* gives compact-letter significance groups on the ordering column (Holm-corrected paired tests over the 15 matched runs per method; methods sharing no letter differ at $p < 0.05$; Appendix A.2). ‡HASTEN requires full enumeration before search and is excluded from the significance ranking (dash). Method names follow Section 3.

			Glide 1M		**Surrogate**			**Budget-Response**		
Method	Family	Sig.	@10k	@1k	1M	10M	100M	Peak	Budget	Avg
HASTEN‡	Active learning	–	37.5	33.2	43.6	183.5	407.8	45.6	2,500	36.0
Genetic Algorithm	Evolutionary	**a**	**21.5**	18.0	16.2	35.0	49.0	29.1	2,500	20.0
SpaceGA	Evolutionary	b	17.2	14.6	**17.7**	31.4	47.5	20.4	2,500	13.8
Iterative Surrogate Expansion	Active learning	b	16.8	23.2	10.7	14.7	9.2	23.3	2,500	20.4
SALSA-inspired Thompson	Bandit	bc	16.6	**39.6**	12.1	30.4	24.4	**39.6**	1,000	**29.2**
ROCS-X-inspired Bandit	Bandit	bcd	14.4	25.4	11.4	33.2	41.7	29.6	2,500	19.4
Thompson Sampling	Bandit	c	13.9	24.8	10.5	23.6	24.7	24.8	1,000	17.2
SpaceHASTEN XGB	Similarity search	bcd	13.7	28.0	17.1	**109.1**	**163.3**	28.0	1,000	20.8
Random Forest	Active learning	d	11.7	4.8	11.5	14.8	6.1	11.7	10,000	6.8
Reaction-weighted random	Random	e	3.4	2.6	2.1	3.4	2.3	3.5	10,000	2.9
Monte Carlo Tree Search	Metaheuristic	f	2.3	1.5	3.3	3.0	3.1	2.3	10,000	1.5
Hill Climb	Metaheuristic	f	2.0	1.1	2.6	2.1	2.5	2.0	10,000	1.5
Product-weighted random	Random	g	0.9	0.7	1.0	1.1	1.1	0.9	10,000	0.7

strong surrogate-to-Glide rank correlation ($\rho = 0.84$, below) suggests the effect is largely genuine sample efficiency, but only a scale-up against measured docking would settle it. Elite-set DCRF rises monotonically with no plateau, so the much larger production libraries should support larger reductions still—an extrapolation SynthonBench certifies only to 100M (Appendix A.1). Surrogate-refit methods such as ISE and the pooled random forest instead peak at 10M and fall back at 100M, as a fixed 10,000-call refit budget cannot keep pace with a 100M universe, though both stay well above random.

Better surrogate performance predicts better Glide Docking performance: method rankings transfer strongly between every pair of the four runs (Spearman ρ from 0.86 to 0.95 over 11 shared methods), including from the 1M surrogate to the Glide endpoint ($\rho = 0.84$; Appendix Figure 3). This is a method-level correlation, not a product-level one: a TGFR1 parity audit finds essentially no rank agreement between the surrogate labels and the corrected Glide screen (finite-score Spearman 0.002). That sensitivity is itself a benchmark feature—SynthonBench exposes when search policies transfer across fixed oracles and when they do not—while the main Glide Docking result uses the filtered lipophilicity-penalized utility throughout.

The remaining breadth—slot-additive versus interaction-heavy synthetic objectives, exact two-objective selectivity frontiers, deterministic score/diversity portfolios, scaffold and cluster coverage, and Glide Docking molecule grids—is deliberately appendix material.

6. Discussion

SynthonBench is intentionally simple in method ambition and strict in evaluation. The point is not that GA or Thompson sampling are the best possible optimizers, but that cheap adaptive policies already recover far more elite products than random under a small budget—so any heavier learned method must justify its wall-clock cost against that bar. The benchmark's job is to make that bar exact and to expose where methods fail.

The Glide Docking checks sharpen this. Product-weighted random is a cost anchor, not the baseline a practitioner would choose once reaction allocation is learned; the Genetic Algorithm, the strongest current public baseline, still reaches only 21.7% of the exact top-1000 ceiling, its advantage is target-dependent, and surrogate parity checks show that surrogate gains do not always transfer— precisely the failure modes a benchmark should expose.

GA outperforming both in-house Thompson variants is best read as a protocol result, not a general verdict on Thompson

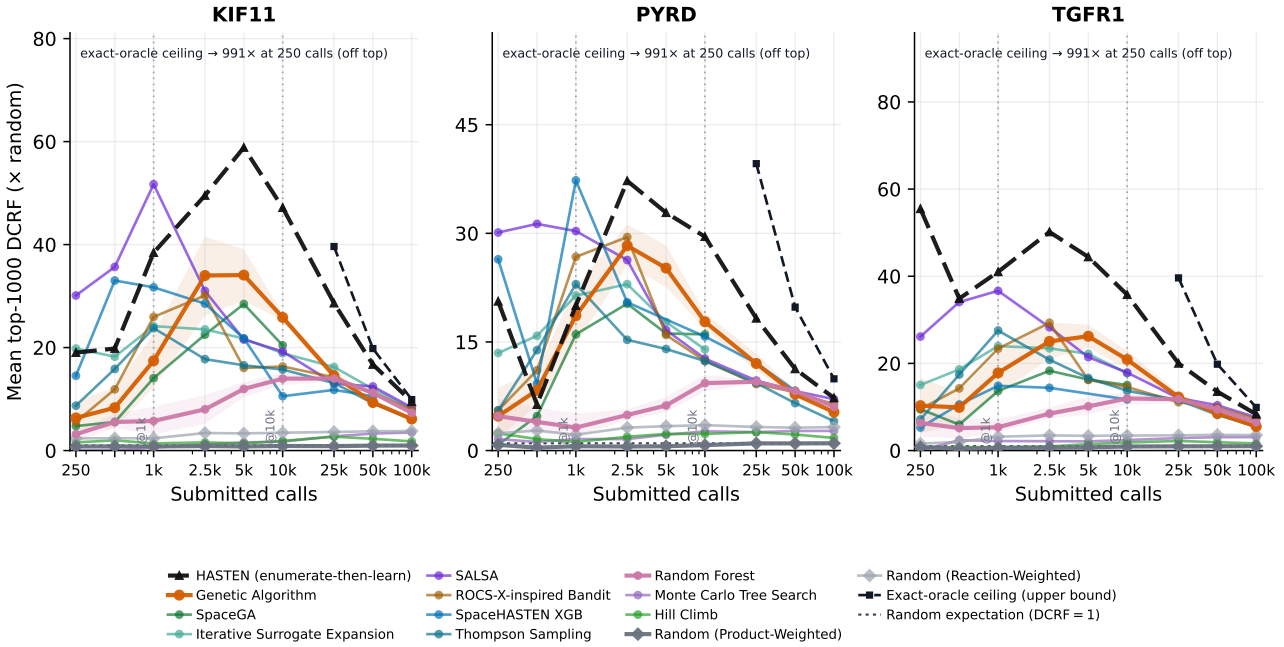


Figure 1. Budget-response for the medchem-filtered Glide Docking benchmark: top-1000 DCRF (top-1000 recall divided by B/N; random expectation 1, dotted line) at budgets from 250 to 100,000 calls on KIF11, PYRD, and TGFR1, for the ten in-house leaderboard methods and the non-enumerated SpaceHASTEN XGB backend (SH). The enumerate-then-learn HASTEN reference (dashed black) is overlaid for context: it requires full 1M enumeration before search, so it is not a synthon-space search method but indicates the achievable ceiling where enumeration is feasible. Curves are means and bands are 95% bootstrap confidence intervals over seeds within each target. Budget points at or below 10,000 are read from the primary leaderboard traces (the 10,000-call guide reproduces the Appendix Figure 2 leaderboard); larger budgets and the SH curve use the extended campaigns detailed in Appendix A.2. Method abbreviations follow Section 3.

sampling; the exact top-k audit shows that recombination-based search exploits reusable high-value synthons particularly well. How this relates to the Klarich et al. (2024) setting, which includes no GA baseline, is discussed in Appendix A.8.

The same pattern holds off docking entirely. Across the ten shared-budget non-docking tasks (Synthetic Properties, Lead Rediscovery, and Lead Optimization; Appendix A.13) the typed GA is the unique top-ranked method (Table 9), and its contrast with the factored bandits is diagnostic. Those bandits post competitive top-1000 DCRF yet *negative* exact best-utility gap closure—they concentrate the budget on a high-utility neighborhood (good local enrichment) without reaching the global optimum (poor exploration). GA wins on both axes at once, closing 0.761 of the exact gap while retaining strong DCRF, which is the operational signature of a balanced exploration/exploitation trade-off rather than a docking-specific quirk.

The budget-response result has a practical reading: method choice should track the oracle's cost. When calls are cheap and plentiful, population search (GA) wins; when calls are precious, factored bandits extract more per call. A benchmark that reported a single budget would hide this—which is why SynthonBench reports the full sweep.

Limitations. SynthonBench grades search efficiency against fixed oracles, not biological truth: most large-scale rows use LightGBM surrogate scores (held-out lowest-0.1% recall 0.153–0.279) rather than measured docking, so rankings may shift under live docking, additional measured-docking tables, or scaffold-disjoint splits. The fixed-table design requires a finite enumerated universe, so we certify behavior only to 100M products and assume—rather than prove—that the strong cross-scale rank agreement extends to the 10^{11}–10^{12}-product spaces used in practice; all synthons also derive from a single make-on-demand space (Freedom Space). Finally, DCRF measures elite-set recovery, not chemotype diversity or out-of-distribution generalization, and does not penalize concentration on a few high-scoring neighborhoods—similarity-expansion methods such as SH are especially prone to this, so their large surrogate-scale gains should be confirmed under diversity-aware, scaffold-disjoint evaluation. We expand on scope, licensing, and future directions in Appendix A.1.

7. Conclusion

SynthonBench evaluates the part of molecular design hidden between molecule generation and experimental follow-up: choosing which synthesizable candidates to evaluate when the chemical space is too large to enumerate. By separating reaction-native search from exact product-id audits, it reports virtual-screening metrics (top-k recall, DCRF, exact-oracle percentage, regret, and Pareto coverage) across surrogate scale, three measured Glide fixed tables, shared-budget Glide Docking tasks, and the non-docking suites. Simple adaptive policies recover substantially more elite products than random sampling, reaction allocation changes recovery, and measured-score tables expose target and oracle sensitivity that surrogate-only evaluations miss. SynthonBench is openly released (`https://github.com/mireklzicar/synthonbench`; `pip install synthonbench`; tag `synthonbench v0.1.0-genbio`; demo at `https://synthonbench.mireklzicar.com/`).

Impact Statement

SynthonBench targets a central bottleneck in early drug discovery: searching the billions-to-trillions of synthesizable make-on-demand compounds for promising candidates. As the DCRF metric makes concrete, adaptive policies recover the same elite products while evaluating orders of magnitude fewer molecules than exhaustive or random screening. Because the search is confined to synthon-defined, make-on-demand spaces, the recovered hits are directly orderable, so this efficiency translates into lower early-phase screening cost and shorter timelines from target to testable compound. By measuring and comparing these methods on a common footing, SynthonBench helps the community build and select the screening engines that deliver that speedup.

As with any virtual-screening result, docking and surrogate scores are proxies, not biological activity, and optimization outputs should pass medicinal-chemistry review, safety assessment, and experimental validation before use. SynthonBench reports fixed-oracle metrics and preserves filter and failure accounting to keep these limitations visible, and scopes wet-lab efficacy, toxicity, and clinical interpretation out of the benchmark itself.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. doi: 10.1038/s41586-024-07487-w.

Bellmann, L., Penner, P., and Rarey, M. Topological similarity search in large combinatorial fragment spaces. *Journal of Chemical Information and Modeling*, 61(1):238–251, 2021.

Bellmann, L., Klein, R., and Rarey, M. Calculating and optimizing physicochemical property distributions of large combinatorial fragment spaces. *Journal of Chemical Information and Modeling*, 62(11):2800–2810, June 2022. ISSN 1549-960X. doi: 10.1021/acs.jcim.2c00334. URL http://dx.doi.org/10.1021/acs.jcim.2c00334.

Bemis, G. W. and Murcko, M. A. The properties of known drugs. 1. molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996. doi: 10.1021/jm9602928.

Beroza, P., Crawford, J. J., Ganichkin, O., Gendelev, L., Harris, S. F., Klein, R., Miu, A., Steinbacher, S., Klingler, F.-M., and Lemmen, C. Chemical space docking enables large-scale structure-based virtual screening to discover ROCK1 kinase inhibitors. *Nature Communications*, 13 (1):6447, 2022. doi: 10.1038/s41467-022-33981-8.

BioSolveIT GmbH. CoLibri, SpaceLight, FTrees and infiniSee. https://www.biosolveit.de/, 2026. Sankt Augustin, Germany.

Breiman, L. Random forests. *Machine Learning*, 45:5–32, 2001.

Brown, N., Fiscato, M., Segler, M. H. S., and Vaucher, A. C. Guacamol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3):1096–1108, 2019.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

Bubeck, S., Munos, R., and Stoltz, G. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*, pp. 23–37, 2009.

Butina, D. Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. *Journal of Chemical Information and Computer Sciences*, 39(4):747–750, 1999. doi: 10.1021/ci9803381.

Cadence Design Systems. Cadence unveils ai-driven ROCS X ai-enabled molecular search, unlocking multi-trillion molecule virtual screening. Press release, 25 September 2025. Cadence newsroom, 2025.

Chen, S., Lin, T., King, I., Lyu, M. R., and Chen, W. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems*, volume 27, pp. 379–387, 2014.

Chen, T. and Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.

Cheng, C. and Beroza, P. Shape-aware synthon search (SASS) for virtual screening of synthon-based chemical spaces. *Journal of Chemical Information and Modeling*, 64(4):1251–1260, 2024. doi: 10.1021/acs.jcim.3c01865.

Coley, C. W., Eyke, N. S., and Jensen, K. F. Autonomous discovery in the chemical sciences part I: Progress. *Angewandte Chemie International Edition*, 59(51):22858–22893, 2020. doi: 10.1002/anie.201909987.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002. doi: 10.1109/4235.996017.

Eisenhuth, P., Liessmann, F., Moretti, R., and Meiler, J. Ultra-large library screening with an evolutionary algorithm in Rosetta (REvoLd). *Communications Chemistry*, 8:335, 2025. doi: 10.1038/s42004-025-01758-x.

Frey, N. C. et al. Lab-in-the-loop therapeutic antibody design with deep learning. *bioRxiv*, 2025. doi: 10.1101/2025.02.19.639050.

Friesner, R. A., Banks, J. L., Murphy, R. B., Halgren, T. A., Klicic, J. J., Mainz, D. T., Repasky, M. P., Knoll, E. H., Shelley, M., Perry, J. K., Shaw, D. E., Francis, P., and Shenkin, P. S. Glide: A new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy. *Journal of Medicinal Chemistry*, 47(7):1739–1749, 2004. doi: 10.1021/jm0306430. URL https://doi.org/10.1021/jm0306430. PMID: 15027865.

Gao, W., Fu, T., Sun, J., and Coley, C. W. Sample efficiency matters: A benchmark for practical molecular optimization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21342–21357, 2022a.

Gao, W., Mercado, R., and Coley, C. W. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. In *International Conference on Learning Representations*, 2022b.

Gao, W., Luo, S., and Coley, C. W. Generative artificial intelligence for navigating synthesizable chemical space. *Proceedings of the National Academy of Sciences*, 122(41): e2415665122, 2025. doi: 10.1073/pnas.2415665122.

Garcia-Ortegon, M., Simm, G. N. C., Tripp, A. J., Hernandez-Lobato, J. M., Bender, A., and Bacallado, S. DOCKSTRING: Easy molecular docking yields better benchmarks for ligand design. *Journal of Chemical Information and Modeling*, 62(15):3486–3502, 2022. doi: 10.1021/acs.jcim.1c01334.

Garnett, R., Krishnamurthy, Y., Xiong, X., Schneider, J., and Mann, R. Bayesian optimal active search and surveying. In *International Conference on Machine Learning*, pp. 843–850, 2012.

Geballe, M. and Lee, A. Extreme scale search with ROCS X: Trillion-scale virtual screening with synthon-driven 3d search. OpenEye Science Brief, Cadence Molecular Sciences. Science brief PDF, 2026. Source: OpenEye miniCUP Presentation, October 2025, Japan.

Gentile, F., Agrawal, V., Hsing, M., Ton, A.-T., Ban, F., Norinder, U., Gleave, M. E., and Cherkasov, A. Deep docking: A deep learning platform for augmentation of structure based drug discovery. *ACS Central Science*, 6 (6):939–949, 2020. doi: 10.1021/acscentsci.0c00229.

Geurts, P., Ernst, D., and Wehenkel, L. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. doi: 10.1007/s10994-006-6226-1.

Gorgulla, C., Boeszoermenyi, A., Wang, Z.-F., Fischer, P. D., Coote, P. W., Das, K. M., Malets, Y. S., Radchenko, D. S., Moroz, Y. S., Scott, D. A., Fackeldey, K., Hoffmann, M., Iavniuk, I., Wagner, G., and Arthanari, H. An open-source drug discovery platform enables ultra-large virtual screens. *Nature*, 580(7805):663–668, 2020.

Graff, D. E., Shakhnovich, E. I., and Coley, C. W. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical Science*, 12:7866–7881, 2021. doi: 10.1039/D0SC06805E.

Grigg, T. G., Burlage, M., Scott, O. B., Taouil, A., Sydow, D., and Wilbraham, L. Active learning on synthons for molecular design, 2025.

Grygorenko, O. O., Radchenko, D. S., Dziuba, I., Chuprina, A., Gubina, K. E., and Moroz, Y. S. Generating multibillion chemical space of readily accessible screening compounds. *iScience*, 23(11):101681, 2020. doi: 10.1016/j.isci.2020.101681.

Hollingsworth, S. A. and Dror, R. O. Molecular dynamics simulation for all. *Neuron*, 99(6):1129–1143, 2018. doi: 10.1016/j.neuron.2018.08.011.

Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Advances in Neural Information Processing Systems*, volume 34, pp. 3595–3607, 2021.

Jensen, J. H. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019. doi: 10.1039/C8SC05372C.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Kalliokoski, T. Machine learning boosted docking (HASTEN): An open-source tool to accelerate structure-based virtual screening campaigns. *Molecular Informatics*, 40 (9):2100089, 2021. doi: 10.1002/minf.202100089.

Kalliokoski, T., Turku, A., and Käsnänen, H. SpaceHASTEN: A structure-based virtual screening tool for nonenumerated virtual chemical libraries. *Journal of Chemical Information and Modeling*, 65(1):125–132, 2025. doi: 10.1021/acs.jcim.4c01790. URL https://doi.org/10.1021/acs.jcim.4c01790. PMID: 39710946.

Kaplan, A. L., Confair, D. N., Kim, K., Barros-Álvarez, X., Rodriguiz, R. M., Yang, Y., Kweon, O. S., Che, T., McCorvy, J. D., Kamber, D. N., Phelan, J. P., Martins, L. C., Pogorelov, V. M., DiBerto, J. F., Slocum, S. T., Huang, X.-P., Kumar, J. M., Robertson, M. J., Panova, O., Seven, A. B., Wetsel, A. Q., Wetsel, W. C., Irwin, J. J., Skiniotis, G., Shoichet, B. K., Roth, B. L., and Ellman, J. A. Bespoke library docking for 5-HT2A receptor agonists with antidepressant activity. *Nature*, 610(7932): 582–591, 2022. doi: 10.1038/s41586-022-05258-z.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Klarich, K., Goldman, B., Kramer, T., Riley, P., and Walters, W. P. Thompson sampling: An efficient method for searching ultralarge synthesis on demand databases. *Journal of Chemical Information and Modeling*, 64(4):1158–1171, 2024. doi: 10.1021/acs.jcim.3c01790. URL https://doi.org/10.1021/acs.jcim.3c01790.

Kohn, W. and Sham, L. J. Self-consistent equations including exchange and correlation effects. *Physical Review*, 140(4A):A1133–A1138, 1965. doi: 10.1103/PhysRev.140.A1133.

Koziarski, M., Rekesh, A., Shevchuk, D., van der Sloot, A., Gaiński, P., Bengio, Y., Liu, C.-H., Tyers, M., and Batey, R. A. RGFN: Synthesizable molecular generation using GFlowNets. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Landrum, G. RDKit: Open-source cheminformatics. https://www.rdkit.org/, 2025.

Liu, F., Mailhot, O., Glenn, I. S., Vigneron, S. F., Bassim, V., Xu, X., Fonseca-Valencia, K., Smith, M. S., Radchenko, D. S., Fraser, J. S., Moroz, Y. S., Irwin, J. J., and Shoichet, B. K. The impact of library size and scale of testing on virtual screening. *Nature Chemical Biology*, 21(7):1039–1045, 2025. doi: 10.1038/s41589-024-01797-w.

Lübbers, J., Lessel, U., and Rarey, M. Enhanced calculation of property distributions in chemical fragment spaces. *Journal of Chemical Information and Modeling*, 64(6):2008–2020, Mar 2024. ISSN 1549-960X. doi: 10.1021/acs.jcim.4c00147. URL http://dx.doi.org/10.1021/acs.jcim.4c00147.

Luttens, A., Cabeza de Vaca, I., Sparring, L., Brea, J., Martinez, A. L., Kahlous, N. A., Radchenko, D. S., Moroz, Y. S., Loza, M. I., Norinder, U., and Carlsson, J. Rapid traversal of vast chemical space using machine learning-guided docking screens. *Nature Computational Science*, 5:301–312, 2025. doi: 10.1038/s43588-025-00777-x.

Lyu, J., Wang, S., Balius, T. E., Singh, I., Levit, A., Moroz, Y. S., O'Meara, M. J., Che, T., Algaa, E., Tolmachova, K., Tolmachev, A. A., Shoichet, B. K., Roth, B. L., and Irwin, J. J. Ultra-large library docking for discovering new chemotypes. *Nature*, 566(7743):224–229, 2019. doi: 10.1038/s41586-019-0917-9.

Moesgaard, L. and Kongsted, J. Introducing SpaceGA: A search tool to accelerate large virtual screenings of combinatorial libraries. *Journal of Chemical Information and Modeling*, 64(21), 2024. doi: 10.1021/acs.jcim.4c01308.

Muchmore, S. W., Debe, D. A., Metz, J. T., Brown, S. P., Martin, Y. C., and Hajduk, P. J. Application of belief theory to similarity data fusion for use in analog searching and lead hopping. *Journal of Chemical Information and Modeling*, 48(5):941–948, 2008. doi: 10.1021/ci7004498.

OpenEye, Cadence Molecular Sciences. ROCS X: AI-enabled molecular search of trillions of products. ACS on-demand presentation. ACS recording, 2025. Public presentation in which OpenEye/Cadence describe ROCS X's use of Bayesian bandits; accessed 27 June 2026.

OpenEye, Cadence Molecular Sciences. ROCS X: Search trillions. find novel hits from beyond the known. OpenEye product page, 2026. Accessed 25 June 2026.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., Kwabi-Addo, D., Beaini, D., Jaakkola, T., and Barzilay, R. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025. doi: 10.1101/2025.06.14.659707.

Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Johansson, S., Chen, H., Nikolenko, S., Aspuru-Guzik, A., and Zhavoronkov, A. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Frontiers in Pharmacology*, 11:565644, 2020. doi: 10.3389/fphar.2020.565644.

Rarey, M. and Stahl, M. Similarity searching in large combinatorial chemistry spaces. *Journal of Computer-Aided Molecular Design*, 15:497–520, 2001.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

Sadybekov, A. V., Sadybekov, A. A., Liu, Y., Iliopoulos-Tsoutsouvas, C., Huang, X.-P., Pickett, J., Houser, B., Patel, N., Tran, N. K., Tong, F., Zvonok, N., Jain, M. K., Sürmeli, N., Roth, B. L., Dror, R. O., Shoichet, B. K., and Katritch, V. Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. *Nature*, 601(7893):452–459, 2022.

Sivula, T., Yetukuri, L., Kalliokoski, T., Käsnänen, H., Poso, A., and Pöhner, I. Machine learning-boosted docking enables the efficient structure-based virtual screening of giga-scale enumerated chemical libraries. *Journal of Chemical Information and Modeling*, 63(18):5773–5783, 2023. doi: 10.1021/acs.jcim.3c01239. URL https://doi.org/10.1021/acs.jcim.3c01239.

Stein, R. M., Kang, H. J., McCorvy, J. D., Glatfelter, G. C., Jones, A. J., Che, T., Slocum, S., Huang, X.-P., Savych, O., Moroz, Y. S., Stauch, B., Johansson, L. C., Cherezov, V., Kenakin, T., Irwin, J. J., Shoichet, B. K., Roth, B. L., and Dubocovich, M. L. Virtual discovery of melatonin receptor ligands to modulate circadian rhythms. *Nature*, 579(7800):609–614, 2020. doi: 10.1038/s41586-020-2027-0.

Swanson, K., Liu, G., Catacutan, D. B., Arnold, A., Zou, J., and Stokes, J. M. Generative ai for designing and validating easily synthesizable and structurally novel antibiotics. *Nature Machine Intelligence*, 6(3):338–353, 2024. doi: 10.1038/s42256-024-00809-7. URL https://doi.org/10.1038/s42256-024-00809-7.

Tang, Y., Moretti, R., and Meiler, J. RosettaAMRLD: A reaction-driven approach for structure-based drug design from combinatorial libraries with monte carlo metropolis algorithms. *Journal of Chemical Information and Modeling*, 65(12):5945–5959, 2025. doi: 10.1021/acs.jcim.5c00497. URL https://doi.org/10.1021/acs.jcim.5c00497.

Wang, L., Wu, Y., Deng, Y., Kim, B., Pierce, L., Krilov, G., Lupyan, D., Robinson, S., Dahlgren, M. K., Greenwood, J., Romero, D. L., Masse, C., Knight, J. L., Steinbrecher, T., Beuming, T., Damm, W., Harder, E., Sherman, W., Brewer, M., Wester, R., Murcko, M., Frye, L., Farid, R., Lin, T., Mobley, D. L., Jorgensen, W. L., Berne, B. J., Friesner, R. A., and Abel, R. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *Journal of the American Chemical Society*, 137(7):2695–2703, 2015. doi: 10.1021/ja512751q.

Warr, W. A., Nicklaus, M. C., Nicolaou, C. A., and Rarey, M. Exploration of ultralarge compound collections for drug discovery. *Journal of Chemical Information and Modeling*, 62(9):2021–2034, 2022. doi: 10.1021/acs.jcim.2c00224.

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., Palmer, A., Settels, V., Jaakkola, T., Jensen, K., and Barzilay, R. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. doi: 10.1021/acs.jcim.9b00237.

Yang, Y., Yao, K., Repasky, M. P., Leswing, K., Abel, R., Shoichet, B. K., and Jerome, S. V. Efficient exploration of chemical space with docking and deep learning. *Journal of Chemical Theory and Computation*, 17(11):7106–7119, 2021. doi: 10.1021/acs.jctc.1c00810.

A. Appendix: Benchmark and Artifact Details

A.1. Scope and Future Work

Surrogate and oracle scope. Most large-scale docking rows use LightGBM surrogate predictions rather than measured docking or wet-lab assays; we treat these rows as structured benchmark oracles, with held-out lowest-0.1% recall of 0.153–0.279 across targets. The 10M and 100M docking-surrogate suites are table-backed exact-audit runs rather than model-backed query-time runs, because model-backed scoring was not practical for full sweeps at those scales; they establish runtime, memory, exact-audit plumbing, and scale behavior. We did not run wet-lab validation, and how well docking itself recovers true binders is out of scope for this benchmark: SynthonBench measures search efficiency against fixed oracles, not the biological correctness of those oracles, so method rankings may change under scaffold-disjoint splits, additional measured-docking tables, or live docking.

Fixed-table extrapolation and licensing. Precomputing oracle values requires a finite enumerated universe, so the benchmark measures search efficiency on spaces up to 100M products and implicitly assumes that the relative behavior of methods extrapolates to the far larger spaces used in practice, without an unexpected plateau. This is an assumption, not a guarantee: results on a 100M space need not be identical at 10^{11}–10^{12} products. It is reassuring that method efficiency and rankings correlate strongly across the 1M/10M/100M scales (cross-run Spearman ρ from 0.86 to 0.95), so even if absolute reduction factors shift at deployment scale, the benchmark remains well suited to promoting the most promising approaches. The license-free property is likewise scoped: it covers bringing a new search method and scoring it against the released tables, not reproducing the licensed-backend baselines (e.g. the FTrees/SpaceLight SpaceHASTEN rows) or regenerating the source oracle tables, both of which still require the original docking and fragment-space software. Finally, all synthon sets are derived from a single make-on-demand space (Freedom Space), which was the library shared with us; we therefore cannot guarantee that every finding transfers to spaces built from different synthons or reaction templates, although the breadth of the 42 reactions and the consistency of the cross-scale trends make broad generalization likely.

Future work. The most direct extensions follow the trends this benchmark exposes. Scaling the exact-audit protocol toward full-size production make-on-demand libraries (10^{11}–10^{12} products), where sample-efficient search matters most, would test whether the monotone cross-scale efficiency trend continues past 100M. The budget-response analysis (Figure 1) motivates methods that perform well across budget regimes rather than at a single operating point, since the leading method changes with budget. Integrating more expensive and higher-fidelity oracles—live docking, physics-based scoring, and ultimately lab-in-the-loop assays—would close the gap between fixed-table search efficiency and prospective experimental value. Further directions include additional measured Glide Docking targets, larger budget sweeps, medchem-filtered universes with alternative property profiles, SpaceProp-style exact property distributions, SMARTS-pattern coverage, scaffold-disjoint surrogate training, diversity-aware follow-up analyses, an exact SALSA reproduction, larger SpaceHASTEN matrices, and prospective task-to-target validation.

A.2. Supplementary Result Figures

These figures and tables support the main-text Results section. Figure 3 is the cross-run transfer matrix (surrogate scales and Glide Docking) and Table 3 is the enumerated-pool external-baseline reference comparison; the small-budget retuning control behind the main-text budget-response columns of Table 2 is reported below.

Budget-response provenance (Figure 1). Budget points at or below 10,000 calls are read from the primary leaderboard traces, so the 10,000-call point reproduces the Figure 2 leaderboard. Larger budgets use extended 100,000-call campaigns on the identical oracle, with two coverage gaps where compute was exhausted: Iterative Surrogate Expansion is run only to 10,000 calls on PYRD and TGFR1, and the ROCS-X-inspired bandit's 100,000-call PYRD point averages four seeds. The SpaceHASTEN XGB (SH) curve spans 250–50,000 calls; each point uses a budget-matched curriculum (cycle budget $= B/5$, an $\approx$20% random warm-up at every budget) for a fair low-budget comparison, and its 100,000-call point is omitted because the FTrees/SpaceLight search saturates the medchem-filtered library before then. The HASTEN reference peaks at $45.6\times$ top-1000 DCRF near 2,500 calls, above every synthon-space method's peak.

Small-budget retuning control. The peak budget in Table 2 differs sharply across methods—1,000 calls for SALSA and the reagent-prior Thompson sampler, 2,500 for the Genetic Algorithm, the ROCS-X-inspired bandit, and ISE, and only 10,000 for the slower-starting random forest—so the headline 10,000-call leader (the Genetic Algorithm) is neither the highest-peaking nor the most budget-efficient method, both of which are SALSA. To verify that the small-budget ordering is not an artifact of configuring every method at the 10,000-call operating point, we retuned the budget-sensitive knobs of each method directly for a 1,000-call budget (three targets, identical oracle and grading). Only the Genetic Algorithm carried a meaningful penalty: halving its population from 96 to 48 raises its mean top-1000 DCRF from $18.0\times$ to $27.0\times$, confirming that its

Table 3. Enumerate-then-learn references on the Glide Docking 1M tasks. The HASTEN and MolPAL rows use enumerate-then-learn pools with 10,000 charged fixed-table lookups, and require full-library enumeration before search, which is why they are kept out of the main non-enumerated leaderboard. Where full enumeration is feasible, HASTEN is strong (mean top-1000 DCRF 37.49, top-1000 recall 0.378), indicating that an enumerate-then-screen protocol can match or exceed the synthon-space methods in that regime. MolPAL+XGB is weaker here (mean top-1000 DCRF 1.62, top-1000 recall 0.016) than we expected; we read this as warranting further investigation or hyperparameter tuning rather than as a ceiling, since MolPAL is a more general framework that exposes many configurable options (surrogate model, acquisition function, and batch/iteration schedule) that we did not exhaustively tune for this benchmark. DCRF is top-1000 docking-call reduction factor; Top1000 and Top100 are exact-recall values at the same 10,000-call budget. The same HASTEN protocol run against the 1M/10M/100M docking-surrogate scales is reported in the HASTEN‡ row of Table 2.

Method	DCRF	Top1000	Top100	Best score	Regret	Seeds
HASTEN	37.49	0.378	0.421	9.60	0.233	5
MolPAL + XGB	1.62	0.016	0.010	8.05	1.789	5

default population is calibrated to larger budgets. Even so, the retuned Genetic Algorithm does not reach SALSA ($39.6\times$), which still leads on all three targets; shrinking SALSA's or ISE's warmup only hurts them and the random forest is essentially unchanged. The low-budget advantage of factored acquisition is therefore intrinsic to the algorithm class rather than a hyperparameter artifact.

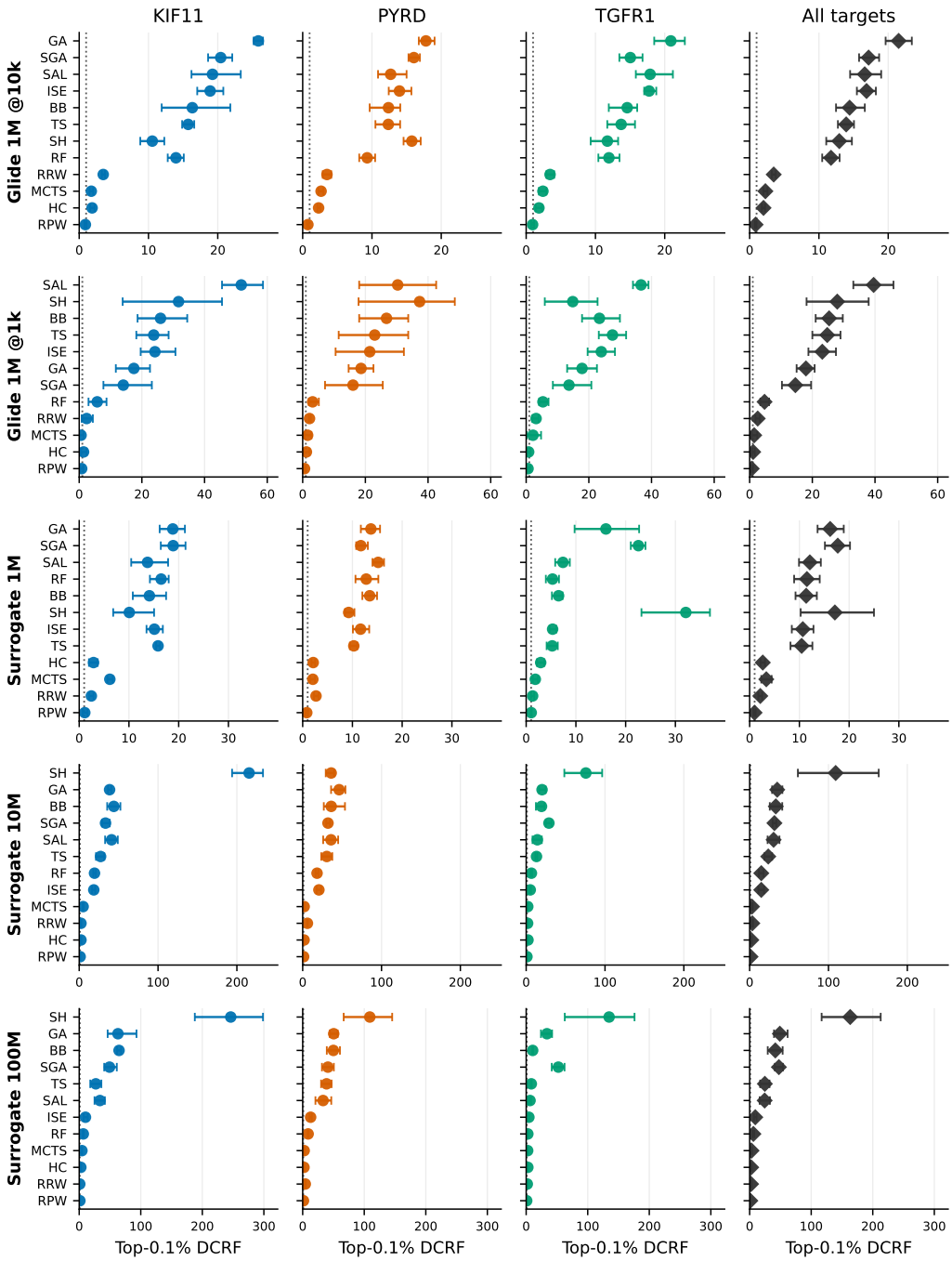


Figure 2. Target-resolved top-0.1% DCRF across protocols, for the non-enumerated method set of Table 2 (the enumerate-then-learn HASTEN reference is excluded). Rows are the medchem-filtered Glide Docking 1M endpoint at the 10,000- and 1,000-call budgets (*Glide 1M @10k*/*@1k*, the two vertical guides in Figure 1) and the 1M, 10M, and 100M docking-surrogate scales; the first three columns are the targets (KIF11, PYRD, TGFR1) and the last column (dark diamonds) is the target-aggregate. In the per-target columns points are means with 95% bootstrap confidence intervals over seeds; the aggregate column instead bootstraps the full 15-run population (per method), which is the more stable estimator and the one used for the significance tests in Table 2. Within each row the methods are sorted by mean rank across the three targets, so the strongest method is on top, and each row uses its own x-axis because the elite-set DCRF magnitudes differ across protocols and grow with the surrogate universe (the audited elite set is the top 1,000 at 1M, top 10,000 at 10M, and top 100,000 at 100M; for the 1M Glide endpoint top-1000 is the top-0.1% audit). The dotted line marks random-equivalent efficiency (DCRF = 1). Efficiency rises with chemical-space size: at 100M the strongest synthon-space methods exceed top-0.1% DCRF of 60 on individual targets (typed GA 62.9 and the ROCS-X-inspired bandit 64.7 on KIF11). The enumerate-then-learn HASTEN reference is a separate full-enumeration ceiling (dashed line in Figure 1) and is omitted here. Exact aggregate values are tabulated in Table 2. Method abbreviations follow the method set (Section 3): SAL is an in-house tabular Thompson analogue of SALSA, BB is the ROCS-X-inspired Bayesian bandit baseline (not the closed-source ROCS X; see Section A.8), and SH (SpaceHASTEN XGB) is the non-enumerated SpaceHASTEN reference; its surrogate-free ablation SHC is reported in Appendix A.10. The figure carries the same roster as Table 2 (the external SpaceGA tool and SH are shown on all five rows); the SH surrogate-scale points are the 3-seed FT/SL audit and equal the Table 2 surrogate values. All values are also tabulated in Table 2.

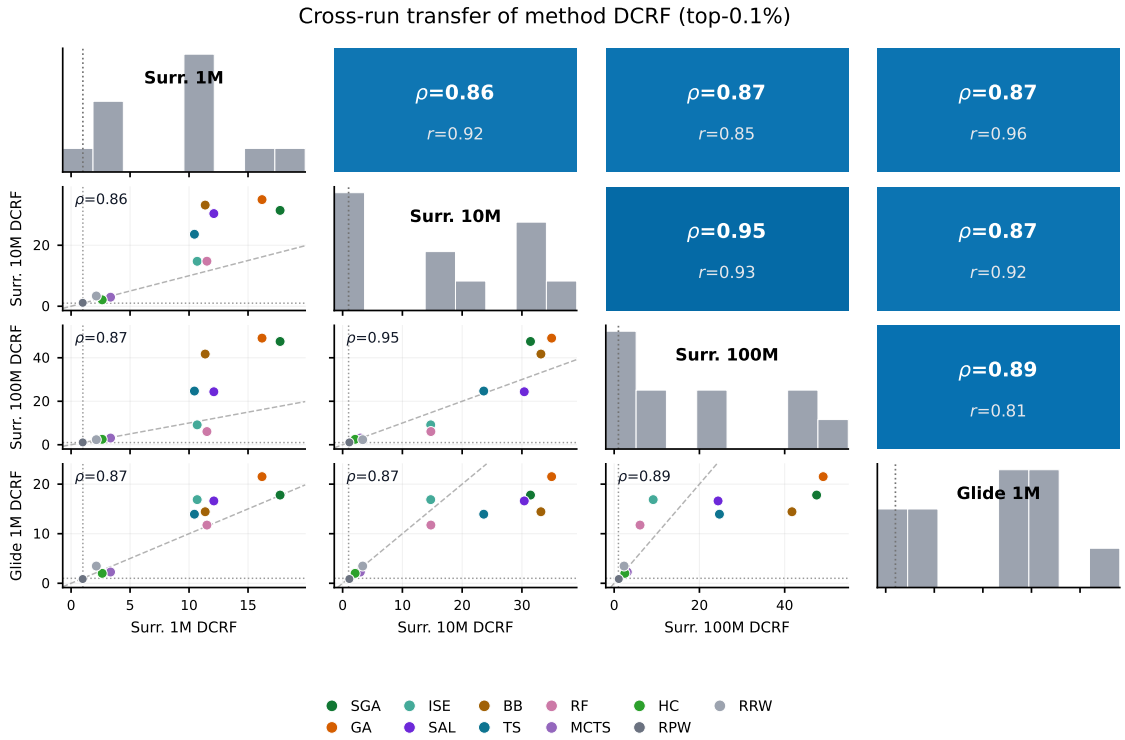


Figure 3. Cross-run transfer of method efficiency across all four runs. Each off-diagonal panel compares per-method DCRF (top-0.1%: top-1000 for the two 1M universes, top-10,000 and top-100,000 for the 10M and 100M surrogate suites) between two runs over the 11 shared methods. Lower triangle: pairwise scatter (one point per method, colored as in the legend) with the dashed identity line and dotted DCRF$=1$ random baselines; diagonal: the DCRF distribution across methods for that run; upper triangle: Spearman ρ and Pearson r, shaded by $|\rho|$. Method rankings transfer strongly between every pair of runs (Spearman ρ from 0.86 to 0.95), including from the 1M surrogate to the medchem-filtered Glide Docking endpoint ($\rho = 0.84$). Method abbreviations follow the method-set overview (Section 3).

A.3. Glide Docking Diagnostics

Table 4 keeps the Glide-only diagnostics from the medchem-filtered lipophilicity-penalized leaderboard (top-1000 recall, best score, and submitted-call validity), now reported by abbreviation. The cross-protocol top-0.1% DCRF summary behind Figure 2 (Table 2) is in the main text, and method names and roles are defined in Section 3.

Table 4. Glide Docking diagnostics for the medchem-filtered lipophilicity-penalized benchmark, by abbreviation (the former main leaderboard with the full method-name column removed). Each target exposes all 42 reactions under one shared 10,000-call budget; values average KIF11, PYRD, and TGFR1 over the released runs. Top1000 is exact top-1000 recall; Best score is $-\mathrm{Glide} - \mathrm{cLogP}$; Valid % is the submitted-call fraction that both passes the filter and has a finite successful Glide score; Filter % ignores the Glide-success condition. Rows are ordered by mean top-1000 DCRF (the ordering column of Table 2), not by the displayed top-1000 recall; the two can disagree for SH/SHC, which search the reduced 38/42-reaction CoLibri universe and so normalize DCRF by a smaller N. Method names and roles are defined in Section 3.

Abbr.	Top1000	Best score	Valid %	Filter %
GA	0.217	9.62	45.7	48.2
SGA	0.173	9.60	38.0	40.7
ISE	0.170	8.88	58.8	61.2
SAL	0.168	8.95	66.1	68.6
BB	0.146	8.77	58.1	60.5
TS	0.141	9.12	53.2	55.6
SH	0.111	9.11	38.3	41.3
RF	0.119	9.12	54.9	57.5
SHC	0.076	8.64	28.6	33.1
RRW	0.035	8.68	35.3	37.9
MCTS	0.023	8.69	16.7	18.5
HC	0.020	8.61	14.3	15.8
RPW	0.009	8.11	11.6	13.1

A.4. Exact-Audit Semantics

SynthonBench separates the optimizer-facing task from the reader-facing audit. During a run, an optimizer only proposes a reaction id and a valid tuple of synthon ids, and receives one scalar utility per charged oracle call. For score-table suites, exact metrics are attached after the run by joining the queried product ids against the reference score table for the same reaction and objective. Thus "exact" means exact with respect to the reference score table and its stable product ids; it does not mean measured docking, biological activity, synthesis success, toxicity, or clinical relevance.

For docking-surrogate score tables, raw SpaceHASTEN-derived predicted docking scores are minimized. The benchmark converts them to utilities by negating the raw scores, so all optimizers maximize utility and all reported best-utility curves have the same sign convention. For a task with reaction r, objective column c, and queried product-id set Q, top-k recall is computed from the sorted score-table reference $\mathrm{Top}_k(r, c)$ as

$$\text{top-}k \text{ recall} = \frac{|Q \cap \mathrm{Top}_k(r, c)|}{|\mathrm{Top}_k(r, c)|}.$$

Top-k DCRF is $top - krecall/(B/N)$, where B is the charged oracle-call budget and N is the exact reaction-space size. It is the number of random-equivalent oracle calls needed to recover the same top-k yield, di-

vided by the method budget. The exact audit also records $\mathrm{DCRF}_k^\star$, the best DCRF any method could obtain with the same budget by querying exact top-k products first, and $\mathrm{DCRF}_k/\mathrm{DCRF}_k^\star$. These fields are named `oracle_call_reduction_factor_top`k and `pct_oracle_call_reduction_factor_top`k in the CSV outputs. Enrichment is the analogous observed top-0.1% hit rate in Q divided by the score-table top-0.1% base rate, and regret is $\max(0, u^\star - \max_{p\in Q} u(p))$, where $u^\star$ is the best table utility for the reaction and objective.

The random baseline and the equivalent metric forms follow from uniform sampling without replacement: for budget B and exact top-k set T_k, $H = |Q \cap T_k| \sim \mathrm{Hypergeometric}(N, k, B)$, so $\mathbb{E}[top - krecall] = B/N$ and the random DCRF is 1. Writing top-k random-equivalent oracle calls as $\mathrm{REOC}_k = top - krecall\, N$ gives $\mathrm{DCRF}_k = \mathrm{REOC}_k/B = top - krecall/(B/N)$, which is exactly the enrichment factor obtained by treating T_k as hits and the charged query set Q as the selected set,

$$\mathrm{EF}_k(Q) = \frac{|Q \cap T_k|/B}{k/N} = \frac{top - krecall}{B/N} = \mathrm{DCRF}_k.$$

The exact-oracle ceiling at the same budget is $\mathrm{DCRF}_k^\star = \min(1, B/|\mathrm{Top}_k|)\, N/B$; its value for the 1M benchmark ($99.1\times$ at $k = 1000$) and the resulting normalized $\mathrm{DCRF}_k/\mathrm{DCRF}_k^\star$ are reported in Section 2.

For selectivity tasks, the optimizer still receives a scalar utility for one target. The exact Pareto metrics in the main paper are post-hoc audits of the queried set against two target utilities. The reported Pareto-GA variant is the exception: it uses the score table to rank the maintained population by non-dominated sorting and crowding distance as in NSGA-II (Deb et al., 2002), while still paying one benchmark query per proposed product.

A.5. Glide Docking Fixed-Table Configuration

The Glide Docking benchmark uses fixed score tables during optimization: methods do not launch Glide on-line. The score tables are generated once, joined by stable product ids, and then converted into the medchem-filtered benchmark utility used by the in-loop task. Table 5 summarizes the configuration visible in the compact run summaries and filter reference.

A.6. Glide Docking Portfolio and Chemotype Analyses

The appendix keeps the portfolio analyses that use the same medchem-filtered $-\mathrm{Glide}-\mathrm{cLogP}$ utility as the main Glide Docking benchmark. Earlier single-reaction size, phase, locality, fixed-1000 ablation, raw-score portfolio, and external-baseline outputs are retained as supplementary reproducibility material, while the paper figures focus on shared-budget multi-reaction tasks.

Together, these plots make the filtered oracle inspectable. The typed-GA grid shows which products are actually retrieved under the in-loop lipophilicity-penalized objective, while the oracle grids show that high raw Glide scores are not sufficient: products must also pass the medchem structural/property profile and the cLogP penalty. The Butina representatives use a strict Morgan-fingerprint Tanimoto cutoff of 0.30, so the displayed molecules are nonredundant representatives of the penalized oracle rather than near-duplicate top hits. Supplementary CSV tables export the same products for follow-up chemotype analysis.

A.7. Implementation Scope

This paper keeps method details only where they change the scientific interpretation: whether a row is in-house, source-reproduced, scoped single-target context, or a context-only reference entry. Full optimizer defaults remain in the code release and metadata rather than as a long appendix table, because several earlier single-reaction-only methods are retained for reproducibility but are not part of the main benchmark result. The runner still enforces the same unique-query guard for every method: duplicate proposals are retried briefly and then replaced by unseen random samples, so exact tables audit unique charged oracle calls.

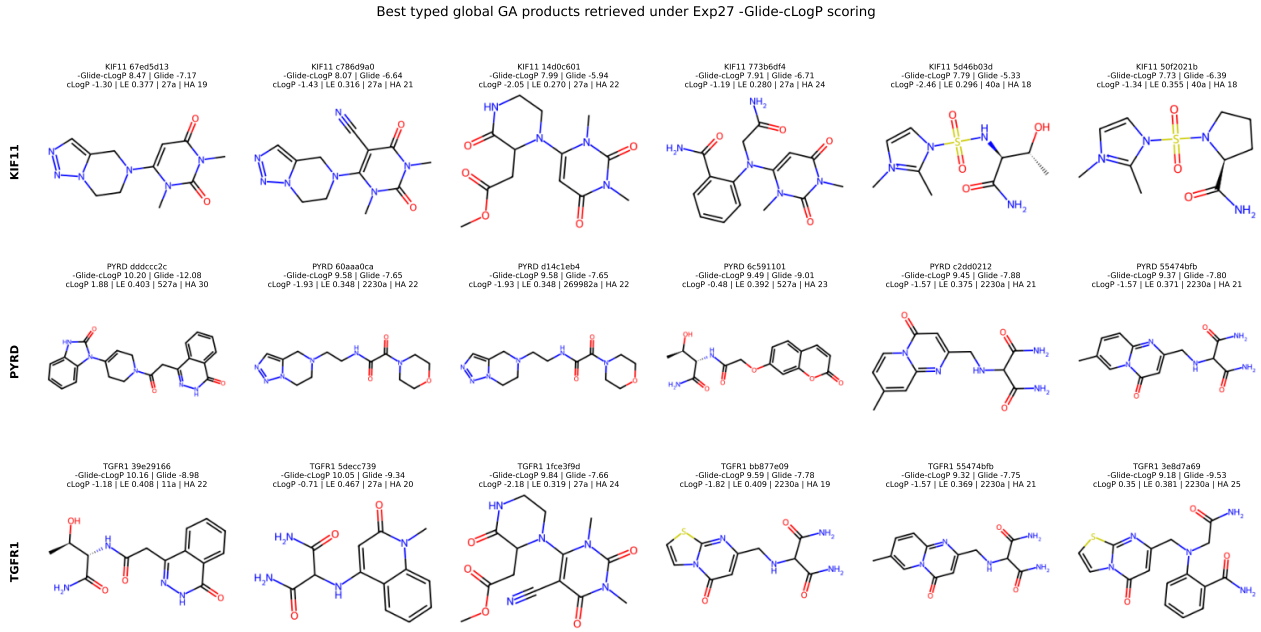


Figure 4. Best products retrieved by the typed GA in the seed-0 medchem-filtered Glide Docking lipophilicity-penalized benchmark. Each row shows the top retrieved $-\text{Glide} - \text{cLogP}$ products for one target after joining the method output back to the filtered score table; labels report the penalized score, raw Glide score, cLogP, ligand efficiency, reaction id, and heavy-atom count.

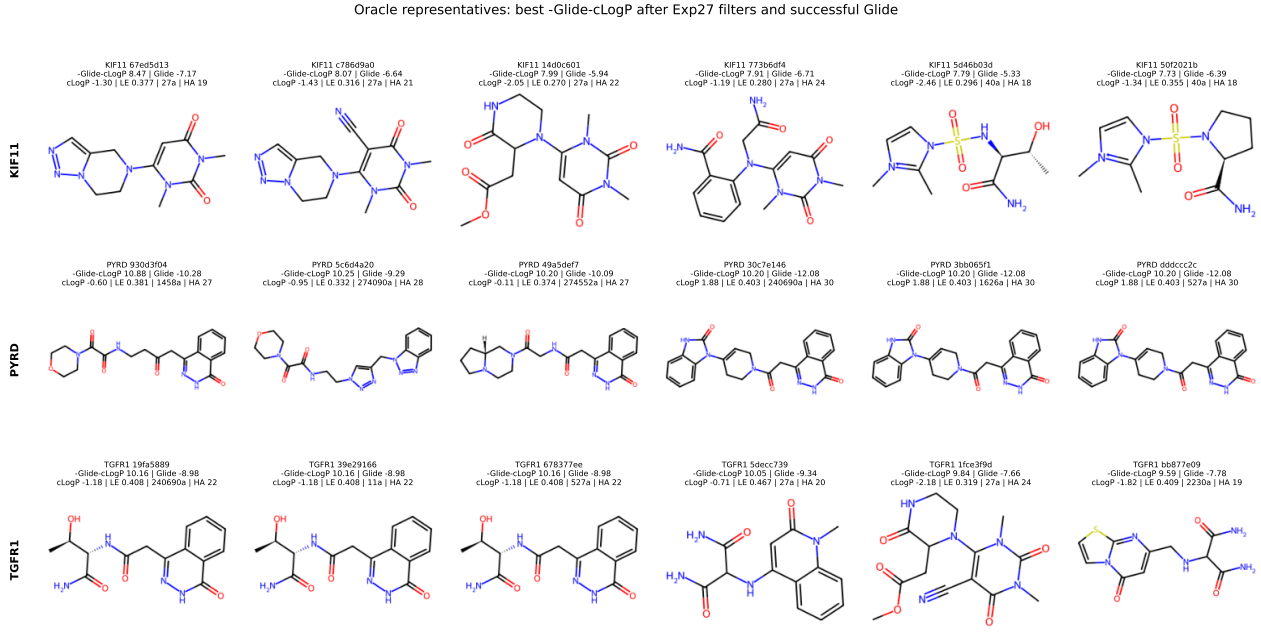


Figure 5. Representative oracle products for the medchem-filtered Glide Docking lipophilicity-penalized benchmark. Each row shows the top $-\text{Glide} - \text{cLogP}$ products for one target after requiring both medchem-filter pass and successful finite Glide score; labels report the penalized score, raw Glide score, cLogP, ligand efficiency, reaction id, and heavy-atom count.

Table 5. Configuration summary for the 1M Glide Docking fixed-table run and the medchem-filtered lipophilicity-penalized benchmark layer. The compact run summaries record Glide HTVS, OPLS_2005, Epik, pH, batching, and worker settings; they do not expose a separate all-target executable version string.

Field	Configuration
Docking source	Target-specific fixed Glide HTVS score tables for KIF11, PYRD, and TGFR1, generated over the 1M synthon enumeration and consumed by the benchmark as lookup tables.
Recorded docking settings	Precision HTVS; force field OPLS_2005; Epik enabled at pH 7.4; one pose retained; 10,000-row batches; chunk size 512; LigPrep and Glide worker settings recorded per target.
Target runs	KIF11: 990,829 input rows, 970,529 successful Glide rows, 13 Glide workers and 16 LigPrep workers. PYRD: 990,829 input rows, 550,895 successful Glide rows, 5 Glide workers and 16 LigPrep workers. TGFR1: 990,829 input rows, 557,413 successful Glide rows, 13 Glide workers and 16 LigPrep workers.
Search universe	990,610 deduplicated benchmark products after product-id canonicalization across 42 reaction templates. Each optimizer receives 10,000 charged submissions per target.
Property filters	The exact medchem product filter requires MW ≤ 500, $-10 \leq$ cLogP ≤ 5, HBA ≤ 10, HBD ≤ 5, rotatable bonds ≤ 10, $0.10 \leq$ fraction Csp3 ≤ 0.80, TPSA ≤ 140, aromatic rings 1–5, fluorine count ≤ 1, and amide count ≤ 2.
Structure filters	Products containing nitro, aldehyde, acyl chloride, thiol, Michael-acceptor enone, vinyl sulfone, acrylonitrile, or nitro-alkene SMARTS patterns fail the exact medchem filter.
Benchmark utility	Valid products maximize $-\mathrm{Glide}-\mathrm{cLogP}$. Filter failures, Glide failures/rejections, NaNs, non-finite logP values, and zero-heavy-atom rows receive utility 0.
Audit semantics	Exact top-k sets are computed from the same valid, filter-passing fixed table and joined to each query record by product id; representative products and Pareto variants use the same filtered utility.

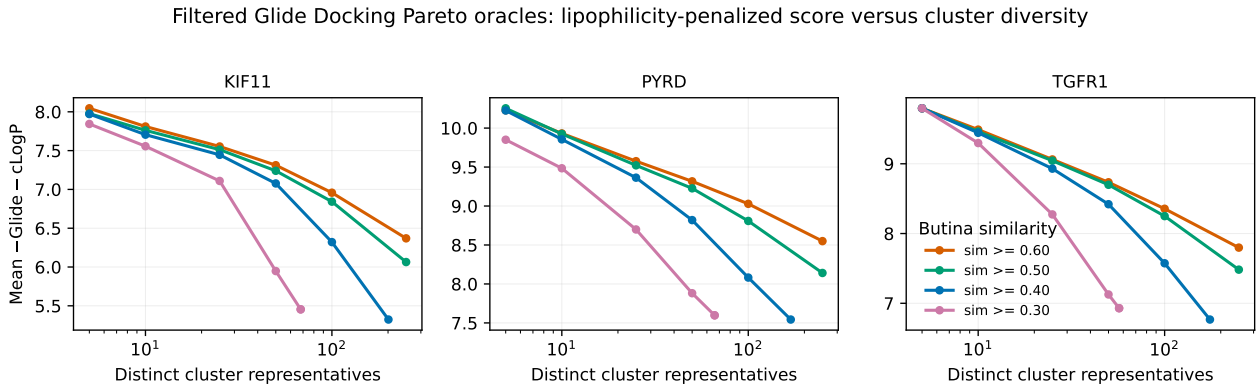


Figure 6. Pareto-oracle fronts for filtered Glide Docking lipophilicity-penalized score under cluster diversity constraints. For each target, the reference pool is the top 5,000 valid filter-passing Glide products by $-\mathrm{Glide}-\mathrm{cLogP}$. Lines select the best representative from each Butina cluster (Butina, 1999) at Morgan-fingerprint Tanimoto similarity cutoffs 0.60, 0.50, 0.40, and 0.30, then greedily remove later representatives that are at least as similar to an earlier higher-scoring representative at the same cutoff. The plot reports mean penalized score as the requested number of nonredundant representatives increases.

Table 6. Filtered lipophilicity-penalized oracle sizes for the medchem-filtered Glide Docking benchmark. Valid products pass the exact medchem filter and have finite successful Glide scores; Best score is $-\mathrm{Glide}-\mathrm{cLogP}$.

Target	Valid products	Valid %	Best score	Glide	Product
KIF11	126,088	12.7	8.47	-7.17	67ed5d13
PYRD	104,429	10.5	10.88	-10.28	930d3f04
TGFR1	108,595	11.0	10.16	-8.98	19fa5889

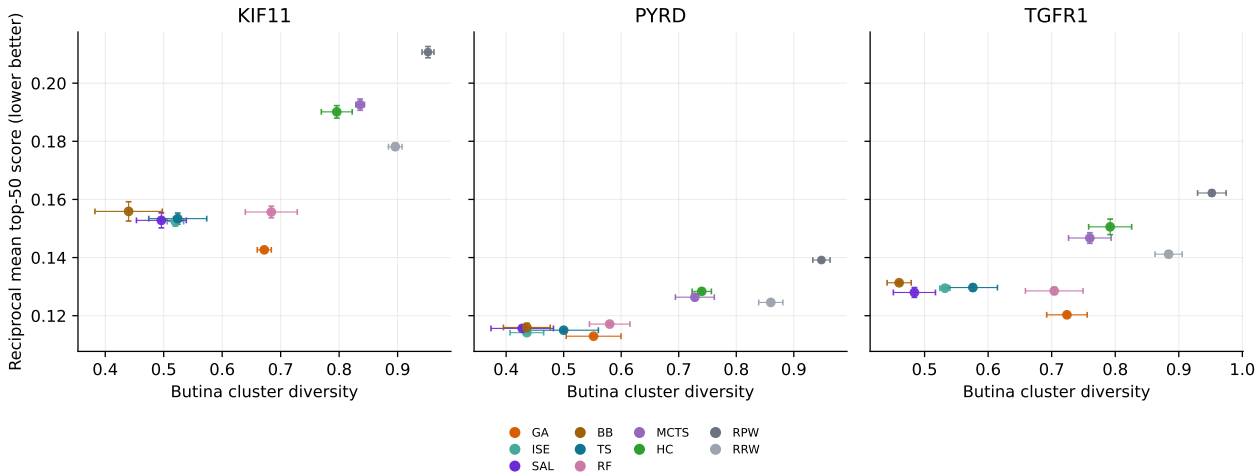


Figure 7. Glide Docking method score/diversity variant with Butina cluster diversity on the x-axis and reciprocal mean top-50 $-$Glide $-$ cLogP score on the y-axis. Lower y is better and higher x is more diverse. Points summarize each method's top 50 valid retrieved products, while the black The shared legend gives method abbreviations and colors.

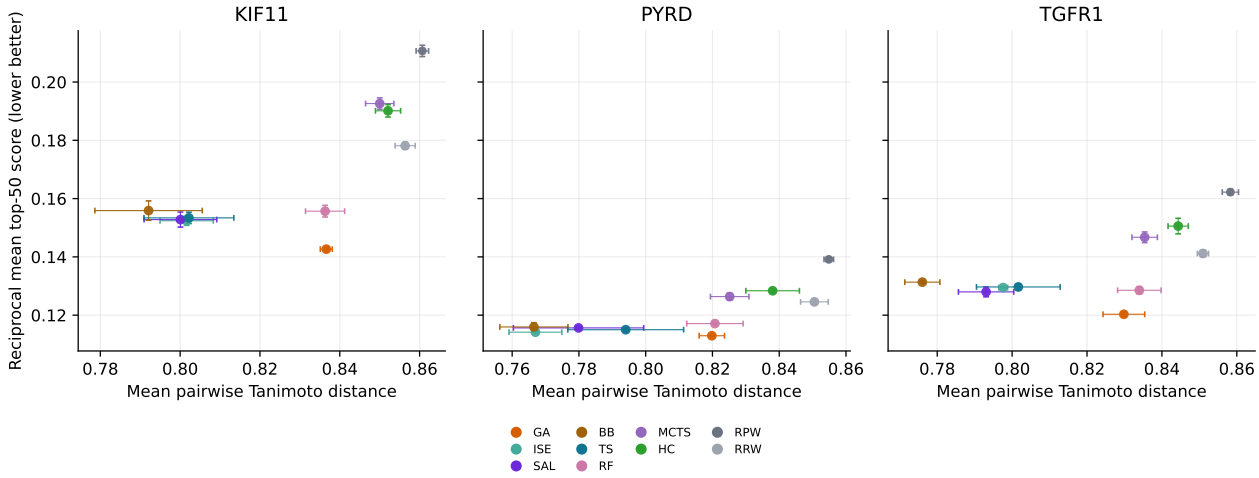


Figure 8. Glide Docking method score/diversity variant with mean pairwise Tanimoto distance on the x-axis. The plot uses the same top-50 valid retrieved products, reciprocal score orientation, and shared method legend as Figure 7.

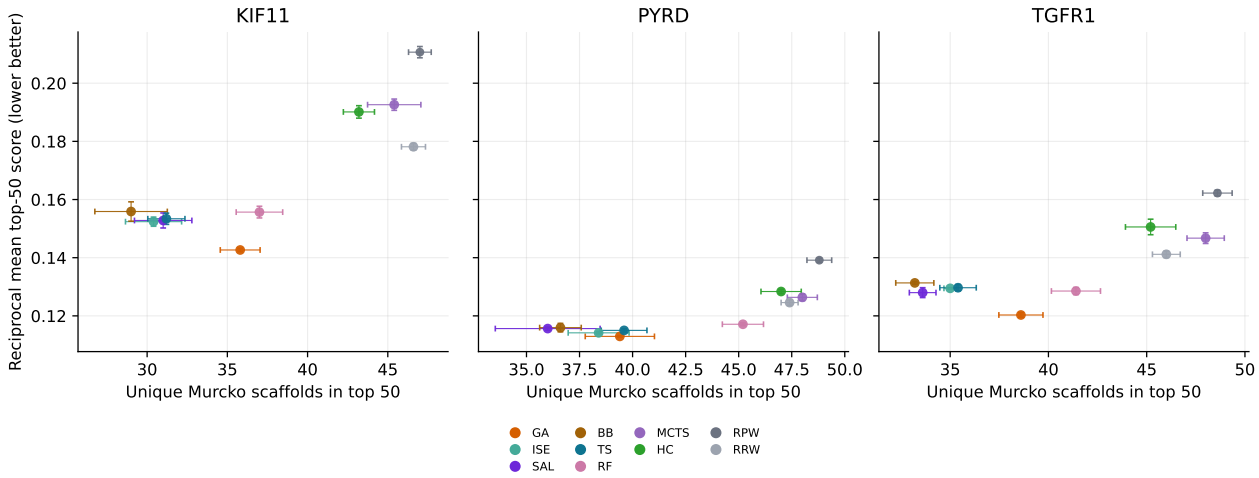


Figure 9. Glide Docking method score/diversity variant with the number of unique Murcko scaffolds among the top 50 valid retrieved products on the x-axis. As in the other variants, lower reciprocal score is better and the shared legend gives method abbreviations and colors.

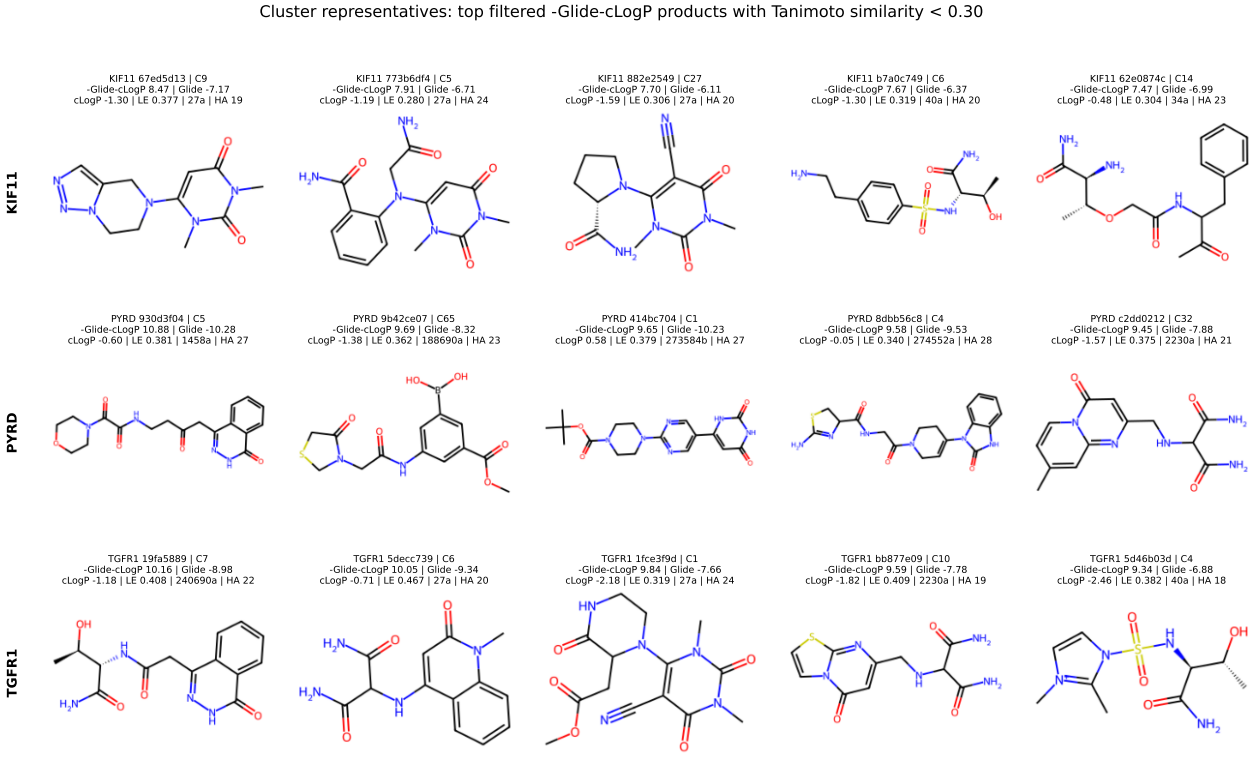


Figure 10. Top nonredundant cluster representatives for the filtered lipophilicity-penalized oracle at Morgan-fingerprint Tanimoto similarity cutoff 0.30. These panels make the diversity-constrained oracle fronts in Figure 6 chemically inspectable rather than only numeric.

A.8. Baseline Implementation Record

This section separates paper baselines from upstream systems that inspired them. A row enters the main leaderboard only if it is run through the common SynthonBench task API with the same submitted-call, duplicate-query, and product-id audit rules. Exact upstream-system reproductions remain valuable future work, but they are not silently substituted for in-house analogues.

SpaceHASTEN. The public SpaceHASTEN repository snapshot inspected for this paper is v0.10/HEAD `758cf0` and depends on Glide, SpaceLight, FTrees, and Chemprop (Kalliokoski et al., 2025). The ISE row is the in-house analogue: it starts from random initial queries, periodically refits a product-fingerprint regressor, expands local synthon neighborhoods, and greedily queries predicted high-utility tuples. Separately, the SpaceHASTEN FTrees/SpaceLight backend comparison is a non-enumerated run over all reactions on KIF11, PYRD, and TGFR1 for 38/42 CoLibri-supported reactions at seed 0, so it is reported directly in the main Glide Docking table rather than in the enumerated-pool baseline table.

SpaceGA. SpaceGA (Moesgaard & Kongsted, 2024) is run from its official repository as an exact upstream-system comparison (not an in-house analogue): a graph genetic algorithm whose mutation operator is a SpaceLight similarity search over the same merged CoLibri space used for SH, so its proposals are library members and its docking calls are scored by the same score-table oracle as every other row. Two adapter changes were required, both behaviour-preserving on well-formed inputs: (i) the upstream package exposes no random-seed control, so we seed Python and NumPy at start-up for reproducibility; and (ii) its SpaceLight-result parser raises on the empty/malformed CSVs that SpaceLight occasionally emits for degenerate crossover queries, which aborts a whole generation under parallel search, so we guard that parse to skip the offending query (identical behaviour otherwise). We run the published defaults (`ECFP4`, crossover 0.2, population 100) under the shared 5×2000 budget (and budget-matched curricula for the low-budget points of the budget-response sweep); the result-name of each hit recovers the canonical product id, so the evaluated set is graded by the standard exact audit.

HASTEN. The audited HASTEN source is the public repository at local HEAD `c30291`; the original method docks selected batches, trains Chemprop, predicts the full compound pool, and docks top predictions (Kalliokoski, 2021; Yang et al., 2019; Sivula et al., 2023). The SynthonBench integration keeps the cycle structure but replaces live docking with a fixed-table $-\mathrm{Glide} - \mathrm{cLogP}$ lookup and, purely for speed and model parity with the other surrogate baselines, substitutes the upstream Chemprop surrogate with an XGBoost regressor over precomputed enumerated 1M fingerprints. It runs five cycles and 2,000 charged calls per cycle on KIF11/PYRD/TGFR1, and is reported as the enumerate-then-learn HASTEN baseline in Table 3 because it requires full-library enumeration before search.

MolPAL. The MolPAL audit uses the public repository `github.com/coleygroup/molpal`, with publication tag `fc0b913` and audited HEAD `a40323` (Graff et al., 2021). Upstream MolPAL exposes CSV-SMILES pools, lookup/docking objectives, and batched active-learning loops; SynthonBench exports filtered lipophilicity-penalized lookup packages, product-id maps, and MolPAL-order HDF5 fingerprints. The reported MolPAL+XGB row uses 2,000 initial queries, four 2,000-query learned batches, and a 10K fixed-table budget on KIF11/PYRD/TGFR1. Like HASTEN, it is fair in charged oracle calls but relies on a fully enumerated pool, so it is kept as an enumerated-pool reference.

SALSA. SALSA is described as factored synthon active learning for non-enumerable combinatorial spaces (Grigg et al., 2025), but no public implementation was found in the primary-literature and repository audit. The code release retains a compatibility UCB sampler, while the reported SAL row implements the Algorithm 1 acquisition loop with empirical per-synthon mean/variance Thompson sampling. It is therefore an analysis-oriented in-house row, not an exact MPNN SALSA system reproduction.

ROCS X-inspired belief bandit. Public OpenEye/Cadence ROCS X material describes reaction-informed synthon libraries, OMEGA/ROCS/FastROCS-style scoring, and explicitly states the use of Bayesian-bandit active learning, but not the proprietary implementation, which is closed-source (OpenEye, Cadence Molecular Sciences, 2026; Geballe & Lee, 2026; Cadence Design Systems, 2025; OpenEye, Cadence Molecular Sciences, 2025). The ROCS-X-inspired Bandit (BB) row converts scalar SynthonBench feedback into an online ROCS-like belief assignment, updates synthon and pair Beta posteriors, and fuses sampled tuple evidence with a Hooper/Dempster-Shafer-style joint-belief rule following the ROCS belief-assignment lineage (Muchmore et al., 2008). It is reported as a in-house analogue built only from scalar SynthonBench feedback, not a ROCS X, ROCS, OMEGA, or FastROCS reproduction; we did not use the ROCS X product or code.

Thompson sampling. The Thompson sampling baseline follows the public Klarich/Walters implementation pattern from `github.com/PatWalters/TS`, audited at HEAD `6909c8` (Klarich et al., 2024). The reported TS row uses the warmup-prior reagent-space sampler through the

common SynthonBench task API; a simpler empirical-slot sampler is retained in the code release only, and an exact upstream-system comparison remains future work.

A.9. Baseline Algorithm Details and Hyperparameters

Table 7 records, for each baseline, a compact algorithm summary, the key hyperparameters used in every reported run, and the mechanism that makes each method search across reactions. All values are the code-release defaults; the full optimizer configurations and the exact allocator implementations live in the released registry (`baselines/registry.py` and `baselines/global_routing.py`). GA and the pooled RF search all 42 reactions directly, so they carry no allocator; the reagent-prior TS, SAL, and BB use the adaptive per-reaction Gaussian-Thompson allocator as an integral part of their bandit design; ISE uses the score-softmax allocator; and the generic per-reaction proposers MCTS and HC are routed by the static product-weighted allocator. The enumerate-then-learn HASTEN reference is included as the last row for completeness (it searches a fully enumerated pool, so it carries no synthon-space allocator); HASTEN and the MolPAL reference are documented in full in Section A.8 and Table 3, and the licensed SpaceHASTEN FTrees/SpaceLight backend in Section A.10.

Table 7. Per-method algorithm summary, key hyperparameters (code-release defaults used in all reported runs), and the reaction-allocation mechanism for each SynthonBench baseline. Allocator definitions are given in the footnote. "ECFP4" denotes Morgan radius-2 fingerprints; "slot-hash" denotes additional hashed synthon-index features.

Method	Algorithm	Key hyperparameters	Reaction mechanism
GA	Elitist typed GA: crossover, per-slot + reaction mutation, random immigrants, unseen-child retries	pop 96; mutate 0.35; reaction-mutate 0.15; immigrants 0.10; elite top 50%; child retries 96	all reactions (no allocator)
SpaceGA	Official graph GB-GA (Moesgaard & Kongsted, 2024); mutation replaced by a SpaceLight analog search, so offspring are library members	pop 100; children 20 (gen size 2,000); crossover 0.2; `ECFP4`; SpaceLight max-results 2,000; 5 generations	SpaceLight (no allocator)
ISE	Random init → periodic ExtraTrees refit on product ECFP4 → mutate elites to expand neighborhoods → greedy query	warmup 32; refit every 64; pool 512; 16 docked + 16 predicted elites; random frac. 0.25; 64 trees; ECFP4 r2/512b + 128 slot-hash	score-softmax allocator
SAL	Factored per-synthon empirical mean/variance; Gaussian Thompson per slot, one synthon per slot	init random 32; batch 64; prior $\mathcal{N}(0,1)$; $\epsilon = 0$	reaction-Thompson allocator
BB	Per-synthon + per-pair Beta posteriors; ROCS belief curve; Hooper fusion $1 - \prod_i (1 - b_i)$	init random 64; pool 256; $\mathrm{Beta}(1,9)$; $F_{\max}$ 0.46, SC_{50} 0.86, slope 6.8; slot/pair reliab. 1.0/0.5; novelty 0.015	reaction-Thompson allocator
TS	Reagent-prior warmup → shared normal–normal Gaussian posterior update per synthon	3 warmup trials/synthon; posterior retries 32	reaction-Thompson allocator
RF	Random forest over Blake2b-hashed reaction/synthon-indicator features; greedy over sampled pool	warmup 128; refit every 1,000; pool 64; 16 trees; feature dim 128	all reactions (no allocator)
MCTS	UCT over slot assignments with progressive widening and stochastic rollouts	exploration 1.4; widening $c = 2.0$, $\alpha = 0.5$; rollout randomness 0.10	product-weighted allocator
HC	GuacaMol-style: rank-weighted elite parent, per-slot mutation, restart on stall	elite 16; mutation rate 1.0; restart patience 64	product-weighted allocator
RPW	Uniform-random valid tuples	—	product-weighted allocator
RRW	Uniform-random valid tuples	—	reaction-uniform allocator
HASTEN‡	Enumerate-then-learn active learning (Kalliokoski, 2021; Sivula et al., 2023): dock a batch, refit a surrogate on enumerated-pool fingerprints, predict the full 1M pool, dock the top predictions; iterate	5 cycles × 2,000 calls (10K budget); XGBoost surrogate (128 trees, depth 4) over precomputed enumerated 1M RDKit 2048-bit fingerprints	enumerated pool (no allocator)

Allocators (one decision per charged call, given shared budget B across reactions): **reaction-Thompson** samples each reaction's empirical-reward Gaussian posterior (scale $\max(\hat{\sigma}/\sqrt{n+1}, 10^{-6})$, prior std 1) and queries the argmax; **score-softmax** draws a reaction from $(1-\epsilon)\,\mathrm{softmax}(\bar{r}/T) + \epsilon\,\mathrm{uniform}$ with $T = 0.25$, $\epsilon = 0.05$; **product-weighted** and **reaction-uniform** are static multinomials over reactions, with weights proportional to product-space size or uniform, respectively. Adaptive allocators take one warmup query per reaction. ‡HASTEN is the enumerate-then-learn reference (last row): it searches a fully enumerated 1M product pool rather than the non-enumerated synthon space, so it has no reaction allocator and is excluded from the main non-enumerated leaderboard; see Section A.8 and Table 3.

A.10. Licensed Backend Software and CoLibri Space Construction

The non-enumerated FTrees/SpaceLight SpaceHASTEN backend (the SH row in Table 2, plus the SHC ablation reported below) relies on the licensed BioSolveIT toolchain, run as command-line tools on Linux x64. For exact reproducibility we record the versions used in all experiments reported here: the CoLibri 9.1 fragment-space construction suite—ReactionSynthesizer 6.1.0, FragspaceMerger 4.1.0, SpaceLightDBCreator 2.1.0 and FragspaceConverter 1.1.0—together with the search tools SpaceLight 2.1.0 and FTrees 7.1.0 and the infiniSee 7.1.1 ("Arke") chemical-space navigation platform (BioSolveIT GmbH, 2026). Candidate retrieval during search uses SpaceLight (fingerprint analog search; default `fCSFP4`) and FTrees (pharmacophore feature-tree search); both read the same `.space` files, and each search is invoked with a bounded thread count and a per-call wall-clock timeout.

CoLibri fragment spaces are built directly from the synthon tables via the synthon/SMIRKS route rather than by enumerating products. For each reaction we emit a per-reaction reaction definition (SMIRKS/RXN) and the per-position synthon reagent files, generate one classical single-reaction frag-

ment space with `reaction_synthesizer -g`, merge the single spaces with `fragspace_merger`, build the matching topological fragment-space database step-wise with `spacelight_db_creator`, and finally produce the searchable `.space` (classical + topological) with `fragspace_converter`. SynthonBench marks synthon connection points with the actinide pseudo-atoms `[U]` and `[Np]`, which CoLibri's substructure matcher does not clip; we therefore remap them to placeholder elements (`[U]` → `[Te]`, `[Np]` → `[Sb]`) identically in the reaction definition and the reagent SMILES using RDKit (Landrum, 2025). The remap is connectivity-preserving and the placeholders are consumed when fragments are joined, so retrieved analogs are clean real molecules with no residual placeholder atoms; the two placeholders never collide with input atoms (verified: zero occurrences in the synthon SMILES). This conversion yields the 38/42 CoLibri-supported reactions reported in the main Glide Docking comparison.

Backend search cost and surrogate-scale coverage. The non-enumerated FTrees/SpaceLight backend is markedly slower per charged call than the in-process surrogates, because every cycle issues many short-lived licensed-backend search subprocesses (one retrieval per reaction per query probe across the 38 supported per-reaction spaces). Each SH (FTrees/SpaceLight + XGB) run—one target, five cycles of up to 2,000 charged calls—takes roughly 20–35 min at the 1M surrogate scale on the shared machine used here, and several-fold longer per run at 10M and 100M (the larger per-reaction spaces raise the cost of each `spacelight/ftrees` query, and the 100M docking-surrogate score table alone is ∼100M rows). SH is searched on the merged CoLibri space and reported across all four runs in Table 2: a five-seed Glide leaderboard and three-seed docking-surrogate sweeps at 1M/10M/100M (the surrogate sweeps run at reduced, background-priority parallelism to avoid disrupting concurrent jobs).

No-surrogate ablation (SHC). SpaceHASTEN's official protocol always pairs FTrees/SpaceLight retrieval with a ranking model, so we report the surrogate-free variant (SHC) only as an ablation rather than in the main leaderboard. Ablating the cycle-local XGB surrogate is costly: on the medchem-filtered Glide leaderboard the backend falls from 13.68 (SH) to 5.16 mean top-1000 DCRF—a 2.7× loss—and on the docking-surrogate scales from 17.1/109.1/163.3 (SH) to 7.4/7.6/14.2 (SHC) at 1M/10M/100M. The surrogate ranking, not FTrees/SpaceLight retrieval alone, therefore drives SpaceHASTEN's elite-recall efficiency.

A.11. Surrogate-Oracle Source Data and Validation

The docking-surrogate tasks use SpaceHASTEN-derived LightGBM (Ke et al., 2017) prediction tables, not measured docking at benchmark time. The source scores used for surrogate training were produced by converting SMILES to 3D structures with Phase/LigPrep and docking with Glide from Schrödinger Suite 2023-4, following the SpaceHASTEN workflow (Kalliokoski et al., 2025; Friesner et al., 2004). Training uses product ECFP4 fingerprints computed from SMILES with radius 2 and 2048 bits, stored as packed bytes and unpacked for LightGBM. The final model package uses LightGBM GPU regression with up to 4000 boosting rounds and early stopping, and applies 3× sample weight to the lowest 10% docking scores in the training split to prioritize tail fitting, since downstream search cares about the most negative scores. The split is random with seed 0: 70% train, 15% validation, and 15% test.

Table 8 reports the held-out metrics used to justify treating these models as structured benchmark oracles. Rank correlations are moderate to good, but elite-tail recall is imperfect. This is why the paper frames the docking-surrogate tracks as surrogate-oracle search tasks and not as evidence of biological activity.

A.12. Single-Reaction Glide Docking and External Baseline Context

Beyond the reported protocols, earlier single-reaction Glide Docking fixed-table runs, raw-score Glide Docking runs, and external-baseline checks use different reporting contracts from Table 4: some allocate separate budgets per reaction, some optimize raw Glide score rather than $-\mathrm{Glide} - \mathrm{cLogP}$, and these external baselines have implementation-specific coverage limits. They are therefore reported as supplementary context rather than mixed into the paper figures and leaderboards. The primary Glide Docking result is the shared-budget medchem-filtered lipophilicity-penalized benchmark and its exact product-id audit.

Table 8. Held-out LightGBM surrogate validation. Lower raw docking scores are better; tail recall is computed for the most negative true scores.

Target	Input rows	Used rows	Test n	RMSE	MAE	R^2	Pearson	Spearman	Lowest 0.1%	Lowest 1%
KIF11	853,539	853,505	128,026	0.642	0.503	0.537	0.734	0.712	0.279	0.383
PYRD	992,001	991,545	148,732	0.734	0.541	0.619	0.788	0.794	0.154	0.281
TGFR1	996,030	995,897	149,385	0.707	0.542	0.518	0.723	0.713	0.153	0.244

A.13. Non-Docking Suites: Synthetic Properties, Lead Rediscovery, and Lead Optimization

The release also includes three matched non-docking suites—Synthetic Properties, Lead Rediscovery, and Lead Optimization—over the same 42-reaction 1M API and one 10,000-call budget per task. The completed matrix covers ten non-docking objectives: additive, pairwise, sparse3, and constrained synthetic structure; ECFP4 and FCFP4 aripiprazole similarity; thresholded lead optimization; Murcko-scaffold-hop similarity; a stricter continuous Druglike MPO; and a Lead+MPO composite that couples aripiprazole similarity with druglike descriptors. The completed appendix matrix uses the same ten in-house all-reactions methods as the main Glide Docking table (Table 2), including the SALSA-inspired tabular Thompson analysis row. These results are appendix evidence for breadth, not a replacement for the measured-score docking result. We additionally audit exact anchors by scoring all 990,829 enumerated products for each non-docking objective and estimating the 10,000-call random best-utility anchor with 2,000 Monte Carlo random draws. Score/diversity product/scaffold-overlap analyses are included as portfolio supplements; an ten-task scaffold-robustness audit measures exact-best-scaffold concentration, off-exact-best-scaffold utility closure, and top-50 scaffold breadth; and molecule examples are exported for the chemically interpretable non-docking tasks. Prospective task-to-target validation remains follow-up work.

Table 9 and Figure 11 summarize the completed matrix. The typed GA is best or tied by mean best utility on 8/10 tasks, holds the best mean within-task rank (1.92) as the unique significance group **a**, and closes 0.761 of the exact best-utility gap between Monte Carlo random and the full-universe optimum. The pooled RF is the next-best exact-gap method (0.397) and the only other group **b** row. The bandit-family rows—Thompson Sampling, the ROCS-X-inspired bandit, and the SALSA-inspired tabular Thompson—remain negative controls under exact best-gap closure (-0.170, -0.189, and -0.227) even though they post competitive Top-1000 DCRF (12.0, 13.1, and 11.8): they concentrate queries on structured or tied objectives without closing the exact gap on average, so they are analysis rows rather than upstream-system wins. The SALSA-inspired row in particular is an analysis-oriented tabular reimplementation, not upstream SALSA performance. The metaheuristic rows (MCTS and hill climbing) sit mid-table with small positive gap closure ($\approx$0.25), while the two random baselines trail in significance groups **d/e**. The stricter Druglike MPO is a non-saturated continuous property objective: its exact optimum is 0.9848 with a single exact-best product, the Monte Carlo random-best anchor is 0.9159, and the typed GA closes 0.767 of that exact gap with Top-1000 DCRF 15.46. The score/diversity portfolio shows that the non-docking suites reward very different operating points: the typed GA is close to exact on additive and constraint objectives but often concentrates chemically, whereas ISE-style methods retain broader top-50 Tanimoto diversity on lead and sparse objectives. The exported molecule-example table makes the lead/property tasks concrete: several methods recover the exact lead-similarity optimum, while Druglike MPO gives the harder property panel. The Lead+MPO task adds a non-saturated multi-objective lead-optimization panel: its exact optimum is 0.8045, the Monte Carlo random-best anchor is 0.6806, pooled RF closes 0.941 of the exact gap, the typed GA closes 0.907, and the ROCS-X-inspired bandit closes 0.861. The scaffold-disjoint supplementary CSVs add a second chemistry check across all ten non-docking objectives. Exact-best-scaffold concentration is usually low, so most top-50 sets are not simple scaffold collapse; feature FCFP4 is the strongest concentration case, reaching 0.17 exact-best-scaffold share for the typed GA. Off-exact-best-scaffold closure is more discriminating: the typed GA keeps closure near 0.59 on the lead/scaffold-hop objectives, pooled RF is strongest on Druglike MPO (0.75) and Lead+MPO (0.81), and the constraint task exposes several policies whose top-50 products remain far below the exact/random anchor even when chemically broad. The run-level CSVs still keep aripiprazole seed-scaffold flags, where the seed Murcko scaffold is absent from every method's top-50 selections. Earlier cross-suite transfer supplements remain available for documentation, but they are not used for the main paper summary because the Glide Docking benchmark now uses the medchem-filtered lipophilicity-penalized utility throughout.

A.14. Statistical and Supplementary Files

The reproducibility package includes statistical supplementary tables for the main exact-audited suites: method summaries, reaction-level bootstrap confidence intervals, product-weighted summaries, size-stratified summaries,

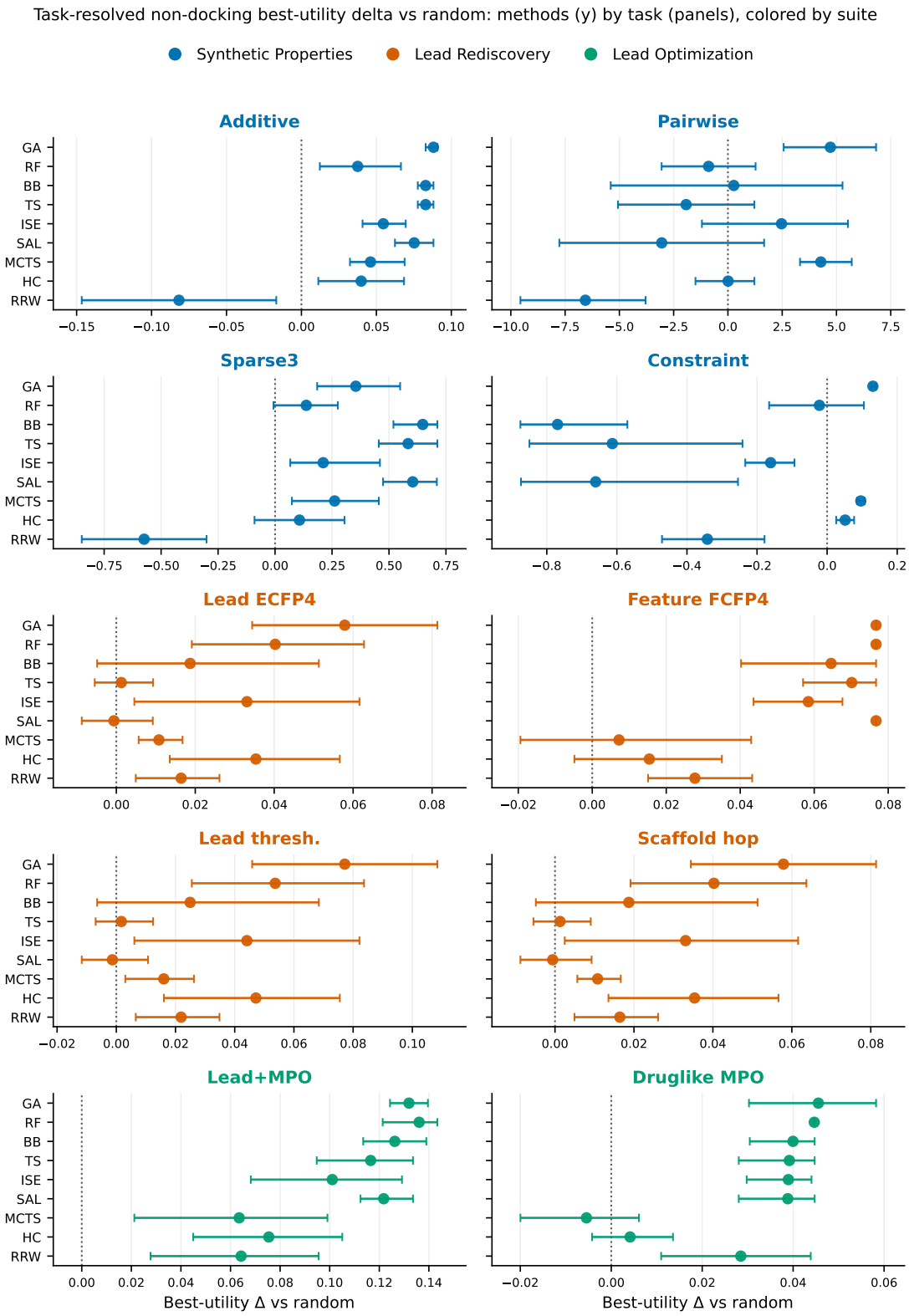


Figure 11. Task-resolved non-docking best-utility deltas versus random—the non-docking analogue of the target-resolved docking view in Figure 2. Each panel is one applied task (the analogue of a docking target); methods are on the y-axis sorted by overall mean within-task rank (strongest on top, the same order in every panel), and points are means with 95% bootstrap confidence intervals over seeds. The metric is the best-utility delta versus the product-weighted random anchor, so the anchor (RPW) is the baseline and is not drawn; the dotted line marks the random baseline (delta $= 0$). Each panel keeps its own x-axis because the non-docking objectives live on different utility scales (the faceting makes scale-sensitive failures visible—strong gains on the pairwise and sparse synthetic objectives but negative deltas on the constraint objective for several policies). Points are colored by suite (Synthetic Properties, Lead Rediscovery, Lead Optimization), reusing the target palette of Figure 2. The applied suite is run for the ten in-house methods of the main Glide Docking table (Table 2); the external SpaceGA, SpaceHASTEN, and HASTEN references are docking-only and do not apply to these abstract objectives. Method abbreviations follow Section 3. The typed GA is positive on every task and leads or ties the field on most.

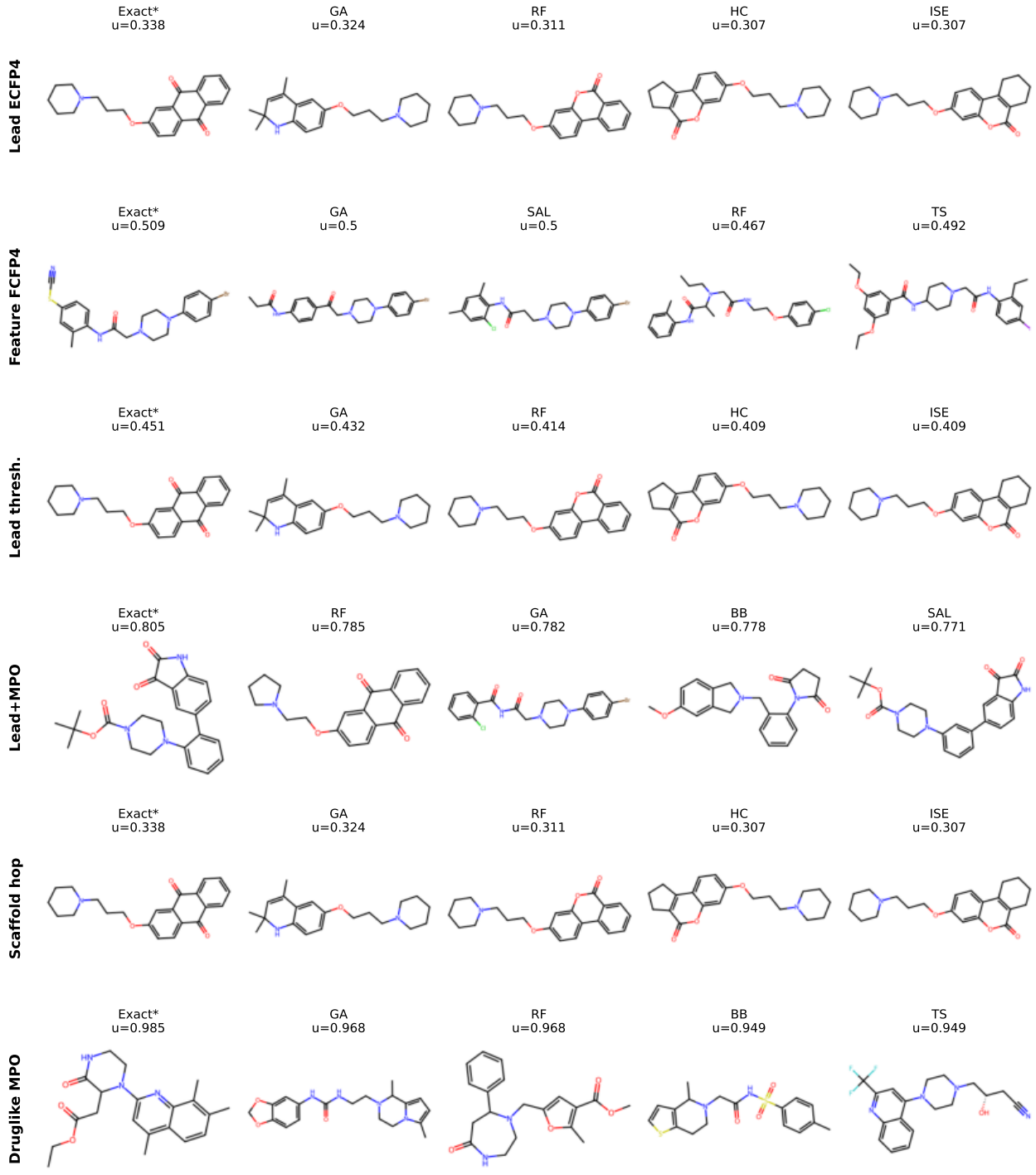


Figure 16. Non-docking-suite molecule examples from exact optima and retrieved candidates. Rows compare objective-specific exact best products with molecules recovered by high-performing methods, providing a structural readout for the non-docking objectives behind the aggregate rank and exact-anchor summaries.

Table 9. Non-docking suites (1M): consolidated per-method summary over the ten Synthetic-Properties / Lead-Rediscovery / Lead-Optimization objectives and a shared 10,000-call budget per task, using the same ten in-house methods as the main Glide Docking table (Table 2). *Best/tied* counts tasks where a method matches the best mean best-utility value; *Rank* is mean within-task rank (lower is better); *Gap clos.* is the exact best-utility gap closure $(u_{\mathrm{method}} - u_{\mathrm{random}})/(u_{\mathrm{exact}} - u_{\mathrm{random}})$ averaged over tasks after scoring the full 990,829-product universe (values below zero underperform the Monte Carlo random-best anchor on average); *DCRF* is the exact Top-1000 call-reduction factor (smoother where elite hits are sparse); *Exact hit* is the rate of recovering a full-universe exact-best product. *Sig.* gives compact-letter significance groups from Holm-corrected paired t-tests on per-(task,seed) within-task rank (methods sharing a letter are not significantly different, $\alpha = 0.05$); the typed GA is the unique group-**a** method. Bold marks the best value in each column. Method names follow Section 3.

Method	Family	Sig.	Best/tied	Rank	Gap clos.	DCRF	Exact hit
Genetic Algorithm	Evolutionary	a	**8**	**1.92**	**0.761**	**21.9**	**0.56**
Random Forest	Active learning	b	2	4.02	0.397	8.7	0.33
ROCS-X-inspired Bandit	Bandit	bc	1	4.34	-0.189	13.1	0.31
Thompson Sampling	Bandit	bcd	0	4.94	-0.170	12.0	0.22
Iterative Surrogate Expansion	Active learning	bc	0	5.02	0.273	5.5	0.18
SALSA-inspired Thompson	Bandit	bcd	1	5.06	-0.227	11.8	0.25
Monte Carlo Tree Search	Metaheuristic	bcd	0	5.84	0.251	2.1	0.15
Hill Climb	Metaheuristic	cd	0	5.88	0.252	2.4	0.09
Reaction-weighted random	Random	de	0	7.08	-0.433	1.7	0.09
Product-weighted random	Random	e	0	7.62	-0.079	1.0	0.09

and paired differences. The main paper cites only the highest-signal values, but the CSV files are included so readers can inspect reaction-level heterogeneity and product-size weighting directly. Earlier single-reaction size/phase/locality analyses, fixed-1000 ablations, and Pareto-GA selectivity experiments are retained as CSV supplementary tables for compatibility and future analysis, but they are intentionally not main-paper evidence for the redesigned benchmark. The older external-baseline tables are likewise not main-paper evidence because they do not share the primary filtered Glide Docking utility.

The older per-reaction 100M appendix audit includes two 126-task grids: docking, with all 42 reactions crossed with the three docking-surrogate targets, and synthetic scale, with additive, pairwise, and sparse3 objectives for the same 42 reactions. Docking is included as compact aggregate, exact-audit, statistics, runtime, accounting-audit, and seed-coverage supplements; the accounting audit confirms that random and every other listed method have complete rows for all 126 tasks. The recommended compact visual scale summary reports top-0.1% DCRF with runtime/memory context and labels the rows as a matched 100M audit. Synthetic is included as compact aggregate, task, runtime, and coverage supplements rather than raw query-log files. The three-target Glide Docking track adds compact fixed-table measured-score summaries, medchem-filtered lipophilicity-penalized exact tables, reaction-bootstrap statistics, multi-reaction Glide task generation, run manifests, target-resolved DCRF summaries, surrogate-to-real transfer tables, and score/diversity Pareto summaries; raw Glide scores are treated as optional large files and raw-score checks remain supplementary files.

A.15. Reproducibility Package Contents

The reproducibility package is designed for inspection and quick execution, not for distributing the full raw run tree. It contains the paper PDF and source, a code archive, exact and aggregate tables, figure data files, metadata cards, run manifests, command logs, and checksum manifests. The compact package intentionally excludes tens of gigabytes of raw per-task query records and large score tables. Package-level checksums are included so readers can verify consistency of the exported files. The reproducibility package accompanies the public release.

For the public release, the code release carries the compact reproducibility surface, including the files covered by the code checksum manifest. A separate result-only archive can mirror these compact files if the live form asks for a dataset URL, but the GenBio submission path does not rely on a second download. The code repository is `https://github.com/mireklzicar/synthonbench`.

The largest excluded resources are raw query records and full score tables. The reproducibility package includes compact summaries and checksum prefixes for these resources, so readers can verify regeneration without downloading the full raw run tree.