# Minibinder Lab: The Reliability Gap Of Agents For Designing High Quality Protein Binders

**Anda-Raluca Epure** [1] **Deepa Mal Korani** [2] **Jonathan Hedley** [1] **Maxim Secor** [2]
**Alejo Nevado-Holgado** [3] **Philip Torr** [1]

## Abstract

We introduce Minibinder Lab, an agentic framework for iterative design and optimisation of minibinders, small protein structures emerging as a key modality in drug discovery. While generative models enable rapid *in silico* candidate generation, selecting and improving binders remains expert-driven, requiring target-specific reasoning, structural inspection, interface scoring, sequence novelty assessment. Minibinder Lab systematises this process through 12 specialised agents coordinating generation, analysis, critique, refinement over design rounds, combining large language model (LLM) reasoning with established structural and sequence-level tools alongside target-specific literature evidence. The framework is evaluated on three therapeutically relevant targets, PD-L1, PDGFR, CD3, using two open- and two closed-source LLMs, showing that agents reason over multi-source protein design data and propose redesign strategies judged reasonable by two domain experts. However, systematic analysis of agent behaviour exposes a critical safety-relevant failure mode: while reasoning models successfully integrate literature evidence into design decisions, they cannot self-correct when selecting incorrect tool configurations, persisting with erroneous choices despite contradictory downstream outputs. In a high-stakes scientific domain where unchecked agent errors could propagate flawed candidates into experimental pipelines, this inability to recognise and recover from its own mistakes is a reliability boundary that must be addressed before autonomous deployment in the wild.

[1]Department of Engineering Science, University of Oxford, Oxford, United Kingdom [2]Molecular AI, Novo Nordisk, Lexington, MA [3]Department of Psychiatry, University of Oxford, Oxford, United Kingdom. Correspondence to: Anda-Raluca Epure <anda-raluca.epure@eng.ox.ac.uk>.

*Published at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026.* Copyright 2026 by the author(s).

## 1. Introduction

Despite rapid progress in protein generative modeling, the design of high-affinity protein minibinders remains an iterative, expert-driven workflow: candidates are generated, evaluated for structural and interface quality, and then manually tuned based on those signals. While generative models (e.g., RFdiffusion (Watson et al., 2023), AlphaProteo (Zambaldi et al., 2024), Boltz-2 (Passaro et al., 2025), La-Proteina (Geffner et al., 2025)) can now propose diverse binders, they do not remove the main bottleneck: interpreting noisy, often conflicting outputs from standard analysis tools (Rosetta (Leman et al., 2020), GetContacts (GetContacts), DSSP (Hekkelman et al., 2025)) and translating them into actionable design decisions. This process requires months of specialised work and often results in failure (Styles et al., 2025; Overath et al., 2025).

Multi-agent systems offer a natural architecture for this problem: decompose the workflow into specialised roles and let LLM-powered agents coordinate across them. Recent work has demonstrated the promise of this approach in nanobody design (Swanson et al., 2025), materials discovery (Liu et al., 2025), and scientific hypothesis generation (Gottweis et al., 2026). However, deploying such systems in open-ended scientific environments, where agents must reason over heterogeneous tool outputs, handle stochastic variation, and make consequential decisions across many iterations, raises questions about reliability that controlled benchmarks do not capture.

We introduce **Minibinder Lab**, a closed-loop framework of 12 specialised agents that automate minibinder generation, structural analysis, and iterative redesign. We evaluate the framework on three therapeutically relevant targets (PD-L1, PDGFR, CD3) using two open-weight and two closed-source LLMs, and report both design quality and a systematic analysis of agent behaviour across 20 iterations. Our contributions are:

- We propose a multi-agent framework for *in silico* minibinder discovery that integrates literature retrieval, structure generation, multi-tool analysis, and configuration optimisation, achieving structural metrics com-

petitive with a one-shot BoltzGen (Stark et al., 2025) baseline using a fraction of the design budget.

- We identify concrete reliability boundaries in agentic scientific reasoning. Domain expert evaluation of reasoning traces from the Oracle Agent reveals that agents successfully integrate multi-source evidence and produce literature-grounded diagnostic reasoning, but cannot self-correct when selecting incorrect tool configurations — persisting with erroneous choices even when downstream outputs contradict them. Agents also oscillate between design hypotheses without convergence criteria and do not systematically compare predicted effects against observed outcomes, exposing gaps in experimental planning discipline that must be addressed before autonomous deployment.

Broader impacts and dual-use considerations of our work are discussed in Appendix: Impact.

## 2. Related Work

**Protein binder design.** Recent generative models, including RFdiffusion (Watson et al., 2023), BoltzGen (Stark et al., 2025), AlphaProteo (Zambaldi et al., 2024), and La-Proteina (Geffner et al., 2025), can now propose diverse binder candidates *in silico*. However, these models address only the generation step; selecting, analysing, and iteratively improving candidates remains a manual, expert-driven process that constitutes the primary bottleneck in binder development (Styles et al., 2025).

**Multi-agent systems in science.** LLM-powered multi-agent frameworks have demonstrated promising results across scientific domains: Virtual Lab (Swanson et al., 2025) designed and validated SARS-CoV-2 nanobodies, ChatBattery (Liu et al., 2025) proposed experimentally validated cathode materials, Robin (Ghareeb et al., 2026) identified novel treatments for macular degeneration, and the Co-Scientist (Gottweis et al., 2026) discovered new epigenetic targets. In drug discovery specifically, ChemCrow (M. Bran et al., 2024) and ProteinCrow (Ponnapati et al., 2025) augment LLMs with specialised chemistry and protein-design tools. These systems report primarily on task success, designs produced, hypotheses validated, but do not systematically analyse agent failure modes, self-correction behaviour, or reliability across extended multi-iteration workflows. Our work addresses this gap: Minibinder Lab is, to our knowledge, the first multi-agent framework for iterative binder optimisation, and the first to report a structured analysis of where agentic reasoning succeeds and fails over 20 closed-loop design iterations, evaluated by domain experts.

## 3. Methods

### 3.1. Overview

The **Minibinder Lab** is a multi-agent framework for closed-loop *in silico* minibinder generation. The system consists of **12 specialised agents** that generate, analyse, summarise, and iteratively refine candidate minibinders, as shown in Figure 1. This choice is motivated by recent evidence that agent effectiveness depends on the alignment between coordination strategy and task structure (Kim et al., 2026); we therefore hypothesise that a set of simpler, specialised agents is better suited to this decomposable workflow than a single general-purpose agent, given how the task naturally decomposes into literature retrieval, structure generation, preprocessing, structural analysis, reporting, and configuration updates. Full descriptions of the agents are provided in Appendix A.

Each target run begins with a one-time target analysis step. The **Literature Expert** Agent constructs a biological brief from UniProt, RCSB PDB, and Europe PMC, recording accession identifiers, functional annotations, available structural information, reported interaction sites, and relevant literature context. This brief is cached and reused throughout the target run, avoiding repeated literature retrieval during iterative design. The agent's tools are described in Appendix B.2.1.

The main design loop then proceeds for a fixed number of iterations. At each iteration, the **Designer** Agent uses BoltzGen (Stark et al., 2025) to generate candidate minibinders from the current YAML configuration. The resulting structures are preprocessed by converting structural files to a common format, adding hydrogens, and applying restrained energy minimisation. Analysis agents then compute complementary descriptors of each candidate, including sequence novelty, backbone geometry, secondary structure, non-covalent contacts, and interface-quality metrics. Tool-level details for the **Designer, Preprocessor, Novelty, Distogram, Ramachandran, GetContacts, DSSP,** and **Rosetta** Agents are provided in Appendices B.2.2–B.2.9.

The **Reporter** Agent consolidates the design and analysis outputs into a structured iteration report. The **Oracle** Agent interprets this report together with the literature brief, novelty analysis, raw structural metrics, and cross-iteration history, then updates the configuration for the next design round. To keep the optimisation space controlled, the Oracle may edit only three parts of the BoltzGen configuration: the designed binder length range, target binding or non-binding residue constraints, and optional secondary-structure constraints. Finally, the **Orchestrator** manages configuration handoff and repeats the workflow. Reporter, Oracle, and Orchestrator details are provided in Appendices B.2.10, B.2.11, and B.2.12.

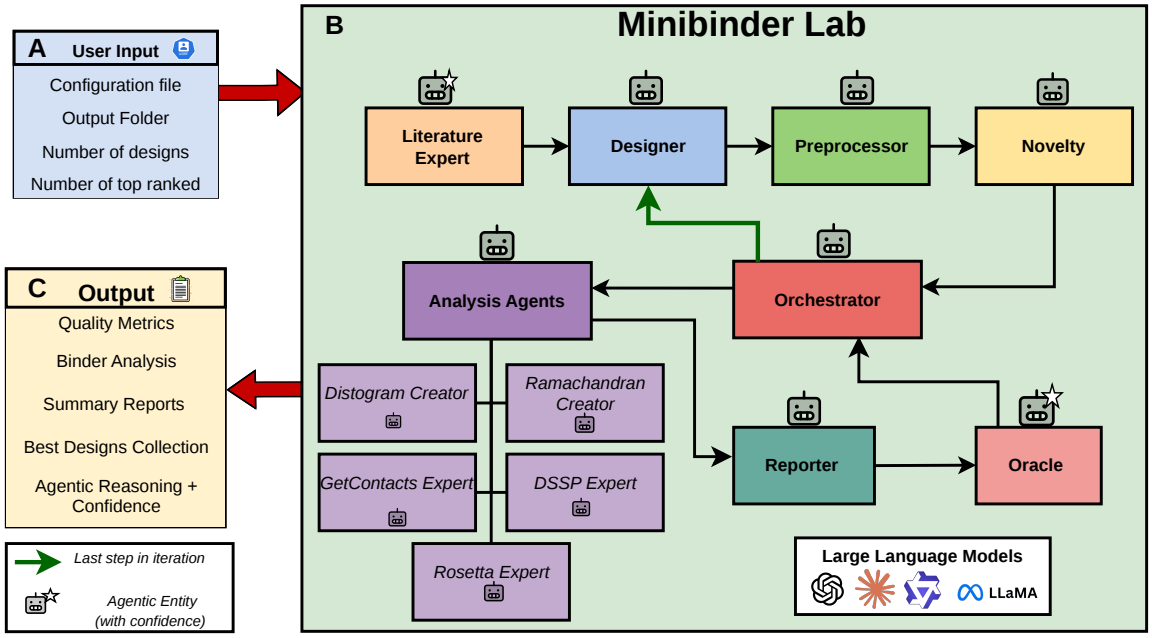



*Figure 1.* The **Minibinder Lab** Workflow diagram. **A.** The user provides a set of specifications for the design of minibinders for a specific protein target of pharmacological importance. **B.** The Minibinder Lab Scientist model consists of 12 agents, the *Literature Expert, Designer, Preprocessor, Novelty, Distogram Creator, Ramachandran Creator, GetContacts Expert, DSSP Expert, Rosetta Expert, Reporter, Oracle, Orchestrator*, which simulate a research environment, proposing minibinder structures, analysing and improving them until they reach the standards desired by the user. Once this is completed, the results are communicated back to the user. Note that the *Literature Expert Agent* runs only in the first iteration. **C.** The Minibinder Lab provides the user with multiple interpretable files: quality metrics, binder analysis files (e.g. distogram and Ramachandran plots, files resulting from GetContacts, DSSP, Rosetta), summary reports, a folder with the best designs from all iterations, and detailed agentic reports with confidence in the structure changes.

### 3.2. Agentic Implementation

**Skills.** The workflow is implemented using a skills-based prompting setup[1] (Xu & Yan, 2026), in which each agent is equipped with specialised instructions and access to the tools required for its role. This allows each agent to operate as a constrained domain-specific component rather than as a general-purpose language model. The full skills, prompts, and tool interfaces are provided in Appendix B.

Details regarding implementation and compute are given in Appendix D.3.

### 3.3. Evaluation

**Protocol.** To ensure comparability across experiments while keeping the computational budget tractable, we use a fixed evaluation protocol. We evaluate the framework on three molecular targets: PD-L1, PDGFR, and CD3 - further discussion is provided in Appendix C.1, with biological background on all three targets in Appendix C.2. For each target, Minibinder Lab is run for 20 closed-loop iterations. At each iteration, 100 candidate designs are generated, and the top 10 according to the BoltzGen filtering and ranking logic are analysed, with a subset added to a cumulative best-binder pool, which then undertakes a representative-clustering filtering, to ensure diversity (detailed in Appendix D.2). Reported metrics at each iteration are then computed using the top 10 designs for that metric from this pool. The same agent workflow, analysis suite and iteration budget are used for all targets. The large language models (LLMs) used to instantiate the agentic components are reported in Appendix D.1. We make explicit our use of LLMs throughout this work in Appendix: LLM Usage.

**Metrics.** At each iteration, we evaluate generated designs using both model-derived confidence metrics and structure-based analysis metrics. From BoltzGen (Stark et al., 2025), we record complex-quality and confidence metrics including ipTM, pTM, PAE, RMSD-based filters, and the number of structures passing the BoltzGen filters. Downstream analysis adds sequence-novelty estimates from MMseqs (Hauser et al., 2016; Steinegger & Söding, 2017), backbone-quality checks from Ramachandran analysis and DSSP (Hekkelman

[1] https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview

et al., 2025), interface contact fingerprints from GetContacts (GetContacts), and Rosetta (Leman et al., 2020)-derived interface metrics such as interface energy, buried surface area, shape complementarity, hydrogen bonds, unsatisfied polar contacts. Metric definitions are provided in Appendix D.4.

**Baseline.** For each target, we compare against a one-shot BoltzGen (Stark et al., 2025) run using the same initial configuration file provided to Minibinder Lab. BoltzGen generates 2000 binders, from which the top 200 are retained using its standard filtering logic to match the total Minibinder Lab design budget. For each metric, we report the average over the top 10 retained designs.

**Domain Expert Evaluation.** To assess the interpretability and scientific quality of the Oracle Agent's reasoning, we asked two domain specialists to review 14 reasoning traces from GPT-5.4-mini-powered runs, seven each for PD-L1 and PDGFR targets. The reviewers consisted of one protein-design specialist and one chemical physicist with expertise in molecular modelling and physical analysis. Reviewers were asked to assess whether the Oracle's proposed configuration changes were supported by the available evidence, the reasoning appropriately integrated outputs from the analysis agents, uncertainty and stochastic variation were handled cautiously, and whether tool failures or missing data were acknowledged. The resulting qualitative feedback is summarised in Sections 4.2-4.4.

## 4. Results and Discussion

### 4.1. Iterative refinement achieves competitive structural performance.

Across all three targets, by 20 iterations, Minibinder Lab approaches or exceeds the one-shot BoltzGen (Stark et al., 2025) baseline on individual metrics. As shown in Table 1 for PDGFR, ipTM, pTM, and Boltz (Passaro et al., 2025) confidence score generally increase, while PAE and RMSD decrease. Similar trends are observed in Table 2 and Table 3 for the PD-L1 and CD3 targets, respectively. Convergence trajectories for all targets are shown in Figures 7, 8, and 9 in Appendix E. It should be noted that our comparison applies fairness principles: after 20 iterations, the agent has generated as many designs as the BoltzGen baseline.

The best-performing model varies by target. For PD-L1 and PDGFR, QWEN-3.5-27B outperforms closed-source models on structural metrics, suggesting that a relatively small open-weight reasoning model can support effective configuration planning for well-characterised targets when constrained by specialised tools. For CD3, the most challenging target, GPT-5.4 MINI and CLAUDE SONNET 4.6 performs best, possibly reflecting the value of stronger reasoning capabilities when literature evidence and structural constraints are harder to integrate. For PDGFR LLAMA-3.1-8B-INSTRUCT achieves strong numerical results (second best performing model for the structural metrics), despite being prone to hallucinated reasoning across runs (see Section 4.4) - the competitive numbers obtained are not due to agentic reasoning and refinement, but due to BoltzGen (Stark et al., 2025) structural design generative variance, together with cumulative-best-pool selection effect. This is interesting reliability limitation, since it highlights that the LLAMA-3.1-8B-INSTRUCT **fails to act as an agent**.

These results establish that the multi-agent workflow produces designs of sufficient quality to make the behavioural analysis in the following sections meaningful: we are studying agents that can improve binder quality, not agents that fail at the task entirely.

### 4.2. Agents produce literature-grounded, multi-source reasoning.

Two domain experts reviewed 14 Oracle reasoning traces from GPT-5.4 MINI runs (seven each for PD-L1 and PDGFR protein targets). Both experts agreed that the Oracle consistently anchors its decisions to the literature brief produced by the Literature Expert Agent, linking proposed configuration changes to known epitopes, functional regions, and previously reported binding interfaces rather than justifying changes solely from aggregate structural scores. Incorporating literature context into protein design pipelines is a promising direction, as also suggested by Latent Labs Team et al. (2026).

Beyond literature grounding, experts highlighted the system's ability to **synthesise evidence from multiple independent analyses** before proposing configuration updates - an example of a complete reasoning trace can be found in Appendix G.3. Each analysis agent captures a different aspect of candidate quality: GetContacts for interface residue contacts, Rosetta for energetics, DSSP for secondary structure, MMseqs for sequence novelty, and structural sanity checks from distogram and Ramachandran analyses. The Oracle does not appear to trust any single sub-agent in isolation; instead, it weighs partially independent evidence streams before making a configuration change. This behaviour is particularly useful when candidates score well on one metric while failing on another.

The experts also identified instances of **diagnostic reasoning**. For PDGFR, the Oracle repeatedly diagnosed the core problem as "interface registration" — poor epitope targeting — and systematically ruled out alternative hypotheses (backbone collapse, backbone strain, incorrect fold topology, binding to non-productive regions) before concluding that target-region selection, rather than global structural quality, was the limiting factor. This ability to make deductions from heterogeneous protein analysis data and literature evidence is encouraging for future binder-design workflows.

*Table 1.* **Minibinder Lab results (PDGFR).** Pass rate (%) and structural/sequence quality metrics across models and iteration budgets ($N \in \{1, 10, 20\}$). All experiments are run 3 times with different initalisation (random seed available for open-source models only). Best results per metric in **bold**, when comparing averages. The **R** column marks reasoning models (✓). Metrics: *ipTM* (interface predicted TM-score), *PAE* (predicted aligned error, Å), *pTM* (predicted TM-score), *RMSD* (Å, vs. reference), *Conf.* (Boltz confidence score), *pIdent* (% sequence identity to known binders), *Cov.* (BLAST query coverage, %). Higher is better except for PAE, RMSD, pIdent and coverage (lower is better, marked ↓). Full metric definitions are reported in Appendix D.4. Full definitions of the models used are present in Appendix D.1.

| Model | $N$ | R | ipTM (↑) | pTM (↑) | PAE (↓) | RMSD (↓) | Conf. (↑) | pIdent (↓) | Cov. (↓) |
|---|---|---|---|---|---|---|---|---|---|
| *Generalist Reasoning Models* | | | | | | | | | |
| GPT-5.4 MINI | 1 | ✓ | $0.5007 \pm 0.0133$ | $0.7854 \pm 0.0064$ | $4.6933 \pm 0.2841$ | $1.5476 \pm 0.1392$ | $0.6256 \pm 0.0075$ | $40.1639 \pm 2.3673$ | $0.8923 \pm 0.0173$ |
| GPT-5.4 MINI | 10 | ✓ | $0.5625 \pm 0.0125$ | $0.8058 \pm 0.0019$ | $3.6275 \pm 0.0799$ | $1.0406 \pm 0.0830$ | $0.6467 \pm 0.0057$ | $39.1344 \pm 0.3477$ | $0.8589 \pm 0.0012$ |
| GPT-5.4 MINI | 20 | ✓ | $0.5812 \pm 0.0100$ | $0.8116 \pm 0.0015$ | $3.2556 \pm 0.0707$ | $0.9745 \pm 0.0648$ | $0.6541 \pm 0.0044$ | $39.0850 \pm 0.2677$ | $\mathbf{0.8590 \pm 0.0087}$ |
| CLAUDE-SONNET 4.6 | 1 | ✓ | $0.4768 \pm 0.0242$ | $0.7750 \pm 0.0031$ | $5.2402 \pm 0.6358$ | $1.6018 \pm 0.2444$ | $0.6218 \pm 0.0093$ | $40.4405 \pm 3.2953$ | $0.8606 \pm 0.0533$ |
| CLAUDE-SONNET 4.6 | 10 | ✓ | $0.5711 \pm 0.0166$ | $0.8039 \pm 0.0003$ | $3.4874 \pm 0.2263$ | $1.0315 \pm 0.0208$ | $0.6496 \pm 0.0015$ | $39.8445 \pm 1.4342$ | $0.8647 \pm 0.0113$ |
| CLAUDE-SONNET 4.6 | 20 | ✓ | $0.5861 \pm 0.0151$ | $0.8103 \pm 0.0014$ | $3.2656 \pm 0.1681$ | $0.9666 \pm 0.0246$ | $0.6570 \pm 0.0025$ | $39.4830 \pm 0.4137$ | $0.8689 \pm 0.0077$ |
| *Open-Weight Baselines* | | | | | | | | | |
| LLAMA-3.1-8B-INST | 1 | — | $0.4911 \pm 0.0495$ | $0.7794 \pm 0.0126$ | $4.9351 \pm 0.9009$ | $1.6052 \pm 0.0692$ | $0.6256 \pm 0.0225$ | $29.1722 \pm 25.8789^{*}$ | $0.5824 \pm 0.5071^{*}$ |
| LLAMA-3.1-8B-INST | 10 | — | $0.5804 \pm 0.0344$ | $0.8102 \pm 0.0046$ | $3.3564 \pm 0.4469$ | $0.9760 \pm 0.0254$ | $0.6570 \pm 0.0137$ | $46.7062 \pm 11.9747$ | $0.8968 \pm 0.0339$ |
| LLAMA-3.1-8B-INST | 20 | — | $0.5957 \pm 0.0266$ | $0.8140 \pm 0.0051$ | $3.1040 \pm 0.2787$ | $0.9331 \pm 0.0243$ | $0.6634 \pm 0.0102$ | $47.1765 \pm 12.8940$ | $0.8958 \pm 0.0447$ |
| QWEN-3.5-27B | 1 | ✓ | $0.5339 \pm 0.0247$ | $0.7916 \pm 0.0050$ | $3.9513 \pm 0.2778$ | $1.3634 \pm 0.0862$ | $0.6371 \pm 0.0106$ | $37.5000 \pm 2.4556$ | $f0.8555 \pm 0.1278$ |
| QWEN-3.5-27B | 10 | ✓ | $0.5976 \pm 0.0141$ | $0.8140 \pm 0.0106$ | $3.0960 \pm 0.1225$ | $0.9334 \pm 0.0597$ | $0.6628 \pm 0.0039$ | $\mathbf{38.9469 \pm 1.6742}$ | $0.8641 \pm 0.0119$ |
| QWEN-3.5-27B | 20 | ✓ | $\mathbf{0.6040 \pm 0.0172}$ | $0.8192 \pm 0.0087$ | $\mathbf{3.0177 \pm 0.1676}$ | $\mathbf{0.8601 \pm 0.0471}$ | $\mathbf{0.6686 \pm 0.0077}$ | $40.8280 \pm 1.6226$ | $0.8787 \pm 0.0219$ |
| *BoltzGen Baseline* | | | | | | | | | |
| BoltzGen | — | — | 0.5950 | **0.8236** | 3.0250 | 0.9487 | 0.6619 | 40.6574 | 0.8671 |

*Result* *: One of the iterations of the model did not produce any result for these metrics, hence we set them to 0 and not use in ranking.

### 4.3. Agents handle stochastic variation with discipline.

The Oracle Agent exhibits conservative and disciplined behaviour in two respects. First, it consistently applies a **noise-floor rule** when interpreting iteration-to-iteration changes in the number of designs passing BoltzGen filters. Because each iteration involves stochastic structure generation, small fluctuations in pass count do not necessarily indicate meaningful improvement or regression. The Oracle interprets this uncertainty symmetrically: it refuses to over-celebrate small increases and refuses to overreact to small decreases when changes fall within estimated noise thresholds. Domain experts viewed this behaviour positively, as it prevents the workflow from making large configuration changes in response to random variation.

Second, the Oracle follows a **conservative exploration strategy**. As shown in Figures 7h, 8h, and 9h, for the PD-L1, PDGFR, and CD3 targets respectively, and in Appendix E, the majority of configuration changes target the `binding_types` field — specifying where the binder should or should not attach — rather than modifying binder length or secondary structure simultaneously. Each iteration includes explicit reasoning for fields the Oracle chooses to modify or leave unchanged, with specific evidence cited from the literature or analysis agents - such an example is provided in Appendix G.2 for the most challenging target, CD3. Domain experts highlighted this as a useful prevention mechanism against a failure mode in which the agent randomly changes multiple design parameters at once, making their individual effects untraceable. Self-assessed confidence is generally moderate across models and targets (Appendix F, Figure 10), with GPT-5.4 MINI relatively underconfident in early iterations and CLAUDE-SONNET-4.6 more confident on PD-L1.

### 4.4. Agents fail to self-correct, plan experiments, or learn from their own history.

Despite the encouraging reasoning behaviours described above, systematic analysis of agent traces reveals concrete reliability failures that would need to be resolved before autonomous deployment.

**No self-correction on tool configuration errors.** When the Oracle selects an incorrect tool configuration — for example, specifying binding residues that produce consistently poor downstream metrics — it does not recognise the error and revert. Similarly, when analysis tools fail to execute (e.g., GetContacts producing no output), the Oracle acknowledges that key interface metrics are unavailable and proceeds with reduced confidence, but it does not rerun the failed tool, flag the issue as systematic, or pause the design loop until the missing evidence is restored. The agent recognises incomplete information but cannot recover from it. This type of behaviour usually points to the presence of hallucinations, mostly observed for the Llama model, highlighting that instruct models are not suitable for such a complex reasoning task - more details on the failure modes of the Llama model are given in Appendix I.

**Hypothesis oscillation without convergence criteria.** In multiple runs, the Oracle oscillates between hotspot hypotheses in the `binding_types` and `not_binding` fields, reverting to earlier configurations or making small residue-

*Table 2.* **Minibinder Lab results (PD-L1).** Pass rate (%) and structural/sequence quality metrics across models and iteration budgets ($N \in \{1, 10, 20\}$). Best results per metric in **bold**, when comparing averages. The **R** column marks reasoning models (✓). Metrics: *ipTM* (interface predicted TM-score), *PAE* (predicted aligned error, Å), *pTM* (predicted TM-score), *RMSD* (Å, vs. reference), *Conf.* (Boltz confidence score), *pIdent* (% sequence identity to known binders), *Cov.* (BLAST query coverage, %). Higher is better except for PAE, RMSD, pIdent and coverage (lower is better, marked ↓). Full metric definitions are reported in Appendix D.4. Full definitions of the models used are present in Appendix D.1.

| Model | $N$ | R | ipTM (↑) | pTM (↑) | PAE (↓) | RMSD (↓) | Conf. (↑) | pIdent (↓) | Cov. (↓) |
|---|---|---|---|---|---|---|---|---|---|
| *Generalist Reasoning Models* | | | | | | | | | |
| GPT-5.4 MINI | 1 | ✓ | $0.3783 \pm 0.0870$ | $0.7545 \pm 0.0213$ | $8.0130 \pm 3.2497$ | $2.1083 \pm 0.0861$ | $0.6608 \pm 0.0296$ | $26.8417 \pm 23.2698$* | $0.5963 \pm 0.5164$* |
| GPT-5.4 MINI | 10 | ✓ | $0.5365 \pm 0.0174$ | $0.8043 \pm 0.0051$ | $3.7219 \pm 0.4191$ | $1.6933 \pm 0.0656$ | $0.7109 \pm 0.0066$ | $39.8521 \pm 0.3688$ | $0.8645 \pm 0.0168$ |
| GPT-5.4 MINI | 20 | ✓ | $0.5702 \pm 0.0005$ | $0.8135 \pm 0.0020$ | $3.1614 \pm 0.0185$ | $1.5853 \pm 0.0367$ | $0.7232 \pm 0.0016$ | $39.8726 \pm 0.0134$ | $0.8521 \pm 0.0109$ |
| CLAUDE-SONNET 4.6 | 1 | ✓ | $0.3715 \pm 0.0446$ | $0.7721 \pm 0.0028$ | $7.9226 \pm 1.0924$ | $2.1820 \pm 0.0213$ | $0.6700 \pm 0.0074$ | $39.8461 \pm 3.1172$ | $\mathbf{0.8199 \pm 0.0115}$ |
| CLAUDE-SONNET 4.6 | 10 | ✓ | $0.5532 \pm 0.0358$ | $0.8099 \pm 0.0100$ | $3.5606 \pm 0.3592$ | $1.7086 \pm 0.0450$ | $0.7141 \pm 0.0067$ | $39.7833 \pm 0.2611$ | $0.8552 \pm 0.0042$ |
| CLAUDE-SONNET 4.6 | 20 | ✓ | $0.5882 \pm 0.0144$ | $0.8205 \pm 0.0164$ | $3.1546 \pm 0.0995$ | $1.5893 \pm 0.0909$ | $0.7227 \pm 0.0017$ | $39.8950 \pm 0.5482$ | $0.8563 \pm 0.0057$ |
| *Open-Weight Baselines* | | | | | | | | | |
| LLAMA-3.1-8B-INST | 1 | — | $0.3735 \pm 0.0467$ | $0.7690 \pm 0.0052$ | $8.1414 \pm 2.0258$ | $2.1143 \pm 0.0337$ | $0.6690 \pm 0.0084$ | $42.0208 \pm 2.8540$ | $0.8257 \pm 0.0733$ |
| LLAMA-3.1-8B-INST | 10 | — | $0.5513 \pm 0.0337$ | $0.8129 \pm 0.0143$ | $3.6738 \pm 0.3652$ | $1.6702 \pm 0.0724$ | $0.7179 \pm 0.0082$ | $40.1722 \pm 1.2001$ | $0.8655 \pm 0.0087$ |
| LLAMA-3.1-8B-INST | 20 | — | $0.5803 \pm 0.0252$ | $0.8241 \pm 0.0136$ | $3.3235 \pm 0.2621$ | $1.5238 \pm 0.0799$ | $0.7243 \pm 0.0050$ | $40.2768 \pm 1.0319$ | $0.8684 \pm 0.0090$ |
| QWEN-3.5-27B | 1 | ✓ | $0.4259 \pm 0.0013$ | $0.7618 \pm 0.0227$ | $6.4529 \pm 0.0540$ | $2.1723 \pm 0.0336$ | $0.6754 \pm 0.0094$ | $\mathbf{37.4583 \pm 1.1990}$ | $0.8827 \pm 0.0472$ |
| QWEN-3.5-27B | 10 | ✓ | $0.5733 \pm 0.0243$ | $0.8236 \pm 0.0184$ | $3.4260 \pm 0.2284$ | $1.6089 \pm 0.0356$ | $0.7215 \pm 0.0061$ | $39.2224 \pm 0.6410$ | $0.8537 \pm 0.0149$ |
| QWEN-3.5-27B | 20 | ✓ | $\mathbf{0.5944 \pm 0.0229}$ | $\mathbf{0.8283 \pm 0.0162}$ | $\mathbf{3.0975 \pm 0.2437}$ | $\mathbf{1.5081 \pm 0.0040}$ | $0.7263 \pm 0.0051$ | $39.9425 \pm 1.1649$ | $0.8485 \pm 0.0089$ |
| *BoltzGen Baseline* | | | | | | | | | |
| BoltzGen | — | — | 0.5859 | 0.8252 | 3.1003 | 1.5131 | **0.7313** | 40.3928 | 0.8737 |

*Result* *: One of the iterations of the model did not produce any result for these metrics, hence we set them to 0 and not use in ranking.

level edits without a clear convergence criterion. More details and an illustrative example are provided in Appendix H. The agent does not hold configurations fixed to estimate stochastic variability, nor does it systematically compare its predicted effects with later outcomes.

**No retrospective calibration.** The Oracle does not revisit previous predictions to assess whether its reasoning was correct. When predicting that a configuration change will improve a specific metric, it does not check whether the prediction held in the subsequent iteration. This means the agent cannot learn from its own mistakes within a single run — each iteration's reasoning is locally grounded but globally unanchored, with no clear trajectory to an optimum.

**Hallucination in open-weight models.** LLAMA-3.1-8B-INSTRUCT produced multiple hallucinated or weakly grounded claims across targets and iterations, which in turn highlight its failure to act as an agent and reason accordingly using the data available. A detailed analysis is provided in Appendix I. Smaller hallucinations were observed in other models. We did not observe clear hallucinations in CLAUDE-SONNET-4.6 traces, though the small number of runs prevents conclusions about model-level reliability.

Taken together, these failures reveal a consistent pattern: the agents can synthesise evidence and produce reasonable single-step reasoning, but lack the meta-cognitive capabilities required for multi-step experimental planning — holding hypotheses fixed, running controls, and calibrating predictions against outcomes. This gap between competent local reasoning and absent global planning represents a concrete reliability boundary for deploying agentic systems in open-ended scientific workflows.

## 5. Limitations

While our work represents a promising first step towards bridging the gap between multi-agent systems and interpretable minibinder design, several limitations remain. First, our study has limited statistical power. Although the results are promising and show that Minibinder Lab can approach, and in some cases exceed, the BoltzGen baseline, all experiments were repeated only three times across random seeds (where available). To support robust confidence intervals, future work should repeat each model-target setting 5-10 times to better quantify variability in pass rates, structural metrics, and model rankings. Further, to increase statistical power and reliability, confidence intervals based on an equivalent number of runs for the BoltzGen (Stark et al., 2025) baseline should be reported for equitable comparison.

Second, we have not performed a full ablation analysis. Several design choices may affect performance, including the number of Minibinder Lab iterations, the number of designs generated at each iteration, the retained-candidate budget, the LLM temperature, and the YAML fields the Oracle is allowed to edit. Although we evaluate four large language models, broader comparisons across additional open-weight and closed-source models would also be valuable.

Finally, all results in this work are computational. Generated binders are evaluated using structural confidence scores, interface metrics, sequence-similarity analyses, and expert review of reasoning traces, but they are not experimentally validated. Wet-lab testing will ultimately be required to assess expression, folding, binding affinity, specificity, and functional activity.

*Table 3.* **Minibinder Lab results (CD3).** Pass rate (%) and structural/sequence quality metrics across models and iteration budgets ($N \in \{1, 10, 20\}$). Best results per metric in **bold**, when comparing averages. The **R** column marks reasoning models (✓). Metrics: *ipTM* (interface predicted TM-score), *PAE* (predicted aligned error, Å), *pTM* (predicted TM-score), *RMSD* (Å, vs. reference), *Conf.* (Boltz confidence score), *pIdent* (% sequence identity to known binders), *Cov.* (BLAST query coverage, %). Higher is better except for PAE, RMSD, pIdent and coverage (lower is better, marked ↓). Full metric definitions are reported in Appendix D.4. Full definitions of the models used are present in Appendix D.1.

| Model | $N$ | R | ipTM (↑) | pTM (↑) | PAE (↓) | RMSD (↓) | Conf. (↑) | pIdent (↓) | Cov. (↓) |
|---|---|---|---|---|---|---|---|---|---|
| *Generalist Reasoning Models* | | | | | | | | | |
| GPT-5.4 MINI | 1 | ✓ | $0.2599 \pm 0.2306$** | $0.4768 \pm 0.4129$** | $4.6691 \pm 4.2529$** | $1.2451 \pm 1.1935$** | $0.4300 \pm 0.3725$** | $29.3667 \pm 25.6929$** | $0.5045 \pm 0.4397$** |
| GPT-5.4 MINI | 10 | ✓ | $0.5914 \pm 0.0203$ | $0.7725 \pm 0.0026$ | $2.0215 \pm 0.2278$ | $0.8109 \pm 0.0283$ | $0.7215 \pm 0.0089$ | $44.7345 \pm 0.5642$ | $0.8130 \pm 0.0151$ |
| GPT-5.4 MINI | 20 | ✓ | $0.6136 \pm 0.0174$ | $\mathbf{0.7816 \pm 0.0009}$ | $\mathbf{1.8248 \pm 0.2245}$ | $0.7573 \pm 0.0236$ | $0.7286 \pm 0.0058$ | $45.1523 \pm 0.6457$ | $0.8278 \pm 0.0109$ |
| CLAUDE-SONNET 4.6 | 1 | ✓ | $0.3542 \pm 0.0376$ | $0.7031 \pm 0.0027$ | $7.7170 \pm 1.2116$ | $1.3880 \pm 0.1663$ | $0.6491 \pm 0.0121$ | $43.7878 \pm 4.2519$ | $0.7551 \pm 0.0795$ |
| CLAUDE-SONNET 4.6 | 10 | ✓ | $0.5769 \pm 0.0023$ | $0.7705 \pm 0.0060$ | $2.2358 \pm 0.0834$ | $0.7766 \pm 0.0285$ | $0.7142 \pm 0.0055$ | $43.6307 \pm 0.6311$ | $0.8215 \pm 0.0306$ |
| CLAUDE-SONNET 4.6 | 20 | ✓ | $0.6046 \pm 0.0142$ | $0.7786 \pm 0.0072$ | $1.8554 \pm 0.2054$ | $\mathbf{0.7312 \pm 0.0178}$ | $0.7264 \pm 0.0009$ | $\mathbf{44.0102 \pm 0.8410}$ | $0.8187 \pm 0.0250$ |
| *Open-Weight Baselines* | | | | | | | | | |
| LLAMA-3.1-8B-INST | 1 | — | $0.4111 \pm 0.0418$ | $0.7074 \pm 0.0244$ | $6.2761 \pm 1.5054$ | $1.2837 \pm 0.1938$ | $0.6481 \pm 0.0080$ | $30.7667 \pm 26.6553$* | $0.4877 \pm 0.4224$* |
| LLAMA-3.1-8B-INST | 10 | — | $0.5767 \pm 0.0242$ | $0.7677 \pm 0.0029$ | $2.2614 \pm 0.2988$ | $0.8162 \pm 0.0107$ | $0.7119 \pm 0.0065$ | $44.8001 \pm 3.1265$ | $0.8108 \pm 0.0245$ |
| LLAMA-3.1-8B-INST | 20 | — | $0.6020 \pm 0.0178$ | $0.7802 \pm 0.0030$ | $2.0248 \pm 0.1902$ | $0.7559 \pm 0.0117$ | $0.7245 \pm 0.0045$ | $44.4663 \pm 1.6078$ | $0.8093 \pm 0.0105$ |
| QWEN-3.5-27B | 1 | ✓ | $0.4404 \pm 0.0559$ | $0.7339 \pm 0.0106$ | $4.9396 \pm 1.2867$ | $1.1623 \pm 0.0882$ | $0.6812 \pm 0.0095$ | $42.9700 \pm 3.2300$ | $0.8199 \pm 0.0560$ |
| QWEN-3.5-27B | 10 | ✓ | $0.5742 \pm 0.0175$ | $0.7653 \pm 0.0047$ | $2.2768 \pm 0.1265$ | $0.7811 \pm 0.0359$ | $0.7153 \pm 0.0051$ | $46.0755 \pm 2.6603$ | $0.8145 \pm 0.0054$ |
| QWEN-3.5-27B | 20 | ✓ | $0.6026 \pm 0.0196$ | $0.7787 \pm 0.0030$ | $2.0369 \pm 0.0533$ | $0.7384 \pm 0.0274$ | $0.7265 \pm 0.0086$ | $46.3354 \pm 0.9690$ | $\mathbf{0.8069 \pm 0.0221}$ |
| *BoltzGen Baseline* | | | | | | | | | |
| BoltzGen | — | — | **0.6201** | 0.7746 | 1.8364 | 0.7708 | **0.7337** | 45.9528 | 0.8102 |

*Result* *: One of the iterations of the model did not produce any result for these metrics, hence we set them to 0 and not use in ranking.
*Result* **: One of the iterations of the model did not contain any saved binders, therefore we set all metrics to 0 and not use in ranking.

## 6. Future Work

Future work will address the limitations identified above. First, we will run the pipeline across more independent random seeds for higher statistical power. We will also conduct systematic ablation studies to assess the contribution of key design choices. Second, we will improve the reliability of the agentic reasoning process. Because Minibinder Lab relies on language-model agents, hallucinations and weakly grounded claims remain possible. While this limitation cannot be removed entirely by prompt design alone, it can be reduced through stricter tool-output validation, citation checking, schema constraints, and expert-informed prompt refinement. Future versions should also include stronger consistency checks between the Oracle's reasoning and the outputs produced by the analysis agents.

Finally, we will extend the framework with stronger self-improvement and decision-making mechanisms. One direction is to introduce an additional JUDGE agent that evaluates Oracle recommendations against a domain-expert-defined scoring schema. This agent could encourage more principled hotspot selection, discourage uninformative configuration changes, and help decide when to exploit a promising hypothesis, explore a new region of the design space, or hold the configuration fixed to estimate stochastic variability. In the longer term, such feedback could be formalised using reinforcement-learning or bandit-style optimisation, where configuration edits are treated as actions and downstream structural or experimental outcomes provide rewards.

## 7. Conclusions

Minibinder Lab addresses a challenging problem in computational drug discovery: the efficient and interpretable generation of high-quality protein binders. We show that a multi-agent workflow can iteratively improve candidate designs, with structural metrics approaching and in some cases exceeding those obtained from a one-shot state-of-the-art protein generative model. Domain-expert evaluation further suggests that the system can produce useful reasoning by integrating literature, structural, energetic, novelty, and interface-analysis outputs.

At the same time, our results show that agentic reasoning remains limited across iterations. In particular, the Oracle can oscillate between design hypotheses, lacks explicit control experiments, and does not yet systematically calibrate predicted improvements against observed outcomes. Future work will address these limitations by improving robustness, strengthening hypothesis tracking and retrospective calibration, and increasing the biological, physical, and chemical awareness of the design loop.

## Acknowledgements

ARE is supported by the EPSRC Centre for Doctoral Training in Healthcare Data Science (EP/Y035321/1). JGH is supported by a Novo Nordisk Postdoctoral Fellowship in partnership with the University of Oxford, and the David Cockayne Junior Research Fellowship at Linacre College, Oxford.

## References


Ai, J.-Y., Liu, C.-F., Zhang, W., and Rao, G.-W. Current status of drugs targeting PDGF/PDGFR. *Drug Discovery Today*, 29(7):103989, July 2024. ISSN 13596446. doi: 10.1016/j.drudis.2024.103989. URL https://linkinghub.elsevier.com/retrieve/pii/S1359644624001144.

Butcher, J., Krishna, R., Mitra, R., Brent, R. I., Li, Y., Corley, N., Kim, P. T., Funk, J., Mathis, S., Salike, S., Muraishi, A., Eisenach, H., Thompson, T. R., Chen, J., Politanska, Y., Sehgal, E., Coventry, B., Zhang, O., Qiang, B., Didi, K., Kazman, M., DiMaio, F., and Baker, D. De novo Design of All-atom Biomolecular Interactions with RFdiffusion3. *bioRxiv*, pp. 2025.09.18.676967, November 2025. ISSN 2692-8205. doi: 10.1101/2025.09.18.676967. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC12458353/.

Chai Discovery Team, Boitreaud, J., Dent, J., Geisz, D., McPartlon, M., Meier, J., Qiao, Z., Rogozhnikov, A., Rollins, N., Wollenhaupt, P., and Wu, K. Zero-shot antibody design in a 24-well plate. *bioRxiv*, pp. 2025.07.05.663018, January 2025. doi: 10.1101/2025.07.05.663018. URL http://biorxiv.org/content/early/2025/07/06/2025.07.05.663018.abstract.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., Haas, R. J. d., Bethel, N., Leung, P. J. Y., Huddy, T. F., Pellock, S., Tischer, D., Chan, F., Koepnick, B., Nguyen, H., Kang, A., Sankaran, B., Bera, A. K., King, N. P., and Baker, D. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378 (6615):49–56, 2022. doi: 10.1126/science.add2187. URL https://www.science.org/doi/abs/10.1126/science.add2187. _eprint: https://www.science.org/doi/pdf/10.1126/science.add2187.

Eastman, P., Galvelis, R., Peláez, R. P., Abreu, C. R. A., Farr, S. E., Gallicchio, E., Gorenko, A., Henry, M. M., Hu, F., Huang, J., Krämer, A., Michel, J., Mitchell, J. A., Pande, V. S., Rodrigues, J. P., Rodriguez-Guerra, J., Simmonett, A. C., Singh, S., Swails, J., Turner, P., Wang, Y., Zhang, I., Chodera, J. D., De Fabritiis, G., and Markland, T. E. OpenMM 8: Molecular Dynamics Simulation with Machine Learning Potentials. *The journal of physical chemistry. B*, 128(1):109–116, January 2024. ISSN 1520-6106. doi: 10.1021/acs.jpcb.3c06662. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10846090/.

Geffner, T., Didi, K., Cao, Z., Reidenbach, D., Zhang, Z., Dallago, C., Kucukbenli, E., Kreis, K., and Vahdat, A. LaProteina: Atomistic Protein Generation via Partially Latent Flow Matching, July 2025. URL http://arxiv.org/abs/2507.09466. arXiv:2507.09466 [cs].

GetContacts. GetContacts. URL https://getcontacts.github.io/.

Ghareeb, A. E., Chang, B., Mitchener, L., Yiu, A., Szostkiewicz, C. J., Shved, D., Gyimesi, G. J., Laurent, J. M., Wright, S. M., Razzak, M. T., White, A. D., Finnemann, S. C., Hinks, M. M., and Rodriques, S. G. A multi-agent system for automating scientific discovery. *Nature*, May 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10652-y. URL https://doi.org/10.1038/s41586-026-10652-y.

Gong, J., Chehrazi-Raffle, A., Reddi, S., and Salgia, R. Development of PD-1 and PD-L1 inhibitors as a form of cancer immunotherapy: a comprehensive review of registration trials and future considerations. *Journal for ImmunoTherapy of Cancer*, 6(1):8, December 2018. ISSN 2051-1426. doi: 10.1186/s40425-018-0316-z. URL https://jitc.bmj.com/lookup/doi/10.1186/s40425-018-0316-z.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Sirkovic, P., Myaskovsky, A., Glowaty, G., Weissenberger, F., Orlandi, A., Popovici, D., Palepu, A., Rong, K., Tanno, R., Saab, K., Zhang, F., Blum, J., Carroll, A., Kulkarni, K., Tomašev, N., Zverinski, D., Rendulic, I., Vedadi, E., Hasler, F., Rimanic, L., Boia, M., Budiselic, I., Feinstein, B., Bellaiche, M., Sheffer, T., Freyberg, J., Ratcliff, J., Bertolli, O., Chou, K., Hassidim, A., Gokturk, B., Vahdat, A., Guan, Y., Dhillon, V., Vaishnav, E. D., Lee, B., Costa, T. R. D., Penadés, J. R., Peltz, G., Matias, Y., Manyika, J., Hassabis, D., Xu, Y., Kohli, P., Pawlosky, A., Karthikesalingam, A., and Natarajan, V. Accelerating scientific discovery with Co-Scientist. *Nature*, May 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10644-y. URL https://doi.org/10.1038/s41586-026-10644-y.

Hauser, M., Steinegger, M., and Söding, J. MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics*, 32 (9):1323–1330, May 2016. ISSN 1367-4811. doi: 10.1093/bioinformatics/btw006.

Hekkelman, M. L., Salmoral, D. , Perrakis, A., and Joosten, R. P. DSSP 4: FAIR annotation of protein secondary structure. *Protein Science*, 34(8):e70208, 2025. ISSN 1469-896X. doi: 10.1002/pro.70208. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.70208. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pro.70208.

Javed, S. A., Najmi, A., Ahsan, W., and Zoghebi, K. Targeting PD-1/PD-L-1 immune checkpoint inhibition for

cancer immunotherapy: success and challenges. *Frontiers in Immunology*, 15:1383456, 2024. ISSN 1664-3224. doi: 10.3389/fimmu.2024.1383456.

Jiang, Y., Chen, M., Nie, H., and Yuan, Y. PD-1 and PD-L1 in cancer immunotherapy: clinical implications and future considerations. *Human Vaccines & Immunotherapeutics*, 15(5):1111–1122, March 2019. ISSN 2164-5515. doi: 10.1080/21645515.2019.1571892. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC6605868/.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2.

Kim, Y., Gu, K., Park, C., Park, C., Schmidgall, S., Heydari, A. A., Yan, Y., Zhang, Z., Zhuang, Y., Liu, Y., Malhotra, M., Liang, P. P., Park, H. W., Yang, Y., Xu, X., Du, Y., Patel, S., Althoff, T., McDuff, D., and Liu, X. Towards a science of scaling agent systems, 2026. URL https://arxiv.org/abs/2512.08296.

Latent Labs Team, Bridgland, A., Crabbé, J., Kenlay, H., Pretorius, D., Schmon, S. M., Hilmkil, A., Bartke-Croughan, R., Rombach, R., Flashman, M., Matteson, T., Mathis, S., Nelson, A. W. R., Yuan, D., Obika, A., and Kohl, S. A. A. Latent-X: An Atom-level Frontier Model for De Novo Protein Binder Design, July 2025. URL http://arxiv.org/abs/2507.19375. arXiv:2507.19375 [q-bio].

Latent Labs Team, Schmon, S. M., Pretorius, D., Mathis, S., Bartke-Croughan, R., Puvanendran, A., Vuckovic, J., Kenlay, H., Vlachynská, M., Bridgland, A., Grishin, I., Over, S., Li, D., Li, B., Crabbé, J., Hilmkil, A., Nelson, A. W. R., Yuan, D., Obika, A., and Kohl, S. A. A. Latent-Y: A Lab-Validated Autonomous Agent for De Novo Drug Design, 2026. URL https://arxiv.org/abs/2603.29727. Version Number: 2.

Leman, J. K., Weitzner, B. D., Lewis, S. M., Adolf-Bryfogle, J., Alam, N., Alford, R. F., Aprahamian, M., Baker, D., Barlow, K. A., Barth, P., Basanta, B., Bender, B. J., Blacklock, K., Bonet, J., Boyken, S. E., Bradley, P., Bystroff, C., Conway, P., Cooper, S., Correia, B. E., Coventry, B., Das, R., De Jong, R. M., DiMaio, F., Dsilva, L., Dunbrack, R., Ford, A. S., Frenz, B., Fu, D. Y., Geniesse, C., Goldschmidt, L., Gowthaman, R., Gray, J. J., Gront, D., Guffy, S., Horowitz, S., Huang, P.-S., Huber, T., Jacobs, T. M., Jeliazkov, J. R., Johnson, D. K., Kappel, K., Karanicolas, J., Khakzad, H., Khar, K. R., Khare, S. D., Khatib, F., Khramushin, A., King, I. C., Kleffner, R., Koepnick, B., Kortemme, T., Kuenze, G., Kuhlman, B., Kuroda, D., Labonte, J. W., Lai, J. K., Lapidoth, G., Leaver-Fay, A., Lindert, S., Linsky, T., London, N., Lubin, J. H., Lyskov, S., Maguire, J., Malmström, L., Marcos, E., Marcu, O., Marze, N. A., Meiler, J., Moretti, R., Mulligan, V. K., Nerli, S., Norn, C., Ó'Conchúir, S., Ollikainen, N., Ovchinnikov, S., Pacella, M. S., Pan, X., Park, H., Pavlovicz, R. E., Pethe, M., Pierce, B. G., Pilla, K. B., Raveh, B., Renfrew, P. D., Burman, S. S. R., Rubenstein, A., Sauer, M. F., Scheck, A., Schief, W., Schueler-Furman, O., Sedan, Y., Sevy, A. M., Sgourakis, N. G., Shi, L., Siegel, J. B., Silva, D.-A., Smith, S., Song, Y., Stein, A., Szegedy, M., Teets, F. D., Thyme, S. B., Wang, R. Y.-R., Watkins, A., Zimmerman, L., and Bonneau, R. Macromolecular modeling and design in Rosetta: recent methods and frameworks. *Nature Methods*, 17(7):665–680, July 2020. ISSN 1548-7105. doi: 10.1038/s41592-020-0848-2. URL https://www.nature.com/articles/s41592-020-0848-2.

Liu, D., Hu, X., Chen, Z., Wei, W., and Wu, Y. Key links in the physiological regulation of the immune system and disease induction: T cell receptor -CD3 complex. *Biochemical Pharmacology*, 227:116441, September 2024. ISSN 00062952. doi: 10.1016/j.bcp.2024.116441. URL https://linkinghub.elsevier.com/retrieve/pii/S0006295224004246.

Liu, S., Xu, H., Ai, Y., Li, H., Bengio, Y., and Guo, H. Expert-Guided LLM Reasoning for Battery Discovery: From AI-Driven Hypothesis to Synthesis and Characterization, July 2025. URL http://arxiv.org/abs/2507.16110. arXiv:2507.16110 [cs].

M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, May 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00832-8. URL https://doi.org/10.1038/s42256-024-00832-8.

Overath, M. D., Rygaard, A., Jacobsen, C. P., Brasas, V., Morell, O., Sormanni, P., and Jenkins, T. P. Predicting Experimental Success in De Novo Binder Design: A Meta-Analysis of 3,766 Experimentally Characterised Binders, August 2025. URL https://www.biorxiv.org/content/10.1101/2025.08.14.670059v1. ISSN: 2692-8205 Pages: 2025.08.14.670059 Section: New Results.

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*, pp. 2025.06.14.659707, June 2025. ISSN 2692-8205. doi: 10.1101/2025.06.14.659707. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC12262699/.

Pearlman, D. A., Case, D. A., Caldwell, J. W., Ross, W. S., Cheatham, T. E., DeBolt, S., Ferguson, D., Seibel, G., and Kollman, P. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications*, 91(1):1–41, September 1995. ISSN 0010-4655. doi: 10.1016/0010-4655(95)00041-D. URL https://www.sciencedirect.com/science/article/pii/001046559500041D.

Ponnapati, M., Cox, S., Gordon, C. W., Hammerling, M. J., Narayanan, S., Laurent, J. M., Braza, J. D., Hinks, M. M., Skarlinski, M. D., Rodriques, S. G., and White, A. ProteinCrow: A Language Model Agent That Can Design Proteins. July 2025. URL https://openreview.net/forum?id=ljXgWDtqCu.

Stark, H., Faltings, F., Choi, M., Xie, Y., Hur, E., O'Donnell, T., Bushuiev, A., Uçar, T., Passaro, S., Mao, W., Reveiz, M., Bushuiev, R., Pluskal, T., Sivic, J., Kreis, K., Vahdat, A., Ray, S., Goldstein, J. T., Savinov, A., Hambalek, J. A., Gupta, A., Taquiri-Diaz, D. A., Zhang, Y., Hatstat, A. K., Arada, A., Kim, N. H., Tackie-Yarboi, E., Boselli, D., Schnaider, L., Liu, C. C., Li, G.-W., Hnisz, D., Sabatini, D. M., DeGrado, W. F., Wohlwend, J., Corso, G., Barzilay, R., and Jaakkola, T. BoltzGen: Toward Universal Binder Design, November 2025. URL https://www.biorxiv.org/content/10.1101/2025.11.20.689494v1. ISSN: 2692-8205 Pages: 2025.11.20.689494 Section: New Results.

Steinegger, M. and Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, November 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988. URL https://www.nature.com/articles/nbt.3988.

Styles, M. J., Pixley, J. A., Wei, T., Basile, C., Lu, S. S., and Dickinson, B. C. PANCS-Binders: a rapid, high-throughput binder discovery platform. *Nature Methods*, 22(8):1720–1730, August 2025. ISSN 1548-7105. doi: 10.1038/s41592-025-02740-0. URL https://doi.org/10.1038/s41592-025-02740-0.

Swanson, K., Wu, W., Bulaong, N. L., Pak, J. E., and Zou, J. The Virtual Lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646(8085):716–723, October 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09442-9.

UniProtKB P04234. UniProtKB P04234: T-cell surface glycoprotein CD3 delta chain (CD3D). https://www.uniprot.org/uniprotkb/P04234/entry, 2026. Accessed: 2026-05-06.

UniProtKB P09619. UniProtKB P09619: Platelet-derived growth factor receptor beta (PDGFRB). https://www.uniprot.org/uniprotkb/P09619/entry, 2026. Accessed: 2026-05-06.

UniProtKB Q9NZQ7. UniProtKB Q9NZQ7: Programmed cell death 1 ligand 1 (CD274). https://www.uniprot.org/uniprotkb/Q9NZQ7/entry, 2026. Accessed: 2026-05-06.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, August 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06415-8. URL https://www.nature.com/articles/s41586-023-06415-8.

Wu, Z. and Cheung, N. T cell engaging bispecific antibody (T-BsAb): From technology to therapeutics. *Pharmacology & Therapeutics*, 182:161–175, February 2018. ISSN 01637258. doi: 10.1016/j.pharmthera.2017.08.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0163725817302152.

Xu, R. and Yan, Y. Agent skills for large language models: Architecture, acquisition, security, and the path forward, 2026. URL https://arxiv.org/abs/2602.12430.

Zambaldi, V., La, D., Chu, A. E., Patani, H., Danson, A. E., Kwan, T. O. C., Frerix, T., Schneider, R. G., Saxton, D., Thillaisundaram, A., Wu, Z., Moraes, I., Lange, O., Papa, E., Stanton, G., Martin, V., Singh, S., Wong, L. H., Bates, R., Kohl, S. A., Abramson, J., Senior, A. W., Alguel, Y., Wu, M. Y., Aspalter, I. M., Bentley, K., Bauer, D. L. V., Cherepanov, P., Hassabis, D., Kohli, P., Fergus, R., and Wang, J. De novo design of high-affinity protein binders with AlphaProteo, September 2024. URL http://arxiv.org/abs/2409.08022. arXiv:2409.08022 [q-bio].

# Appendix

## Impact

This paper presents work whose goal is to advance the field of Artificial Intelligence in Drug Discovery, focusing on large language models applied to interpretable therapeutic protein binder design, generation, analysis and optimisation. While our work is methodological in nature, it was developed with a concrete use case - applications in computational drug discovery. This goal carries both potential benefits, i.e. accelerating the identification of therapeutic candidates, and risks, i.e. the possibility of misuse for designing harmful compounds. We believe the techniques presented here pose no greater risk than existing publicly available cheminformatics tools and language models, and we encourage responsible use within established scientific and regulatory frameworks.

## LLM Usage

We acknowledge the use of the Large Language Models (LLMs) GPT-5.4 MINI, CLAUDE-SONNET 4.6, QWEN3.5-27B and LLAMA-3.1-8B-INSTRUCT as the backbones for the Minibinder Lab. LLMs were used in the preparation of this manuscript solely for rewording and minor editorial polishing of author-written text, consistent with practices reported in prior work and LaTeX formatting of figures. Authors also acknowledge the use of LLMs for generating TikZ/text figures, analysis code, and modifying SKILL.md files across agents. All technical and scientific content, experimental design, analysis, and conclusions are entirely the work of the authors.

## A. Methods - Agentic Framework

The **Minibinder Lab** consists of 12 highly specialised agents, each with a distinct function, which we describe in more detail in the following subsections.

### A.1. Literature Expert Agent

The **"Literature Expert" Agent** runs only in the first iteration of the multi-agentic workflow, constructing a structured biological brief for the target protein by querying UniProt (function, domains, annotated binding sites), the RCSB PDB (co-crystal discovery and atomic-coordinate–derived interface residues at small atomic distance between carbon atoms), and Europe PMC (mutagenesis data, therapeutic-antibody epitopes, mechanism of action). The resulting JSON brief — comprising published epitopes, functional hotspots, non-productive surfaces, and actionable design guidance with residue-level citations — is saved and reused as a fixed prior by the *Orchestrator* and the *Oracle* agents in all subsequent design iterations, avoiding inefficient per-iteration literature retrieval.

### A.2. Designer Agent

The **"Designer"Agent** proposes backbones and sequences for the intended site/interface of the minibinder. It also computes metrics related to the stability of the folded protein structure.

This agent builds on the state-of-the-art software in protein structure design, BoltzGen (Stark et al., 2025). Upon receiving a configuration file for an input target, together with other parameters based on user preference (Table 4), the tool acts as a practical, end-to-end pipeline for the *in silico* minibinder structure design. We chose to employ BoltzGen instead of the usual pipeline observed in the drug discovery space (i.e. a model for backbone structure generation, e.g., RFdiffusion (Watson et al., 2023), a model for inverse folding, e.g., ProteinMPNN (Dauparas et al., 2022), and a model for refolding, e.g., Chai-1/Chai-2 (Chai Discovery Team et al., 2025)) due to its ability to perform all the usual pipeline steps in a single run, without any prompting or user intervention. Furthermore, it also provides the user with a clear report of the top ranked generated structures, together with significant metrics for *filtering* (e.g RMSD-based metrics, presence of certain classes of amino acids, percentage of a specific amino acid in the structure) and scoring (e.g. AlphaFold (Jumper et al., 2021)-based metrics such as ipTM, pTM for alignment between design and target, folding confidence, PAE score, number of hydrogen bonds, ionic connections etc.), which will be used in the further refined training stages of the agentic framework.

*Table 4.* **Designer Agent Input Parameters**. All the parameters below are a subset of the available parameters employed by the BoltzGen tool, respecting the same specifications from Stark et al. (2025).

| Parameter | Description |
|---|---|
| Input File | ”yaml” configuration file of the target, avoidable regions etc. |
| Output Folder | Folder where the outputs from BoltzGen are stored |
| Protocol | Protocol to be used in design process |
| Number designs | Number of minibinders designs to generate in total |
| Budget | Number of top design to be stored, analysed, improved |

### A.3. Preprocessor Agent

This agent performs preprocessing functionalities required for the generated structures to undergo further analysis: CIF-to-PDB conversion, hydrogen addition, energy minimisation.

### A.4. Novelty Agent

The **“Novelty” Agent** should act as a world-class expert in sequence similarity search, homology detection, and protein design interpretation. The role it has in this pipeline is to run MMseqs2 (Hauser et al., 2016; Steinegger & Söding, 2017) across the preprocessed binder designs produced by the Designer Agent, and then processed by the Preprocessor, and to return a per-design novelty / plausibility profile that the Orchestrator, Reporter, and Oracle can use alongside the structural and energetic agents.

### A.5. Analysis Agents

#### A.5.1. Distogram Creator Agent

The **“Distogram Creator” Agent** offers insight into the compact structural fingerprint of protein geometry, capturing fold shape and compactness, revealing chain-chain proximity patterns, and therefore being used for comparing generated binders across candidates.

#### A.5.2. Ramachandran Creator Agent

The **“Ramachandran Creator” Agent** gives a local stereochemical quality filter that complements global metrics, helping reject candidates with unrealistic backbone geometry even if interface contacts appear good, being a classic structural quality control technique.

#### A.5.3. GetContacts Expert Agent

The **“GetContacts Expert” Agent** explains why a binder may score well or poorly, supporting targeted redesign, and enabling objective interface metrics by employing GetContacts (GetContacts), an important interface interpretability tool.

#### A.5.4. DSSP Expert Agent

This agent adds structural-context annotations that help interpret other metrics, since the secondary structure is a key stability signal, by employing the corresponding open-source DSSP (Hekkelman et al., 2025) tool.

#### A.5.5. Rosetta Expert Agent

The **“Rosetta Expert” Agent** employs the PyRosetta (version of Rosetta (Leman et al., 2020) implemented in Python) tool to give a fast, physically motivated proxy for structural plausibility (clashes, rotamer strain, backbone strain), interaction quality (electrostatics, H-bonds, solvation, packing) and interface strength($\Delta G$-like interface energies, buried surface area, unsatisfied H-bonds), complementing the geometric quality control methods performed by the other *Analysis Agents* (distograms, Ramachandran plots, DSSP) and contact fingerprints (GetContacts) with an energy-based ranking signal.

### A.6. Reporter Agent

The **"Reporter" Agent** collects and extracts key information from all the analysis files resulting from the previously-mentioned agents, and generates a summary report of the most relevant findings.

### A.7. Oracle Agent

The **"Oracle" Agent** reviews the top protein designs from the *"Designer"Agent*, diagnoses why designs fail filters or score poorly, and rewrites the input configuration to maximise pass rate and improve analysis metrics in the next iteration. In particular, this agent is allowed to propose changes to the following **3 fields** in the minibinder configuration file: **sequence** (length range of the designed protein), **binding type** (bias sampling toward productive target regions, by introducing binding or non-binding spots or regions), **secondary structure** (constrain secondary structure of the designed protein, by imposing the presence of $\alpha-$helices, $\beta-$sheets, coils etc.). The Oracle also evaluates the generated designs at each iteration and, based on reasoning, saves the best ones for future reference.

### A.8. Orchestrator

The **"Orchestrator" Agent** acts as the central coordinator of the multi-agent protein design and analysis pipeline, by managing all previous agents, selecting analysis tools, deciding whether to restart/finalise the pipeline, tracking passing structures across iterations, and enforcing configuration handoff between iterations.

## B. Prompts

### B.1. User Prompt

Below, we provide an example of how the user interacts with the Minibinder Lab. The target field should be replaced by PD-L1, PDGFR or CD3, and the output_folder by the name of the folder where the user would like to store the findings. Please note that the user interacts only once with the Minibinder Lab, at the beginning of the workflow - the iterative binder optimisation process is autonomous.

**User Prompt**

```
I need to run a design.  Use the input file {target}_config.yaml and set the output
folder {output_folder}, use the protocol protein-anything, set num_designs to 100,
and set a budget of 10.
```

### B.2. Agentic Prompts

For the purpose of this work, we decided to experiment with the "Agent Skills" (Xu & Yan, 2026) setup proposed in https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview. This is an alternative way of approaching the prompting process of our agents, promoting modularity and the convergence of our agentic entities from general-purpose models to AI domain-experts. In particular, given the high level of specialisation required for each agent to be successful, together with specific protein analysis tools and the requirements for critical reasoning iteratively, we believe standard prompts would be more prone to failures due to their one-off task focus. On the other hand, skills only load on demand, when one agent is activated in the workflow, and do not require the same guidance to be repeated, leading to a more optimal instruction process and the ability of our agents to achieve their functions and contribute to the multi-agentic workflow.

### B.2.1. LITERATURE EXPERT AGENT SKILL

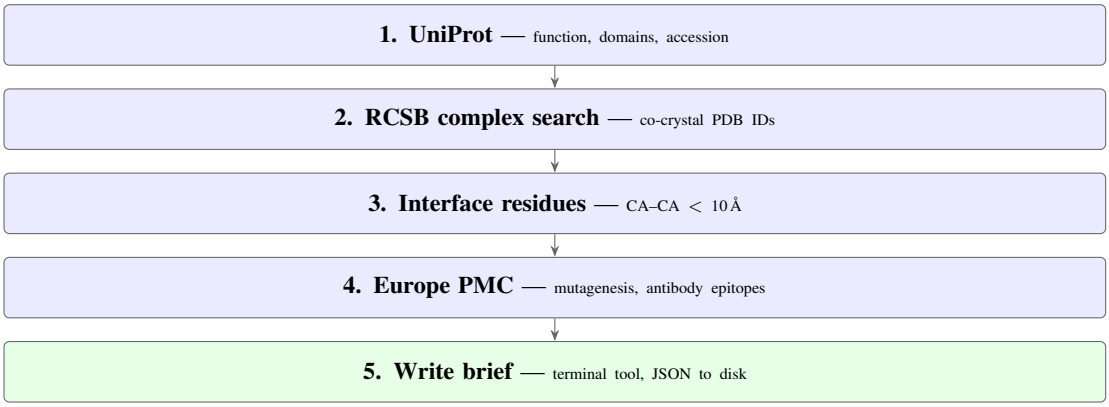



*Figure 2.* **Literature Expert agent (LITERATURE EXPERT).** A one-shot, campaign-level scout that builds a structured biological brief before iterative design begins. The agent calls four evidence-gathering tools in sequence (UniProt → RCSB complex search → interface residues → Europe PMC) and terminates with a single `write_literature_brief` call (per-tool descriptions in Table 5). The resulting JSON brief — containing residue-level epitopes, hotspots, and design advice — is cached on disk and reused by the Orchestrator and Oracle agents across all subsequent iterations, eliminating per-iteration literature lookups.

*Table 5.* Tools available to the Literature Expert Agent (LITERATURE EXPERT). All four research tools are backed by free, keyless REST APIs; results are cached on disk (7-day TTL) so reruns and intra-run duplicates pay zero HTTP cost. The fifth tool is terminal — calling it ends the agent run and delivers the output: a literature brief based on the target of interest.

| Tool | Source | Purpose |
|---|---|---|
| `uniprot_search` | UniProtKB (`uniprot.org/help/uniprotkb`) | Resolve target name to accession. Fetch function, domains, subcellular location, and annotated binding sites. |
| `rcsbcomplex_search` | RCSB PDB (`rcsb.org`) | Find co-crystal / cryo-EM structures containing the target in a multi-chain complex (queries by UniProt accession). |
| `rcsbinterface_residues` | RCSB PDB (`rcsb.org`) | Compute interface residues between two chains of a PDB entry (CA–CA distance $< 10$ Å). Provide ground-truth contact residues from atomic coordinates. |
| `pubmed_search` | Europe PMC (`europepmc.org`) | Search peer-reviewed literature for mutagenesis data, therapeutic antibody epitopes, and mechanism of action (returns PMID + abstract). |
| `writeliterature_brief` | Local Disk | *Terminal tool.* Persist the structured JSON brief (residues, hotspots, design advice) to the path supplied by the Orchestrator and end the run. |

### B.2.2. DESIGNER AGENT SKILL

*Table 6.* Tool available to the Designer Agent (DESIGNER). In particular, it receives 5 arguments: *input_file* (path to a YAML configuration file containing the requirements of the design), *output_folder*, *protocol* (design protocol to be used when generating the structures), *num_designs* (total number of binders to be generated), and *budget* (number of top quality designs to be ranked). The tool writes a json file containing all these input parameters, then it runs the BoltzGen complete pipeline for binder design, as introduced by Stark et al. (2025). Executing the BoltzGen pipeline is terminal — calling it ends the agent run and delivers the standard BoltzGen output - a summarisation report, structural metrics, 3D structures of all generated designs etc.

| Tool | Source | Purpose |
|---|---|---|
| `boltzgenjson tool` | BoltzGen (Stark et al., 2025) | Collect the necessary parameters for a BoltzGen run, write them to a JSON file, and then run the BoltzGen pipeline in-process on a dedicated GPU. |

### B.2.3. PREPROCESSOR AGENT SKILL

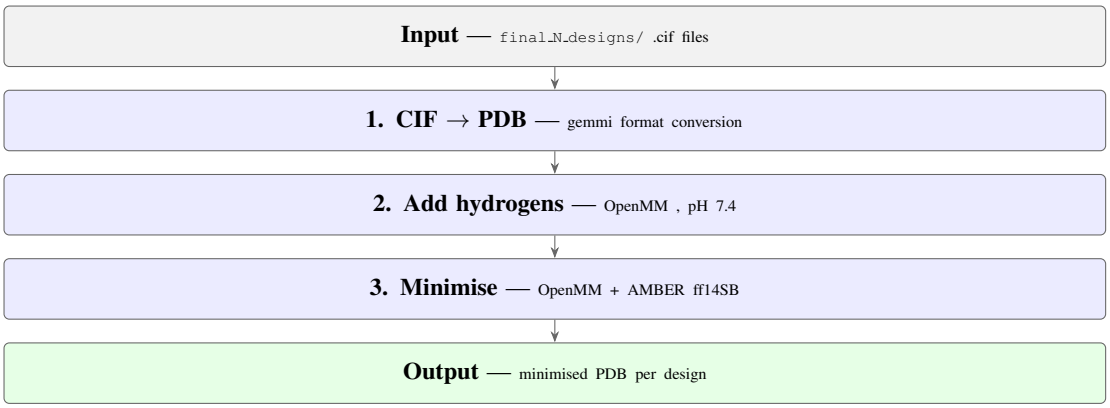



*Figure 3.* **Preprocessor Agent (PREPROCESSOR).** A single-tool agent that prepares BoltzGen-generated structures for downstream analysis. The agent makes exactly one call to `preprocessing_tool`, which internally executes three sequential stages: format conversion (gemmi), pH-aware hydrogen addition (OpenMM Modeller), and restrained vacuum energy minimisation (OpenMM with AMBER). The tool iterates over every file in the subfolder produced by BoltzGen. Intermediate files (converted PDB, hydrogenated PDB) are deleted after minimisation, leaving only the final minimised PDB per design in-place. Per-stage parameters in Table 7.

*Table 7.* Internal stages of `preprocessing_tool` called by the Preprocessor Agent (PREPROCESSOR). All three stages run within a single tool invocation; only the final minimised PDB is retained on disk. Heavy-atom positional restraints keep the global structure near the BoltzGen prediction while allowing local cleanup of newly-placed hydrogens and strained sidechain conformations — a fast physical sanity check before downstream contact analysis and ranking.

| Stage | Library | Purpose |
|---|---|---|
| `cif_to_pdb` | gemmi | Parse the mmCIF hierarchy (models → chains → residues → atoms) and serialise it to PDB records (`ATOM`, `HETATM`, `TER`), preserving Cartesian coordinates $\mathbf{r}_i = (x_i, y_i, z_i)$ for every atom. Pure format conversion; geometry is unchanged. |
| `add_hydrogens` | OpenMM (Eastman et al., 2024) | Infer hydrogen positions from atom types, residue templates, and pH-aware protonation heuristics (default pH = 7.4). Hydrogens are placed to satisfy idealised bond lengths and angles around each heavy atom. Falls back to OpenMM's built-in heuristics if force-field-guided placement fails. |
| `minimise structure` | OpenMM (Eastman et al., 2024), AMBER (Pearlman et al., 1995) | Restrained vacuum energy minimisation $\mathbf{r}^* = \arg\min_{\mathbf{r}} \left[U(\mathbf{r}) + \sum_i k\|\mathbf{r}_i - \mathbf{r}_{i,0}\|^2\right]$ with $k = 10^3$ kJ/(mol nm$^2$) on heavy atoms (defaults: `NoCutoff`, `HBonds` constraints, $\leq 500$ iterations, tolerance 10 kJ/(mol nm)). Removes steric clashes and poor hydrogen placement without perturbing the predicted fold or interface. |
| `preprocessing tool` | Local | *Wrapper tool.* Iterate over every generated structure file in the folder, run the three stages above sequentially per file, delete intermediates, and return a per-design dictionary with stage-level metadata (energy before/after, hydrogens added, restrained-atom count, force-field mode). |

### B.2.4. NOVELTY AGENT SKILL

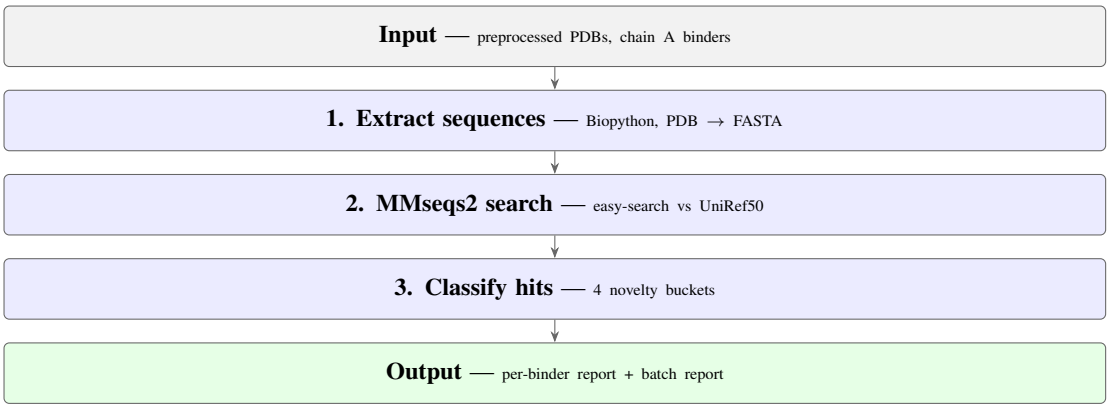



*Figure 4.* **Novelty Agent (NOVELTY).** A single-tool agent that scores the sequence-level novelty and biological plausibility of every preprocessed binder. The agent makes exactly one call to `mmseqs_tool`, which internally executes three sequential stages: chain-A sequence extraction (Biopython PDBParser, PDB → combined multi-entry FASTA), batched MMseqs `easy-search` against the target database (e.g. UniRef50, sensitivity $s = 7.5$, $E$-value cutoff $10^{-3}$), and hit classification into four novelty buckets (`known_scaffold`, `moderate_homology`, `weak_signal`, `no_hit`) using best-hit $E$-value, fractional identity, and query coverage. Per-binder `report` blocks (class, plausibility, novelty, coverage, identity, $E$-value, recommendation) and a batch-level `batch_report` are written to `search_summary.json` for downstream consumption by the Orchestrator, Reporter, and Oracle.

*Table 8.* Internal stages of `mmseqs_tool` called by the Novelty Agent (NOVELTY). All three stages run within a single tool invocation. Classification thresholds: `known_scaffold` requires $E \leq 10^{-20}$, identity $\geq 0.8$, and coverage $\geq 0.8$; `moderate_homology` requires $E \leq 10^{-5}$, identity $\geq 0.25$, and coverage $\geq 0.5$; `weak_signal` catches any remaining hits with $E \leq 10^{-3}$ that fail the moderate threshold; `no_hit` applies when no hits pass the $E$-value cutoff. The agent emits the `batch_report` and per-binder `report` blocks verbatim so the Reporter and Oracle can integrate them programmatically without re-parsing the raw `.m8`.

| Stage | Library | Purpose |
|---|---|---|
| `extract_sequences` | Biopython (PDBParser) | Parse every preprocessed PDB in the binder batch and extract the chain-A binder sequence (three-letter to one-letter mapping, standard amino acids only). All extracted sequences are written to a single combined `query.fasta` so the entire batch is searched in one MMseqs invocation, while remaining per-binder attributable via FASTA headers (PDB base names). |
| `easy_search` | MMseqs (Hauser et al., 2016; Steinegger & Söding, 2017) | Run `mmseqs easy-search` against the target database (e.g. UniRef50) with default sensitivity $s = 7.5$, $E$-value cutoff $10^{-3}$, and `--max-seqs` $= 100$. Produces a tabular `result.m8` with ⟨query, target, pident, alnlen, qstart, qend, evalue, bits⟩ per hit. |
| `classify_hits` | Local | For each query, sort hits by $E$-value and assign a novelty class from {`known_scaffold`, `moderate_homology`, `weak_signal`, `no_hit`} using best-hit $E$-value, fractional identity, and query coverage $=$ (qend $-$ qstart $+ 1)/L_{\text{query}}$. Build a per-query `report` (class, plausibility, novelty, coverage, identity, evalue, alnlen, $n$ hits, interpretation, recommendation) and a batch-level `batch_report` (class counts, known-scaffold list, no-hit list, one-line orchestrator recommendation). |
| `mmseqs_tool` | Local | *Wrapper tool.* The agent's only action: called exactly once per iteration. Default invocation is `command="search_directory"` over the preprocessed binder batch with `chain="A"`; the alternative `command="search"` mode runs a single-file query (e.g. a one-off target-chain sanity check on chain B). Writes `query.fasta`, `result.m8`, and `search_summary.json` to `output_dir`; the JSON is consumed downstream by the Orchestrator, Reporter, and Oracle. |

### B.2.5. DISTOGRAM CREATOR AGENT SKILL

*Table 9.* Tool available to the Distogram Creator Agent (DISTOGRAM CREATOR). A C$\alpha$ distogram is a compact structural fingerprint of protein geometry, capturing fold shape and compactness, revealing chain-chain proximity patterns, being useful for comparing generated binders across candidates.

| Tool | Source | Purpose |
|---|---|---|
| `c_alpha_distogram_tool` | Local Disk | Compute the pairwise distance matrix between all C$\alpha$ atoms and export it as a csv and a heatmap. |

### B.2.6. RAMACHANDRAN CREATOR AGENT SKILL

*Table 10.* Tool available to the Ramachandran Creator Agent (RAMACHANDRAN CREATOR). For each design, it outputs a CSV with per-residue $\phi/\psi$ angles and a PNG hexbin plot for visual interpretation. The tool detects unrealistic backbone conformations, highlights strained loops / outliers and helps separate binder-specific geometry issues from receptor geometry - this is especially important for structures obtained using generative models, such as BoltzGen (Stark et al., 2025), which may produce plausible global structures but poor local torsions.

| Tool | Source | Purpose |
|---|---|---|
| `ramachandran_tool` | Local | Compute backbone dihedral angles $\phi$ and $\psi$ for all residues and generate Ramachandran plots (all chains, binder-only, receptor-only). |

### B.2.7. GETCONTACTS EXPERT AGENT SKILL

*Table 11.* Tools available to the GetContacts Expert Agent (GETCONTACTS EXPERT). Run locally, GetContacts (GetContacts) provides a rich, interpretable interaction fingerprint for binder-target interfaces, by applying geometry-based interaction criteria, e.g., distance cutoffs, angle constraints and atom/residue typing rules. Its significance in our pipeline is crucial - it explains why a binder may score well/poorly, supports targeted redesign, and enables objective interface metrics.

| Tool | Source | Purpose |
|---|---|---|
| `getcontacts_tool` | GetContacts (GetContacts) | Run the static GetContacts analysis on a single structure to identify non-covalent interactions (hydrogen bonds, salt bridges, hydrophobic contacts, Van der Waals forces, aromatic interactions) between selected atom groups or chains. |

### B.2.8. DSSP EXPERT AGENT SKILL

*Table 12.* Tools available to the DSSP Expert Agent (DSSP EXPERT). Installed and running locally, DSSP (Hekkelman et al., 2025) assigns secondary structure from backbone geometry and hydrogen-bond patterns, and reports backbone dihedrals ($\phi, \psi$) and solvent exposure measures. Its role in the pipeline is to add structural-context annotations that help interpret other metrics.

| Tool | Source | Purpose |
|---|---|---|
| `dssp_tool` | DSSP 4 (Hekkelman et al., 2025) | Run DSSP on a generated structure to assign per-residue secondary structure and associated geometric descriptors (including $\phi$, $\psi$, relative accessibility) and export a parsed CSV. |

### B.2.9. ROSETTA EXPERT AGENT SKILL

*Table 13.* Tools available to the Rosetta Expert Agent (ROSETTA EXPERT). Installed locally and deployed using a Python version, Rosetta scores give a fast, physically motivated proxy for structural plausibility (clashes, rotamer strain, backbone strain), interaction quality (electrostatics, H-bonds, solvation, packing) and interface strength/cleanliness (delta-G-like interface energies, buried surface area, unsatisfied H-bonds).

| Tool | Source | Purpose |
|---|---|---|
| `pyrosetta scoring tool` | PyRosetta (Leman et al., 2020) | Run Rosetta's scoring function and InterfaceAnalyzer on every designed structure to produce a decomposed energy breakdown of each complex and interface-specific binding/packing metrics. |

### B.2.10. REPORTER AGENT SKILL

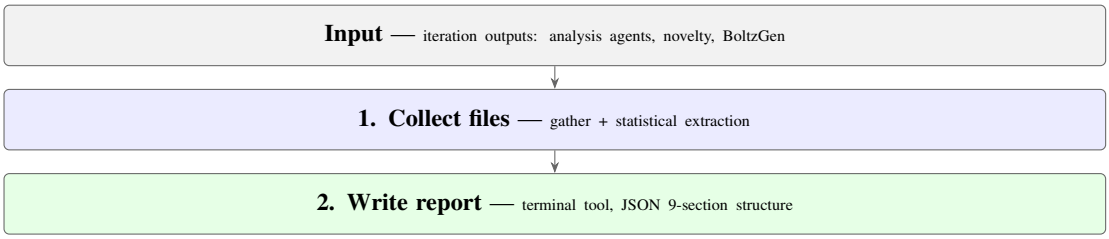



*Figure 5.* **Reporter Agent (REPORTER).** A two-tool synthesis agent that consolidates every per-iteration output into a single structured scientific report. The agent first calls `collect_summary_files` to gather key files from the BoltzGen runs, and the analysis files from the subsequent agents, returning compact statistical extractions. The agent then reasons across the collected data — assembling a 9-section narrative (HEADER, OVERVIEW, DESIGN COMPARISON, PREPROCESSING, STRUCTURAL ANALYSIS, SEQUENCE COMPOSITION, CROSS-TOOL DESIGN SYNTHESIS, SCIENTIFIC INTERPRETATION, CONCLUSIONS) — and persists it via the terminal `write_summary_tool` call. The resulting JSON report is consumed by the Oracle on the next turn and serves as the integrated cross-agent view of the iteration.

*Table 14.* Tools available to the Reporter Agent (REPORTER). The two research tools are running locally, attaining two key functionalities - collecting all the analysis files corresponding to the generated structures from the previous agents, and then using the agent's reasoning to create a summary of the key findings and insights.

| Tool | Source | Purpose |
|---|---|---|
| `collect_summary_files` | Local | Collect specific files from the output folder and return compact statistical extractions of their contents: design metrics and reports from BoltzGen runs, plus all files produced by the Analysis Agents. |
| `write_summary_tool` | Local | *Terminal tool.* Write a structured JSON report containing: header, overview, design comparison (BoltzGen metrics), preprocessing (energy minimisation, hydrogen addition), structural analysis (distogram, Ramachandran, contacts, DSSP, Rosetta scores), sequence composition, cross-tool design synthesis (consensus and disagreements), scientific interpretation, and conclusions. |

### B.2.11. ORACLE AGENT SKILL

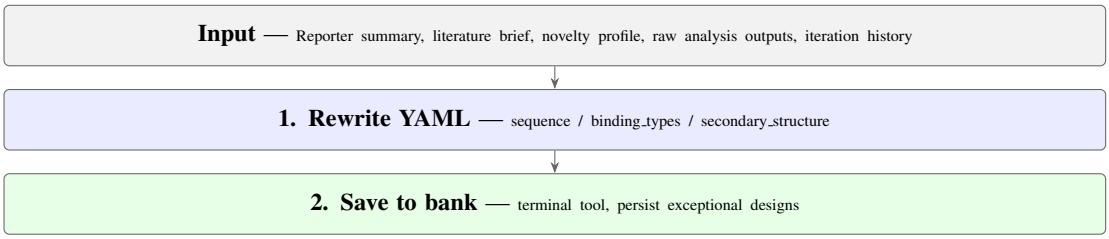



*Figure 6.* **Oracle agent (ORACLE).** A two-tool reviewer that closes the iteration loop. The agent integrates evidence from Reporter's synthesis, Literature Expert (literature brief), Novelty (MMseqs2 novelty profile), the raw Analysis Agents' outputs, and the cross-iteration history, then issues a structured justification (PASSING / KEY FINDINGS / LITERATURE CONTEXT / REPORTER SYNTHESIS / SEQUENCE NOVELTY / STRUCTURAL ANALYSIS / CROSS-ITERATION REASONING / CONFIG CHANGES) and calls `yaml_config_rewrite_tool` to modify exactly three editable fields on the designed-protein entity (`sequence`, `binding_types`, `secondary_structure`) for the next BoltzGen sampling pass. The terminal `save_to_bank_tool` call persists any design meeting the joint-criterion threshold (*e.g.,* dG_separated $< -15$ REU, sc_value $> 0.65$, high iptm/ptm, low binder coil fraction, few Ramachandran outliers) into a cross-run design bank that survives pipeline restarts.

*Table 15.* Tools available to the Oracle Agent (ORACLE). The agent justifies its reasoning in a structured format: passing count, key findings, literature context (from the Literature Expert), Reporter synthesis, sequence novelty (from the Novelty Agent), structural analysis (DSSP, Rosetta, GetContacts, Ramachandran, distogram), cross-iteration reasoning, and proposed config changes — then applies the YAML rewrite tool. Only exceptional designs meeting joint thresholds (dG_separated $< -15$ REU, sc_value $> 0.65$, high iptm/ptm, low binder coil fraction, few Ramachandran outliers) are saved to the bank.

| Tool | Source | Purpose |
|---|---|---|
| `yaml_config_rewrite_tool` | Local | Read a protein-design YAML configuration file and rewrite specific fields (sequence, binding types, secondary structure) on the designed-protein entity. Write the result to a new file by default, or can overwrite the original. |
| `save_to_bank_tool` | Local | *Terminal tool.* Save top-performing design structure files to the bank folder — a persistent collection of the best designs across all pipeline runs, preserved even when the pipeline restarts with a new output folder. |

#### B.2.12. ORCHESTRATOR AGENT SKILL

The Orchestrator Agent does not have any tool available to run as part of the pipeline process. Acting as a central coordinator of the multi-agent protein design and analysis pipeline, its responsibilities can be summarised as follows:

- Manage all the other agents;
- Track passing structures across iterations, and enforce configuration handoff between iterations.

The agent is also able to act the following functionalities: select analysis agents and tools, and decide whether to restart or finalise the workflow. However, for the purpose of this work, we decided to have a deterministic choice - the agent will select all tools and will always restart the process, until the predefined maximum number of iterations is achieved.

## C. Targets

### C.1. Benchmark Targets in the current literature

Previous works, including Stark et al. (2025), Chai Discovery Team et al. (2025), Zambaldi et al. (2024) considered the following molecular targets when presenting the efficiency and capabilities of their models: **PD-L1** (Programmed Death-Ligand 1), **PDGFR** (Platelet-Derived Growth Factor Receptor), **IL-7R**$\alpha$ (Interleukin-7 Receptor-alpha), **TNF**$\alpha$ (Tumor Necrosis Factor-alpha), **InsulinR** (Insulin Receptor). All these targets were chosen due to the availability of existing binders, which were included in the training set of these models. Butcher et al. (2025) added to this list **IL-2R**$\alpha$ (Interleukin-2 Receptor-alpha), **Tie-2** (Angiopoietin-1 Receptor), with Zambaldi et al. (2024) and Latent Labs Team et al. (2025) proposing protein binders for **BHRF1** and **SC2RBD** (SARS-CoV-2 spike protein receptor-binding domain).

### C.2. Targets of our study

We evaluate binder generation on three human protein targets selected to span distinct classes of therapeutically relevant cell-surface proteins: an immune-checkpoint ligand, a growth-factor receptor, and a T-cell signalling component. These proteins are used as representative binder-generation targets with clear biological relevance and established interest in therapeutic protein engineering.

*Table 16.* Targets used for binder generation. UniProtKB accession identifiers and protein names are taken from the corresponding UniProtKB entries.

| Target | Protein | Accession | Brief description |
|---|---|---|---|
| PD-L1 | CD274 | Q9NZQ7 | Cell-surface immune-checkpoint ligand involved in limiting T-cell responses. |
| PDGFR (PDGFR$\beta$) | PDGFRB | P09619 | Cell-surface receptor tyrosine kinase involved in growth-factor signalling, proliferation, migration, and vascular biology. |
| CD3 | CD3D | P04234 | Component of the TCR–CD3 complex on T cells, required for T-cell activation and adaptive immune responses. |

**PD-L1.** Programmed death ligand 1, or PD-L1, is a cell-surface immune-checkpoint protein (UniProtKB Q9NZQ7). Its best-characterised role is binding PD-1 on T cells, which raises the threshold for T-cell activation and limits T-cell effector responses. This pathway is important in normal immune regulation, but it is also widely studied because tumour cells can exploit PD-L1 expression to reduce anti-tumour immune activity (Gong et al., 2018; Jiang et al., 2019).

PD-L1 is included here as a representative extracellular immune-regulatory target. It is biologically well characterised, structurally studied, and clinically prominent because the PD-1/PD-L1 axis is a major target of cancer immunotherapy (Javed et al., 2024).

**PDGFR.** Platelet-derived growth factor receptors are cell-surface receptors that respond to platelet-derived growth factor ligands (UniProtKB P09619). Here, we use PDGFR$\beta$ as the representative PDGFR target. PDGFR is a receptor tyrosine kinase that binds PDGFB and PDGFD homodimers, as well as PDGFA–PDGFB heterodimers, and activates signalling pathways involved in cellular proliferation, migration, survival, vascular support, and tissue remodelling.

PDGFR is included as an example of a therapeutically relevant growth-factor receptor. Abnormal PDGF/PDGFR signalling has been studied in cancer, fibrosis, angiogenesis, and tumour-stroma interactions, making the receptor family a long-standing focus of drug-discovery efforts (Ai et al., 2024). For our purposes, PDGFR provides a different type of surface target from PD-L1: rather than an immune-checkpoint ligand, it is a signalling receptor with an extracellular ligand-binding region suitable for binder generation.

**CD3.** CD3 is part of the T-cell receptor signalling machinery. In this work, we use CD3D, the delta chain of the CD3 complex, as the representative CD3 target. CD3D is a cell-surface protein found as part of the TCR–CD3 complex on T lymphocytes (UniProtKB P04234). When the T-cell receptor recognises antigen, the associated CD3 chains help transmit activation signals into the T cell (Liu et al., 2024).

CD3 is included because it is a central and widely used T-cell surface target in therapeutic protein engineering. For example, many T-cell-engaging bispecific antibodies bind CD3 on T cells and a second antigen on target cells, thereby bringing the two cells into proximity and promoting T-cell-mediated killing (Wu & Cheung, 2018). In this study, however, CD3D is used simply as a biologically important binder-generation target.

## D. Experimental and Evaluation Setup

### D.1. Large Language Model Setup

To power our agentic framework, we made use of a series of large language models, as described in the following table:

*Table 17.* Overview of LLMs used to power the agentic entities.

| Model | Provider | Identifier | #Params | Context |
|---|---|---|---|---|
| GPT-5.4 mini | OpenAI | gpt-5.4-mini-2026-03-17 | — | 400k |
| Claude-Sonnet 4.6 | Anthropic | claude-sonnet-4-6 | — | 1M |
| Llama-3.1-8B-Inst | Meta | Llama-3.1-8B-Instruct | 8B | 128k |
| Qwen-3.5-27B | Qwen | Qwen3.5-27B | 27B | 262k–1M |

For GPT-5.4 MINI, we set the reasoning effort to high, while for CLAUDE-SONNET-4.6, we enable extended thinking with $32,768$ tokens; for both models, we set temperature to $1$ to promote enhanced creativity. For the open source models, QWEN-3.5-27B and LLAMA-3.1-8B-INST, we used the default temperature of $0.6$, in order to balance creativity and precision, given these models have limited capabilities compared to their closed-sourced counterparts. For the closed-source models (GPT and Claude), we set the maximum number of tokens to $65,536$ and use the default decoding strategy; for the open-source models, we set it to $16,384$, and use the greedy decoding strategy.

### D.2. Representative Clustering Strategy

To promote diversity and avoid computing the results on a set of highly similar minibinder structures, we used a *representative clustering* approach. Once the most liked binders are saved (maximum $10$ per iteration), then we group them together with the binders from previous iterations (if available), extract the sequences, and then run the MMseqs2 (Hauser et al., 2016; Steinegger & Söding, 2017) software to assess sequence similarity and create representative clusters. The command used is:

```
mmseqs easy-cluster <fasta_path> <prefix> <tmp_dir> --min-seq-id 0.9 -c 0.9 --cov-mode 0
```

where the *fasta_path, prefix, tmp_dir* are replaced by the corresponding files and folder using a Python script.

In a nutshell, the command basically means that we only place two designs in the same cluster if they are 90% similar in terms of sequence.

### D.3. Compute Resources

Experiments were run on an internal compute cluster using NVIDIA H200 NVL GPUs, each with approximately 150 GB of memory. Each full 20-iteration run for one target used a single GPU. Open-weight models, QWEN-3.5-27B and LLAMA-3.1-8B-INSTRUCT, were hosted locally, with runs taking approximately 29 hours and 11 hours, respectively. Closed-source models, GPT-5.4 MINI and CLAUDE-SONNET-4.6, were accessed through API calls, with the local GPU used for structure generation and analysis; the corresponding 20-iteration runs took approximately 11 hours and 14.5 hours, respectively.

### D.4. Metrics

In this section, we offer a description of the key metrics in our analysis.

*Table 18.* Detailed description of evaluation metrics used to benchmark our pipeline.

| Metric Abbrev. | Full Metric | Direction | Description |
|---|---|---|---|
| ipTM | ipTM | ↑ | Measures the accuracy of the predicted relative positions of the subunits forming the protein-protein complex. |
| PAE | Predicted Aligned Error | ↓ | Measure of confidence in the relative position of two residues within the predicted structure. |
| pTM | pTM | ↑ | Integrated measure of confidence in the prediction of the overall structure of the complex. It is the predicted TM (template modelling) score for a superposition between the predicted structure and the hypothetical true structure. |
| RMSD | Root Mean Squared Deviation | ↓ | Measure of the average distance (or deviation) between corresponding atoms when two similar structures are superimposed. |
| Conf. | Boltz Confidence Score | ↑ | Computed as a weighted average between pLDDT and ipTM, using the formula $0.8 \times \text{pLDDT} + 0.2 \times \text{ipTM}$, as per Passaro et al. (2025). |
| pIdent | Percentage Identity | ↓ | Percentage of identical matches relative to the aligned length, used by the MMseqs2 (Hauser et al., 2016; Steinegger & Söding, 2017) tool. |
| Cov. | Query Coverage | ↓ | Measures the fraction of the query sequence length that is covered by the aligned segments, used by the MMseqs2 (Hauser et al., 2016; Steinegger & Söding, 2017) tool. |

## E. Results for all iterations

In this section we present per-target results across all iterations for each model. For every target we report seven key metrics (ipTM, pTM, PAE, RMSD, pIdent, query coverage, and confidence score) for the top 10 binders from the cumulative most-liked set, together with the distribution of Oracle configuration changes across iterations. Per-target figures follow on the subsequent pages.

### E.1. PD-L1

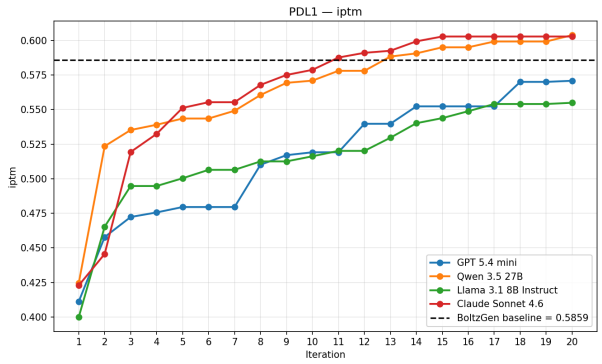
Figure 7 summarises the PD-L1 results across all four models over 20 iterations.

### E.2. PDGFR

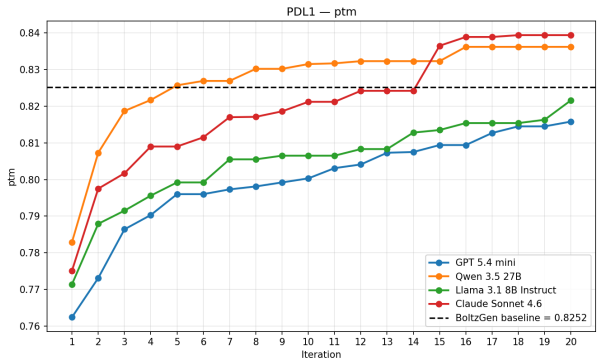
Figure 8 summarises the PDGFR results across all four models over 20 iterations.

### E.3. CD3

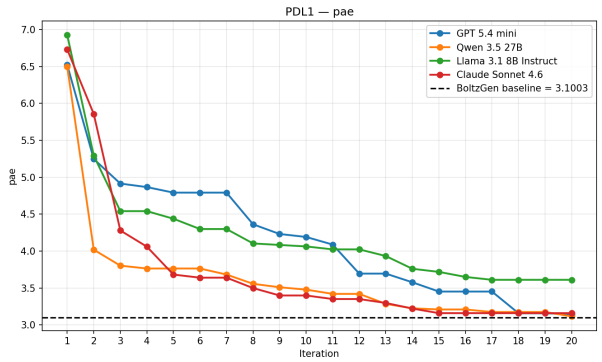
Figure 9 summarises the CD3 results across all four models over 20 iterations.



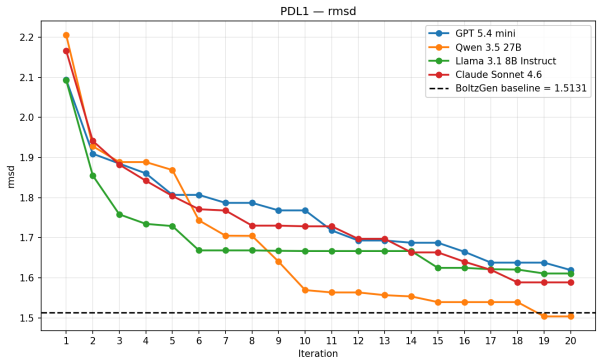

*(a)* ipTM.



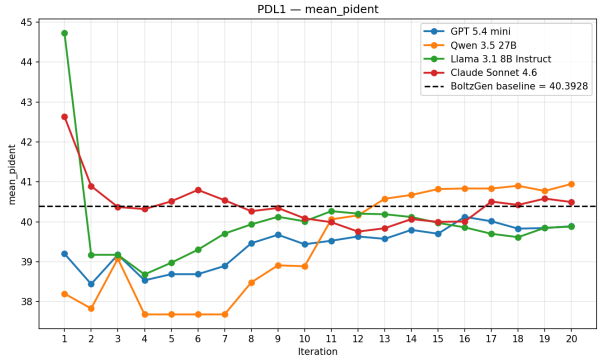

*(b)* pTM.



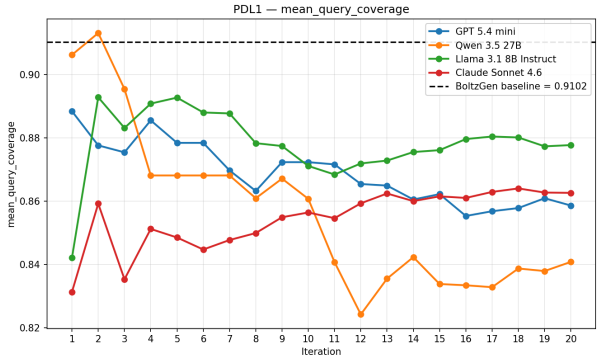

*(c)* PAE.



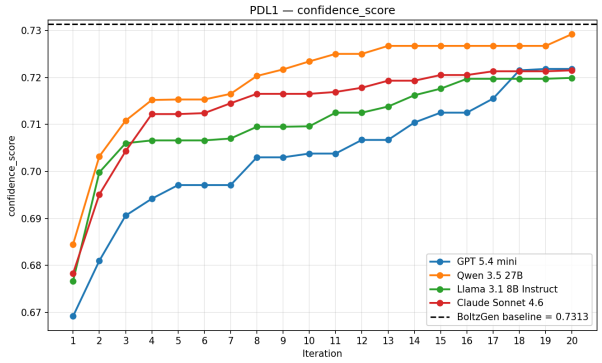

*(d)* RMSD.



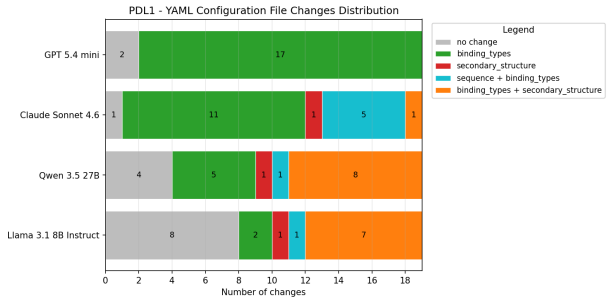

*(e)* pIdent.




*(f)* Query coverage.

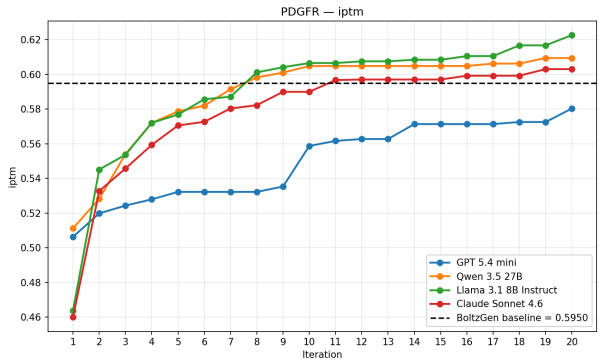



*(g)* Confidence score.

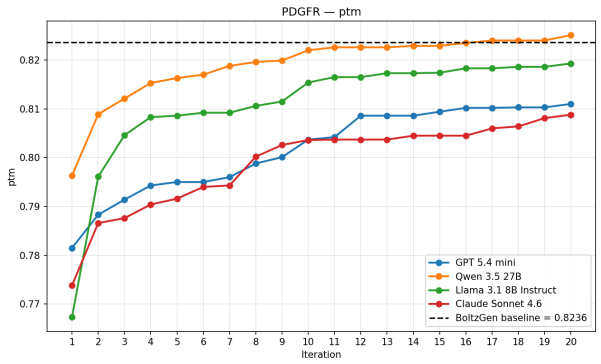



*(h)* Distribution of configuration changes.

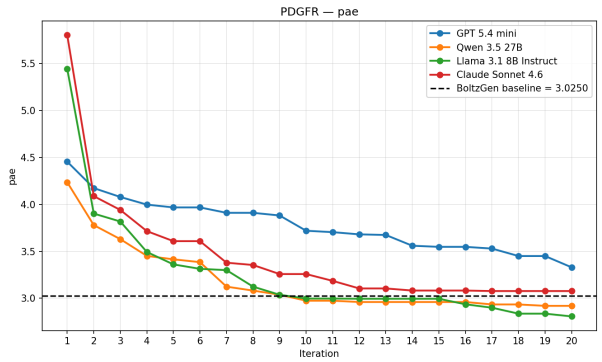
*Figure 7.* PD-L1 metrics for the top 10 binders from the cumulative set of most-liked binders, together with the distribution of configuration changes.



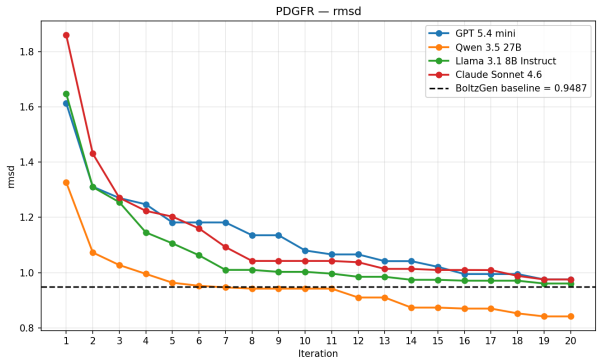

*(a)* ipTM.



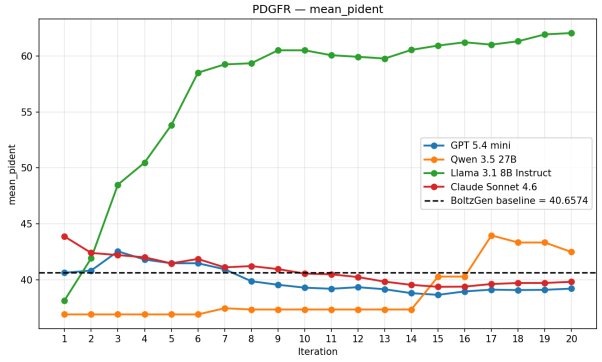

*(b)* pTM.



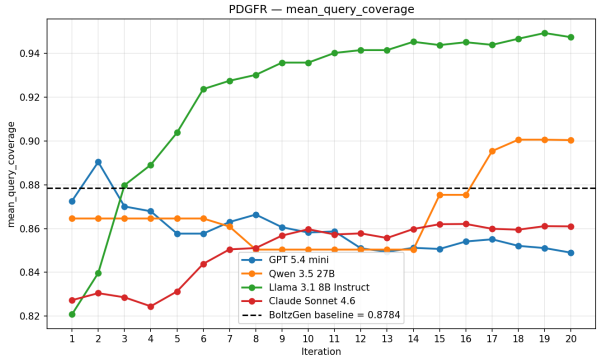

*(c)* PAE.



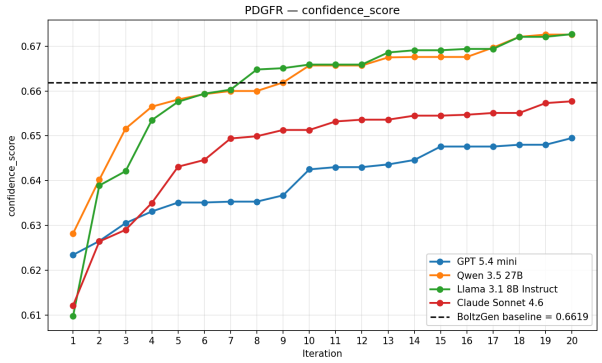

*(d)* RMSD.



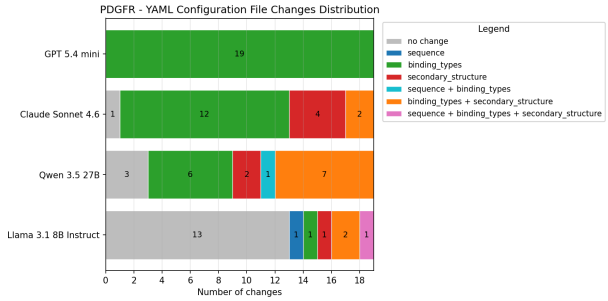

*(e)* pIdent.




*(f)* Query coverage.

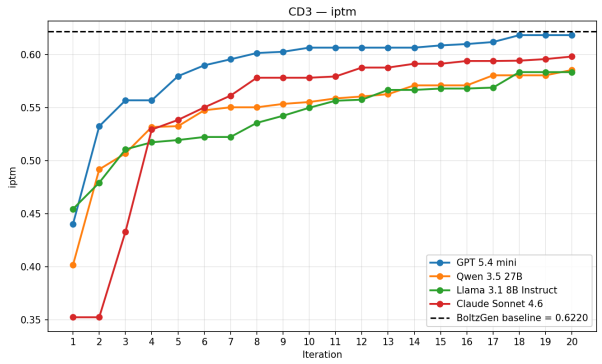



*(g)* Confidence score.

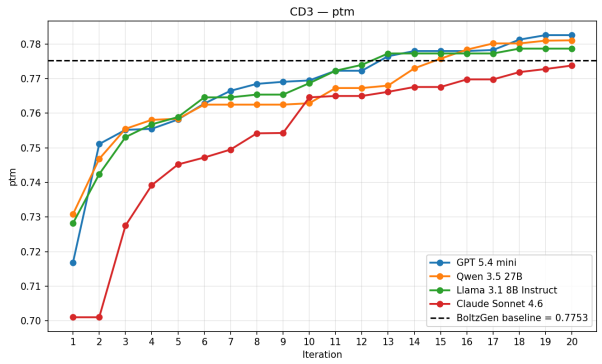



*(h)* Distribution of configuration changes.

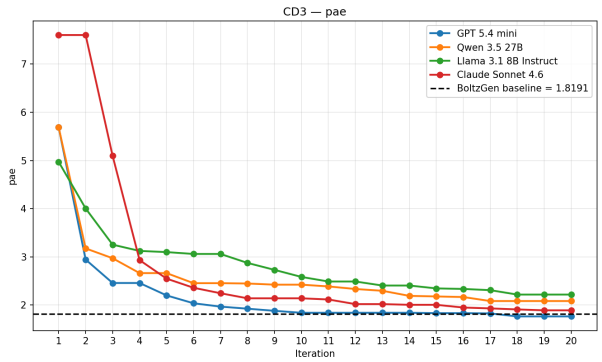
*Figure 8.* PDGFR metrics for the top 10 binders from the cumulative set of most-liked binders, together with the distribution of configuration changes.



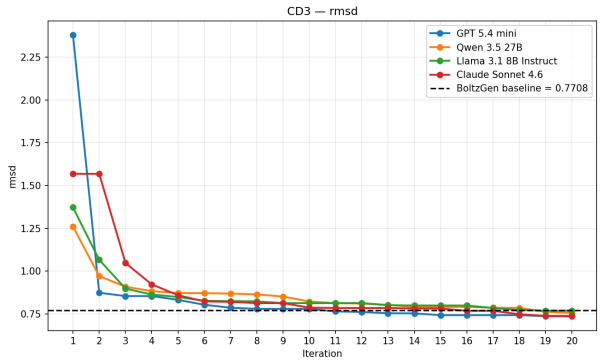

*(a)* ipTM.



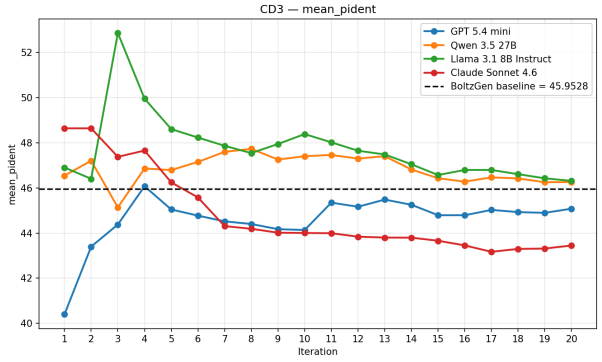

*(b)* pTM.



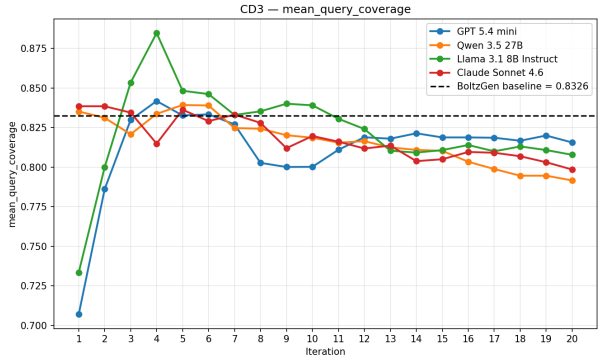

*(c)* PAE.



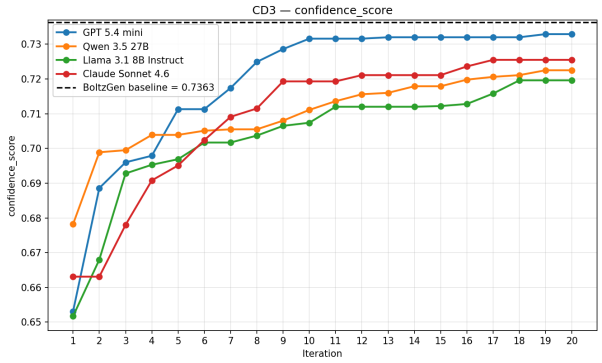

*(d)* RMSD.



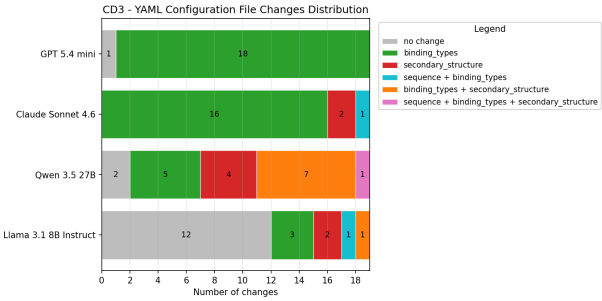

*(e)* pIdent.




*(f)* Query coverage.




*(g)* Confidence score.




*(h)* Distribution of configuration changes.

*Figure 9.* CD3 metrics for the top 10 binders from the cumulative set of most-liked binders, together with the distribution of configuration changes.

## F. Agentic Confidence

In the context of our work, we report agentic confidence in two distinct places of our pipeline: after the Literature Expert collects the relevant information, it will make a suggestion to the Oracle Agent and assess its own confidence; after the Oracle Agent makes its design change suggestions, it will also quantify its own confidence. For simplicity, we ask the agent to use a categorical approach, choosing one of the three available options: low, medium, high. The distribution of the levels of confidence attained by the Literature Expert is shown in Table 19, and for the Oracle Agent is shown in Figure 10.

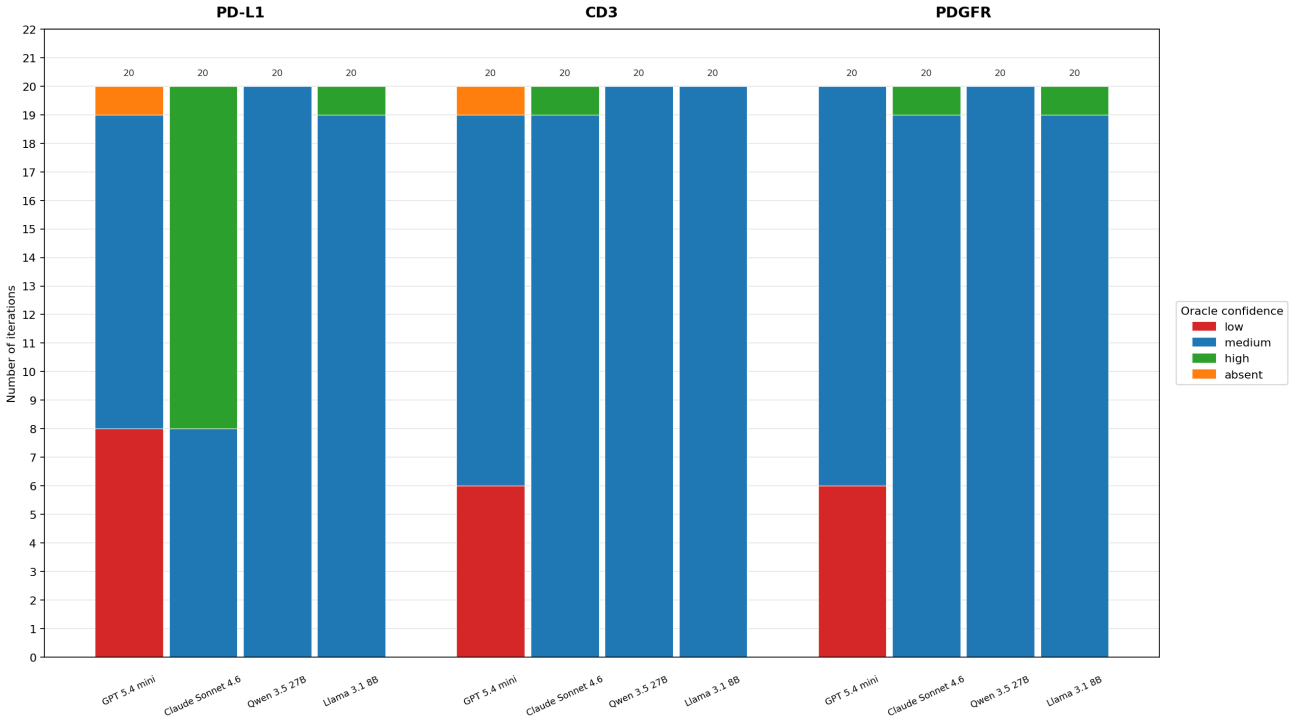



*Figure 10.* Distribution of confidence levels for all targets and models for suggestions provided by the Oracle Agent. Qwen consistently maintains medium confidence, a trend approximately followed by Llama. GPT adopts a more cautious approach, reporting only low and medium confidence in its suggestions. Claude, especially for PD-L1, tends toward high confidence initially, then settles to moderate confidence.

*Table 19.* Confidence levels for suggestions made by the Literature Expert across all targets and models. Open-source models tend toward lower confidence: Llama is consistently underconfident, while Qwen maintains medium confidence except for CD3. GPT and Claude perform identically — highly confident for the well-studied PD-L1 target, medium otherwise.

| Target | GPT-5.4 mini | Claude-Sonnet-4.6 | Qwen3.5-27B | Llama-3.1-8B-Instruct |
|---|---|---|---|---|
| PD-L1 | High | High | Medium | Low |
| CD3 | Medium | Medium | Low | Low |
| PDGFR$\beta$ | Medium | Medium | Medium | Low |

# G. Agentic Reasoning Traces

## G.1. Changes and non-changes for CD3 Target

*Table 20.* Changes and non-changes proposed by the Minibinder Lab, empowered by CLAUDE SONNET 4.6, for CD3. For iteration **8***, we provide complete reasoning traces for the justification on changes and non-changes in Figure 11.

| Iteration | Changes | Not changes |
|---|---|---|
| 1 | `binding_types`: none → `binding=43,45,47,49,80,82,84`, `not_binding=86,87,88,89,90,91,92,93`, on chain B. `sequence`: [prior range] → `85..105`. | `secondary_structure` unchanged. |
| 4 | `binding_types.binding`: `'30, 35, 40, 43, 44, 45, 46, 47, 48, 49, 80, 81, 82, 83, 84, 85'` (16 residues) → `'43, 45, 47, 49, 80, 81, 82, 83, 84, 85'` (10 residues) | `sequence` unchanged, `secondary_structure` unchanged. |
| 8 * | `binding_types.binding`: `23,27,31,43,45,47,49,80,82,84` (10 residues) → `43,45,47,49,80,82,84` (7 residues, BC/FG loops only) | `sequence` unchanged, `not_binding` unchanged, `secondary_structure` unchanged. |
| 12 | `binding_types.binding`: `[25, 27, 30, 43, 45, 47, 49, 80, 82, 84]` → `[43, 45, 47, 49, 80, 82, 84, 85]` | `sequence` unchanged, `secondary_structure` unchanged, `not_binding` unchanged. |
| 16 | `binding_types.binding`: `43,45,47,49,80,82,84,85` → `27,35,43,45,47,49,80,82,84,85` (added N-terminal IgV entry residues 27 and 35) | `sequence` unchanged, `secondary_structure` unchanged. |
| 20 | `binding_types.binding`: `'43, 45, 47, 49, 80, 82, 84, 85'` → `'30, 35, 43, 45, 47, 49, 80, 82, 84, 85'` | `sequence` unchanged, `not_binding` unchanged, `secondary_structure` unchanged. |

## G.2. Justification on changes for CD3 Target, Iteration 8

Figure 11 shows the complete justification offered by the agent for the changes and non-changes applied at iteration 8 from Table 20 in Appendix G.1.

## G.3. Complete Reasoning Trace - PD-L1 Target, GPT 5.4 mini, Iteration 19

Figures 12, 13, 14 provide an example of a complete reasoning trace for the PD-L1 target obtained by using the GPT 5.4 mini model, and exemplified for iteration 19. For Figure 12: Green colour highlights good reasoning where the agent correctly applies literature constraints - recognising that Literature Expert (A10) justifies avoiding the C2 domain (133–225) but does *not* justify hard-excluding residues 82–100 where successful designs naturally bind. This demonstrates proper structural evidence-based constraint logic. For Figure 14: Green highlights correct reasoning - recognising that the Literature Expert (A10) supports C2 domain avoidance but not hard-excluding 82–100. Red highlights problematic reasoning: selective metric emphasis—touting iptm/PAE improvements while glossing over the 6% pass-rate drop and weak mean binding energies ($\approx$ 42.6 REU) and over-reliance on epitope-mask tweaking while refusing to test sequence range or secondary-structure constraints despite persistent VDW-dominance and mediocre interfaces. Orange highlights mixed logic: correctly identifying that designs contact 83–96 flank, but not rigorously interrogating whether rank004, rank005, rank006 (which are mediocre quality) justify the constraint change. Recurrence alone is insufficient without evidence of good engagement.

## CHANGES

### Binding Site Refinement

`binding_types.binding`: `23,27,31,43,45,47,49,80,82,84` (10 residues) → `43,45,47,49,80,82,84` (7 residues, BC/FG loops only)

### Justification

Cross-iteration evidence (iterations 6–8) reveals a persistent 4-pass depression correlated with N-terminal CD3D residues 23, 27, 31. Iteration 6 (BC/FG only): 18/100. Iteration 7 (with 23/27/31): 12/100. Iteration 8 (same): 14/100.

*Structural rationale:* The 10-residue constraint spanning residues 23–84 (61-residue span) creates geometrically demanding requirements for a ∼95-aa binder. The A3 distogram max Cα distance (43–47 Å) cannot simultaneously reach the N-terminal loop and FG loop.

Per A10 (PMID 15534202), the BC/FG loop surface (43–49, 80–85) is **PRIMARY** high-confidence epitope, while N-terminal entry (23–42) is **SECONDARY** with lower confidence.

### Expected Outcome

**Pass rate: +4% (14% → ∼18%)** — recovering toward campaign best (iteration 6) by refocusing on highest-confidence productive epitope.

## NOT CHANGING

### Sequence (85..105)

Binder lengths of 92–102 aa (top-10 by A3 distogram, A5 atom counts) produce appropriately compact IgV-like folds with strong Rosetta interfaces ($dSASA_{int}$ 1629–1953 $Å^2$).

A3 max Cα distances (43–47 Å) confirm appropriate compactness. No terminal coil excess identified from available DSSP. No Ramachandran outlier clustering beyond universal C-terminal capping artifact.

### not_binding (86..116)

Constraint maintained to target entire dimer-interface through TM domain span. Per A10 (PMID 15534202), prevents binders from disrupting the obligate CD3-$\delta/\epsilon$ heterodimer required for TCR-CD3 complex integrity.

Chain B footprints (48–68 residues engaged per GetContacts) within extracellular domain range. No evidence of systematic dimer-interface failures in passing designs.

### secondary_structure

Per-chain DSSP coil fractions unavailable from A6 output. All designs show appropriate IgV-like antiparallel $\beta$ topology (79–87/100 residues with antiparallel H-bond patterns per A6).

Adding SS constraints without satisfying the evidentiary threshold risks over-constraining an already appropriate $\beta$ fold. Not changing.

*Figure 11.* **Iteration 8 Design Decisions: Change Rationale and Preserved Constraints.** *Left:* Binding site refinement focusing the 10-residue constraint on the primary BC/FG epitope surface, justified by cross-iteration evidence and structural geometry. *Right:* Three non-changing design parameters maintained to preserve fold quality, dimer-interface steering, and secondary structure appropriateness. All decisions grounded in structural analysis (e.g., distogram, Rosetta scoring) and literature validation (PMID 15534202, 1XIW).

# Iteration 19→20: Overview, Key Findings & Assessment

## OVERVIEW

PASSING: 20/100 structures passed BoltzGen filters. The 3-pass drop from iter18 (23/100) is explicitly below the orchestrator noise floor ($|\Delta| = 3 < 2 \cdot \sqrt{23} \approx 9.6$), so not overfitting to noise. Relative to best-pass iter5 (26/100), current top-N is structurally better on binding-confidence side, so keeping quality direction and only relaxing one over-constraining negative.

## KEY FINDINGS

**1. Pass Rate Flat:** iter18 had 23% pass rate, iter19 has 20%. Delta is noise, not regime change.

**2. Metrics (iter18 → iter19):**

- design_to_target_iptm: 0.4305 → 0.3709 (0.0595 vs iter18, but +0.0242 vs best-pass iter5 at 0.3467)
- min_design_to_target_pae: 6.5880 → 7.7817 (+1.1937 worse vs iter18, yet 0.8292 better over iter5's 8.6109)
- design_ptm: 0.7680 → 0.7672 (flat, 0.0008)
- filter_rmsd: 2.0595 → 2.1264 (+0.0670 worse)
- Top-N Rosetta dG_separated: rank001 at 57.295601 REU; top-10 mean ≈ 42.62 REU (strongly favorable)

**3. Constraint Analysis:** Hotspot emphasis shifted from iter5 (68/76 pair) to current 61/63/66-focused front-face while retaining 52/54/56/58 and 115/123/124 anchors. That shift did not produce trustworthy pass-rate gain. Best explanation for wobble: over-aggressive negative exclusion around 82..100, which is exactly where several successful top designs bind.

## LITERATURE CONTEXT (A10 Brief)

Human PD-L1 (CD274) is type I single-pass membrane checkpoint ligand with extracellular IgV domain (19–127) and Ig-like C2 domain (133–225). Productive blocking face is PD-1/S79 front-face epitope on IgV, with published contacts at residues 18–21, 47–68, 113–124 (PDB 8RPB, PMID 8RPB). Independent hotspot study (PMID 41218053) highlights Y56, M115, Y123, K124. Current top designs partially aligned: rank001 and rank003 hit 47–56 band (SER50/ASN53/SER56); rank004, rank005, rank006 repeatedly contact 83–96 flank. **A10 only tells us avoid C2 domain (133–225); does *not* justify hard-excluding 82–100**, especially when designs already exploit that band.

## STRUCTURAL ASSESSMENT (A8 Synthesis)

rank001 flagged as best binder: dG_separated 57.3 REU, contacts 111 total (22 directional), dSASA_int 1948.6 Å². rank005 weakest top-10: dG_separated 37.8 REU, dSASA_int 1343.0 Å², sc_value 0.609. rank003 most concerning: low contact density (57 total; 8 directional), highest delta_unsatHbonds (17), known_scaffold (62.80% identity, E=7.2e-30). rank007 is no_hit sequence, treat cautiously.
Batch is VDW-dominated (vdw 47–82 across designs, HBBB only once). Rosetta total_score not monotonic with dG_separated (rank004 total_score 488.5 despite dG_separated 39.2), so should not use total_score alone to steer next round.

*Figure 12.* Iteration 19→20: Overview and Assessment. Pass rate stability analysis (20% pass, flat vs iter18, within noise), metric evolution from iter5 to iter19 (iptm +0.024, PAE 0.829), literature context from Literature Expert (A10) establishing PD-1/S79 front-face as primary epitope and C2 domain as avoidance target, and structural assessment of top designs via Reporter (A8). **Color Highlighting (based on domain experts' comments on the reasoning trace).** **Green** highlights good reasoning where the agent correctly applies literature constraints—specifically.

# Iteration 19→20: Structural Analysis

## SEQUENCE NOVELTY (A11 MMseqs2)

Batch summary: 1 known_scaffold, 5 moderate_homology, 3 weak_signal, 1 no_hit. Healthy regime overall.

- **rank001**: moderate_homology, plaus=high, nov=moderate, cov=0.88, id=41.40%, E=7.6e-09 [prioritize]
- **rank002**: moderate_homology, plaus=high, nov=moderate, cov=0.97, id=36.80%, E=5.8e-08 [prioritize]
- **rank003**: known_scaffold, plaus=high, nov=low, cov=0.97, id=62.80%, E=7.2e-30 [deprioritize]
- **rank004**: moderate_homology, plaus=high, nov=moderate, cov=0.91, id=36.10%, E=2.8e-06 [prioritize]
- **rank005**: weak_signal, plaus=moderate, nov=high, cov=0.91, id=35.70%, E=5.0e-04 [conditional]
- **rank006**: moderate_homology, plaus=high, nov=moderate, cov=0.78, id=36.90%, E=1.5e-07 [prioritize]
- **rank007**: no_hit, plaus=uncertain, nov=uncertain [extra scrutiny]
- **rank008**: weak_signal, plaus=moderate, nov=high, cov=0.77, id=36.80%, E=1.0e-04 [conditional]
- **rank009**: weak_signal, plaus=moderate, nov=high, cov=0.93, id=36.10%, E=1.1e-05 [conditional]
- **rank010**: moderate_homology, plaus=high, nov=moderate, cov=0.99, id=37.80%, E=4.6e-09 [prioritize]

## STRUCTURAL ANALYSIS (Per Agent, Actual Numbers)

**A3 (Distogram):** rank001 214 aa, max_dist 63.77 Å; rank002 225 aa, 66.90 Å; rank003 229 aa, 71.25 Å; rank004 227 aa, 55.25 Å; rank005 214 aa, 72.25 Å; rank006 225 aa, 67.52 Å. Tight 214–229 band, no collapse signal.

**A4 (Ramachandran):** rank001–rank006 all 0 outliers (0.00%), no terminal pile-up. Argues against length/geometry rescue.

**A5 (GetContacts — VDW-dominated):** rank001: 111 contacts (22 dir, 82 VDW); rank002: 97 (22 dir, 74 VDW); rank003: 57 (8 dir, 47 VDW); rank004/rank005/rank006 use 83–96 flank with 91, 68, 103 total contacts respectively.

**A6 (DSSP):** Coil fractions 34.4–41.5% (mixed, not terminal). All show IgV-like $\beta$ topology.

**A7 (Rosetta):** rank001 dG_sep 57.3, dSASA 1948.6, sc 0.672; rank002 dG_sep 46.3, dSASA 1711.2, sc 0.678; rank003 dG_sep 37.8, dSASA 1669.2, sc 0.654; rank004 dG_sep 39.2, dSASA 1691.4, sc 0.620; rank005 dG_sep 37.8, dSASA 1343.0, sc 0.609; rank006 dG_sep 44.3, dSASA 1779.5, sc 0.605.

**A2 Preprocessing:** Hydrogen addition terminal-capping artifact, minimisation succeeded. Not reason to shorten chain.

*Figure 13.* Iteration 19→20 structural analysis. Sequence novelty assessment (A11) showing batch composition and per-rank prioritization. Detailed per-agent structural metrics (A3–A7) confirm tight length band, clean Ramachandran, VDW-dominated contacts, and moderate Rosetta energetics.

# Iteration 19→20: Configuration Decision

## CROSS-ITERATION REASONING

**Iter 5:** 26% pass, iptm=0.3467, PAE=8.6109, hotspots 52,54,56,58,68,76,115,123,124.
**Iter 19:** 20% pass, iptm=0.3709, PAE=7.7817, hotspots 52,54,56,58,61,63,66,115,123,124; not_binding 24..46,82..100,133..225.
**Interpretation:** **Current hotspot set improved binding-confidence metrics (iptm +0.0242, PAE 0.8292)** but pass rate fell from 26% to 20%. Most plausible cause: over-tight 82..100 exclusion, not positive epitope hypothesis. Keep quality-improving hotspots, relax conflicting negative band.

## CONFIG CHANGES FOR ITER 20

**CHANGE 1 — binding_types: Relax not_binding**
Current: `not_binding: 24..46, 82..100, 133..225`
Proposed: `not_binding: 24..46, 133..225`

**WHY:** A5 shows **rank004/rank005/rank006 repeatedly contact 83–96 flank**, yet current 82..100 not_binding suppresses exactly that. **A10 supports avoiding C2 domain, not hard-excluding 82–100**. Keeping positive hotspots preserves quality-improving front-face. Removing over-aggressive exclusion lets sampler recover productive flank modes.

**EXPECTED:** Pass rate +1–3%; modest dG_separated and iptm improvement by enabling coherent front-face interface.

**NOT CHANGING**
**sequence: 90..110** — A3 lengths 214–229 aa, max_dist 55.2–72.2 Å show no collapse. A4 has 0 Ramachandran outliers. OpenMM warnings are terminal artifacts.
**secondary_structure: {}** — A6 shows coil 34.4–41.5%, mixed not terminal. No strained segment to constrain.

*Figure 14.* Iteration 19→20 configuration decision. Cross-iteration comparison establishing that current hotspot set improved binding confidence but pass rate fell, and the decision to relax the 82–100 exclusion. **Color Highlighting (based on domain experts' comments on the reasoning trace).** **Green** highlights correct reasoning: agent recognises Literature Expert (A10) supports C2 domain avoidance but not hard-excluding 82–100. **Red** highlights problematic reasoning: (1) selective metric emphasis—touting iptm/PAE improvements while glossing over the 6% pass-rate drop and weak mean binding energies ( 42.6 REU); (2) over-reliance on epitope-mask tweaking while refusing to test sequence range or secondary-structure constraints despite persistent VDW-dominance and mediocre interfaces. **Orange** highlights mixed logic: agent correctly identifies that designs contact 83–96 flank, but does not rigorously interrogate whether rank004/rank005/rank006 (which are mediocre quality) justify the constraint change. Recurrence alone is insufficient without evidence of high-quality engagement.

## H. Agentic Configuration Failure

The domain-expert evaluation highlighted several recurring failure modes in how the Oracle Agent converts protein-analysis outputs into configuration updates. These failure modes do not negate the evidence-grounded behaviour described in the main text, but they indicate that the current workflow is not yet a fully controlled experimental planner. The main issues concern oscillating hotspot hypotheses, inconsistent use of binding exclusions, lack of controlled repeats, and limited retrospective calibration.

**Oscillating hotspot hypotheses.** A recurring weakness of the current workflow is that the Oracle can oscillate between different hotspot hypotheses without clear convergence. The agent repeatedly rewrites the `binding_types` and `not_binding` fields of the configuration file, sometimes reverting to earlier configurations before drifting again in later iterations. In practice, this can resemble a random walk through a combinatorial space of possible residue-level constraints, rather than a controlled refinement strategy with a clear stopping criterion. By later iterations, the Oracle may still propose small single-residue changes, even when such changes are unlikely to produce statistically detectable improvements given the design noise floor that the agent itself has identified. We show an example of the changes and non-changes provided by the Minibinder Lab in Appendix G.1.

**Inconsistent use of binding exclusions.** Relatedly, experts noted that the Oracle often uses the `not_binding` option in the YAML configuration, but that the choice of excluded regions can change substantially between iterations. Such an example is provided in Table 21. The not_binding block from 600..962 that corresponds to the intracellular domain is physically justified from the literature, so it keeps resurfacing whenever the agent gets a lower signal from the other tools. Furthermore, the agent aggressively blocks regions after observing a small number of designs drifting towards them, but later unblocks the same or similar regions when the metrics do not improve. This behaviour suggests that the Oracle can identify plausible non-productive surfaces, but does not yet maintain a sufficiently stable hypothesis about which exclusions should be retained across iterations.

*Table 21.* Configuration failure flagged by domain experts, for reasoning traces obtained for the PDGFR target, using GPT-5.4 mini model.

| Iteration | Non-binding residues recommended at previous iteration by the Oracle Agent |
|---|---|
| 5 | `1..5, 26..28, 92..95, 600..962` |
| 9 | `1..4, 26..39, 57..65, 92..95, 600..962` |
| 10 | `1..6, 26..39, 47..49, 57..67, 74..85, 92..95, 600..962` |
| 13 | `26..39, 47..49, 57..67, 74..85, 92..166` |
| 17 | `600..962` |
| 20 | `2, 92, 96, 600..962` |

**Lack of controlled repeats.** The experts also noted that the Oracle rarely decides to hold the configuration fixed for several iterations to estimate stochastic variability. For example, in the PDGFR run, the hotspot set is modified at almost every iteration. Although frequent updates are not necessarily undesirable, they make it difficult to determine whether observed metric changes arise from the configuration update itself or from stochastic variation in generated samples. A more controlled strategy would occasionally repeat the same configuration, or test a single-variable change against a matched control, before committing to further edits.

**Limited retrospective calibration.** At each iteration, the Oracle includes an `EXPECTED` block with quantitative predictions about how its proposed changes should improve the next round of generated binders. It also reports cross-iteration reasoning. However, it does not systematically compare these predictions with the observed outcomes in subsequent iterations. This prevents the agent from recalibrating its confidence, identifying which types of recommendations were reliable, and avoiding repeated unsuccessful behaviours. Future versions of Minibinder Lab would benefit from explicit prediction-versus-outcome tracking, allowing the Oracle to learn from its own intervention history rather than only summarising it qualitatively.

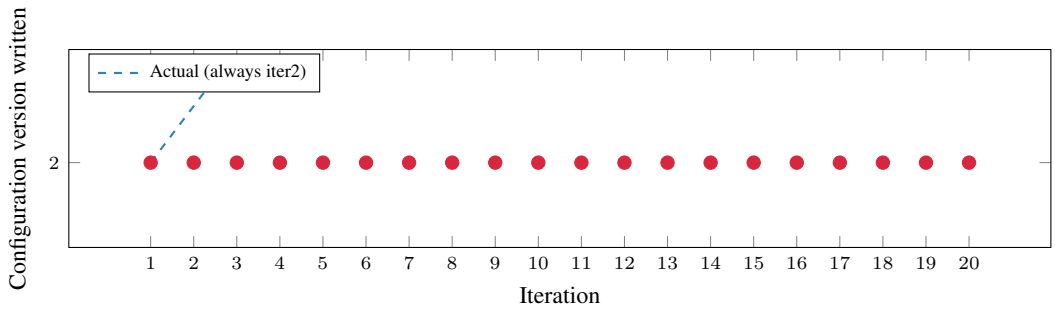
Together, these observations suggest that the Oracle can reason over multi-source protein data, but lacks several behaviours expected from a more disciplined experimental planner. It can identify plausible design problems and propose targeted

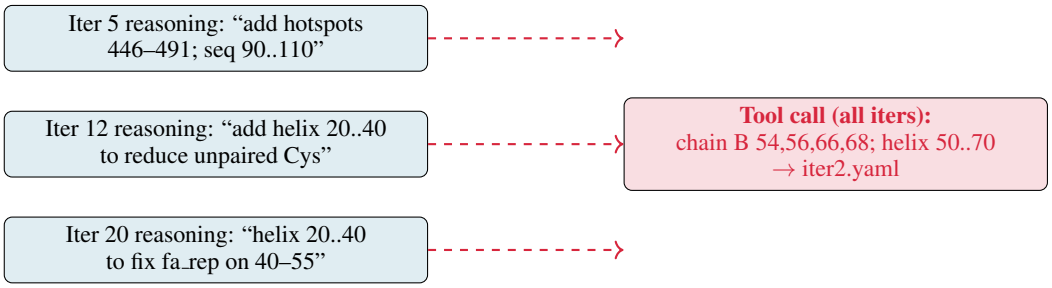
configuration edits, but does not yet robustly track hypotheses, run controlled interventions, or recalibrate its confidence based on previous outcomes. Addressing these failure modes is likely to be important for making agentic binder-design workflows more robust and scientifically reliable.

## I. Failure Analysis - Llama-3.1-8B-Instruct Case Study

**Failure 1: Frozen tool call (configuration never updates).** The Oracle emits the *same* raw JSON tool call at every iteration. Across all 20 rounds the `output_file`, `binding_types`, and `change_summary` fields are **byte-identical**, always targeting `cd3_clean_config_iter2.yaml` (see table below).

- **Evidence:** every iteration writes `output_file = ...iter2.yaml` and `change_summary = "binding_types: chain B 54,56,66,68; helix 50..70"`.
- **Consequence:** the pipeline re-runs the *same* configuration each round; no exploration of the design space occurs.




**Failure 2: Reasoning–action dissociation.** Oracle's free-text reasoning correctly identifies new residues, structural constraints, and metric regressions – but the actual tool call ignores all of it.

- **Evidence:** reasoning proposes chain B hotspots 446–491, helix 20..40, and sequence range 90..110 at various iterations, yet the tool call always emits `chain B 54,56,66,68` and `helix 50..70`.
- **Consequence:** the feedback loop is broken; the Oracle "thinks" it is steering but never actuates a change.