

Proofs, Programs, and Molecules: Comparing Discovery Across Domains

Which mechanisms are shared, and where does the analogy fail?

Complexity, Information, and Cross-Domain Transfer

A common process with different warrants

Each case generates candidates, obtains feedback and establishes a claim. The scarce step can be exact checking, experimental measurement or interpretation.

We first compare the evidence these tasks require, then ask when a fixed workflow needs adaptive decisions or tool use.

The comparison matrix

Domain	Search object	Main validation
Mathematics	Construction or proof	Exact checking / proof
Programs	Executable algorithm	Specification and performance
Molecules	Physical design	Experimental property/function
Mechanisms	Causal explanation	Intervention discrimination

The strongest analogy is at the search-loop level. The sharpest difference is the warrant supplied by evaluation.

Cap-set construction: the precise target

Find a large subset S of F_3^n with no nontrivial triple $x+y+z=0$. Larger $|S|$ improves a lower bound on the extremal size.

The dimension n , representation and checker precision must accompany any reported improvement.

Construction, certificate and theorem are different outputs

A candidate list can be checked directly. A compact construction rule may generate many cases. A proof can establish the rule for every case in a stated family.

The progression increases scope only when each new claim is justified.

FunSearch: a program defines a construction strategy

A language model modifies program components; evaluation selects useful programs. The resulting program can encode structured candidates rather than a raw list.

Interpretability and reuse depend on what the program actually expresses, not simply on the fact that it is code.

Automated evaluators enable broader construction searches

AlphaEvolve extends evolutionary coding across algorithmic and mathematical tasks with automated evaluators.

Georgiev and colleagues study 67 mathematical problems, reporting rediscoveries and improvements. Selected tasks and property-specific checks limit generalization.

Proof search: a different feedback topology

A proof assistant exposes partial proof states, legal transitions and remaining goals. A successful terminal state certifies the formal statement.

This supports feedback at intermediate steps; a whole-candidate score is not the only possible interface.

What a formal kernel certifies

A small trusted checker verifies a derivation under an explicit logical foundation and library.

It does not certify that the informal question was translated correctly, that assumptions are appropriate, or that the theorem is scientifically significant.

Program synthesis: specifying the target is central

A synthesized program must satisfy a specification. Example-based specifications leave multiple compatible programs; logical specifications can impose universal constraints.

Runtime optimization adds an objective to correctness, often under hardware- and workload-specific conditions.

A bounded program example can be fully checked

For integer inputs $x \in \{0, \dots, 255\}$, compare a candidate program with the specification x^2 on all 256 inputs, assuming termination.

Passing all tests proves equivalence on that finite domain; it says nothing about negative inputs or larger integers.

Performance measurements add a statistical layer

A kernel can be functionally correct yet have variable measured runtime because of warm-up, hardware contention or input shape.

A credible speedup specifies the workload distribution, device, baseline, measurement protocol and uncertainty.

Molecular design adds a physical object

A sequence or graph must become a physical object in an experimental context.
Structure and function are related but not identical.

A computationally favorable candidate can fail because of stability, expression, chemistry or environmental interactions.

RFdiffusion: generation plus filtering plus experiments

The RFdiffusion study connects generated structures to experimental tests across reported design tasks. Its claims rely on the measured endpoint and selected targets.

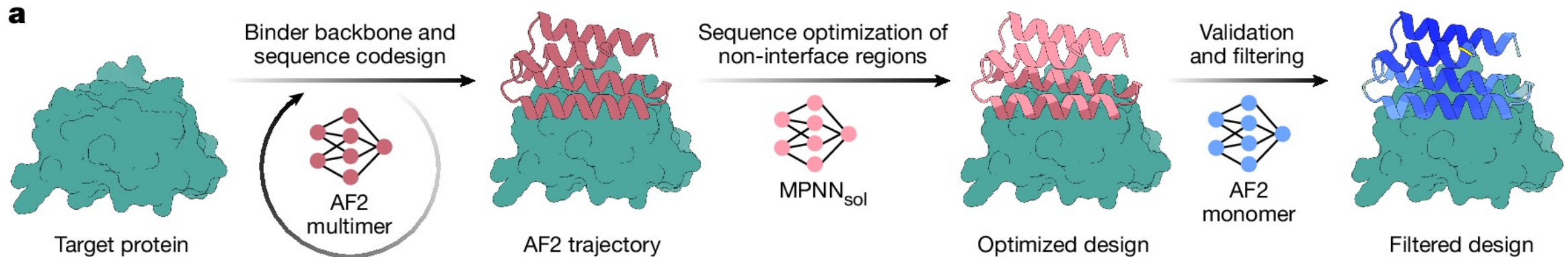
A structural generator can reduce the number of experiments needed by enriching the pool; it does not make experimentation logically unnecessary.

BindCraft: binding and functional outcomes remain separate

BindCraft reports computational design and experimental evidence, including selected functional uses of designed binders.

A binding result is stronger than a prediction, but additional function and selectivity claims require their own evidence.

BindCraft: computational selection precedes binding evidence



AlphaFold 2 (AF2) guides backbone and sequence design. A message-passing neural network (MPNN) optimizes selected sequence regions; another structural prediction filters candidates before experimental testing.

Binding success needs a target and denominator

Target	Binding observed / tested	Fraction
PD-1	13 / 53	24.5%
PD-L1	7 / 9	77.8%
CD45	4 / 16	25.0%

Selected targets from Figure 1b. These are measured binding counts among designs tested, not success probabilities for arbitrary proteins or a randomized comparison of design methods.

Drug-binding protein design exposes coupled constraints

The 2026 neural iterative selection–expansion work jointly navigates sequence, protein geometry and ligand conformation.

The scientific outcome is experimentally supported binding in studied designs, not just an improved internal structural score.

A validated binder can become a scientific instrument

A selective binder can enable perturbation, localization or measurement. It can therefore change the information available for a later mechanism question.

Its usefulness as an instrument depends on behavior in the new context, not only on its original binding measurement.

Resistance: two mechanisms, one observed endpoint

An unstable resistant fraction might be an assay problem. But descendants share a past: an early heritable event could amplify one lineage before exposure.

That connection makes growth history worth varying. The next comparison reconstructs plausible design choices; the following experiment reports Luria and Delbrück's evidence.

Build a comparison that changes history, not the assay

Candidate measurement	What it can reveal
Plate one culture repeatedly	Variation introduced by the terminal assay
Grow independent cultures	Variation accumulated before exposure
Match growth and exposure	Whether unequal conditions explain the contrast

Lineage reasoning suggests separating growth histories while keeping terminal measurement comparable. Other checks remain necessary.

The discriminating signal is a jackpot

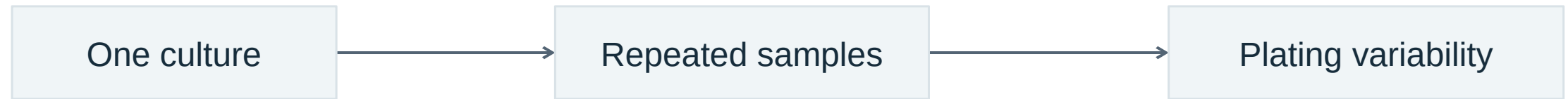
Late event: 1 descendant; early event: $2^4=16$ descendants

Under independent induction at exposure, similar cultures give roughly Poisson counts: variance near the mean.

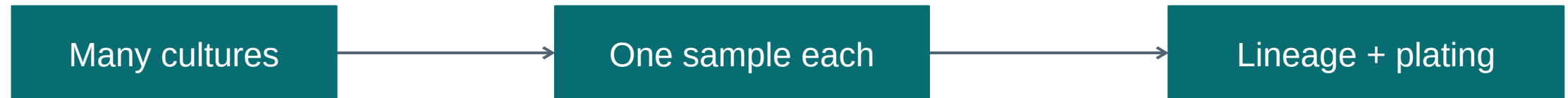
Under pre-existing heritable variation, an early event leaves many descendants. Independent cultures show rare jackpots and variance far above the mean.

A control separates culture history from assay noise

CONTROL



DISCRIMINATING TEST



Control: repeatedly sample one culture to measure sampling and plating variability. Test: grow many independent cultures from small inocula, then plate with excess virus. The study checked growth properties and rapid lysis, reducing explanations based on unequal growth or prolonged post-plating multiplication.

Independent cultures fluctuate far beyond Poisson

Experiment	Cultures	Mean per sample	Corrected variance
1	9	26.8	1,217
11	10	62	3,498
16	20	11.35	694

Selected Table 2 experiments report the following resistant-colony means and sampling-corrected variances. These are culture-level summaries, not independent colony replicates.

Mechanism case: why are phage particles tailless?

Tai is a bacterial antiviral defense studied by He and colleagues. Observing tailless particles leaves two explanations: tails never form, or formed tails fail to attach.

The visible outcome alone cannot distinguish those mechanisms. A decisive observation must separate free-tail formation from attachment.

A control separates formation from attachment

The study examined a mutant background that yields separate procapsids and tails. This makes free tails observable independently of successful attachment.

Prediction under blocked attachment: free tails remain. Prediction under blocked formation: free tails disappear.

In the reported TaiEcA condition, microscopy found capsids without tails, supporting blocked formation.

What this mechanism result leaves unresolved

Tail loss supports a formation defect. It does not by itself establish direct binding to a specific protein or exclude mediation by a host factor.

The authors could not test the proposed interaction successfully because protein J was insoluble in the pull-down setup. Figure 4 microscopy used one replicate per experiment.

Hypothesis generation, discriminating evidence and direct molecular mechanism are separate achievements.

When does a discovery workflow need an agent?

If candidate generation becomes cheap while experiments remain slow, simply producing more candidates can overload validation. The next useful action might be a test, a document lookup or a repaired command.

Boiko et al. combined language models with laboratory automation in Coscientist (2023). Their opportunity was to connect reasoning and tools across tasks; the design comparison follows our reconstruction.

The failure determines which workflow to build

Workflow	Useful when	Remaining limitation
Fixed script	Procedure and contingencies known	Unanticipated failures need help
Adaptive test selection	Next test depends on measurements	Needs a defined action space
Tool-using agent	Next action may require interpretation	Extra decisions create failure modes

Choose the simplest workflow that can respond to the relevant uncertainty. Flexibility helps only if useful recovery or better decisions outweigh its cost and errors.

A repaired command still needs chemical evidence

In Coscientist's reported demonstration, an incorrect instrument command failed. The system consulted documentation, corrected the protocol and executed it successfully.

Subsequent gas chromatography–mass spectrometry supported formation of the intended products. Command success established execution; the measurement supplied chemical evidence.

This was a reported synthesis demonstration with a software correction. It does not establish a new reaction, an autonomous scientific breakthrough or a general discovery success rate.

Confirmation must survive the selection process

L04's 20 null candidates showed how a promising screen can arise by chance. A binder or mechanism claim needs confirmation appropriate to its endpoint.

For binding: use independent measurements and specificity controls. For mechanism: test a rival prediction, with controls for the intervention and measurement.

One evidence ledger, three different conclusions

Case / denominator	Comparison or control	Warrant	Unresolved
Program:256 inputs	Exact specification	Complete finite check	Other inputs
PD-1:13 / 53	Binding assay	Selected tested binders	General efficacy
Resistance:20 cultures	Same-culture samples	Lineage fluctuation	Molecular mechanism

Conclusion: common search does not imply common evidence

Cross-domain comparison is strongest when it preserves the scientific claim and examines how each stage earns its warrant.

Next: how validated results and reusable methods change the difficulty of future problems.

References

Bernardino Romera-Paredes et al. (2024). Mathematical discoveries from program search with large language models. Nature 625, 468–475; online 14 December 2023

Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao and Adam Zsolt Wagner (2025). Mathematical exploration and discovery at scale. arXiv:2511.02864v1; preprint, 3 November 2025

Alexander Novikov et al. (2025). AlphaEvolve: A coding agent for scientific and algorithmic discovery. arXiv:2506.13131; white paper/preprint

Thomas Hubert et al. (2025). Olympiad-level formal mathematical reasoning with reinforcement learning. Nature; journal article, 2025

References · continued

[Sumit Gulwani, Oleksandr Polozov and Rishabh Singh \(2017\). Program Synthesis. Foundations and Trends in Programming Languages 4, 1–119; survey monograph](#)

[Mert Yuksekgonul et al. \(2026\). Learning to Discover at Test Time. arXiv:2601.16175v2, 5 February 2026; preprint](#)

[Haotian Ye et al. \(2026\). Structured Scaling of AI Discovery Across Diverse Scientific Domains. arXiv:2604.19341v2; preprint, 27 July 2026](#)

[Joseph L. Watson et al. \(2023\). De novo design of protein structure and function with RFdiffusion. Nature 620, 1089–1100; journal article](#)

References · continued

Martin Pacesa et al. (2025). One-shot design of functional protein binders with BindCraft. Nature 646, 483–492; online 27 August 2025

Benjamin Fry, Kaia Slaw and Nicholas F. Polizzi (2026). Zero-shot design of drug-binding proteins via neural iterative selection–expansion. Nature 656, 237–249; online 24 June 2026

Judea Pearl and Elias Bareinboim (2014). External Validity: From Do-Calculus to Transportability Across Populations. Statistical Science 29, 579–595; journal article; arXiv posted 2015

S. E. Luria and M. Delbrück (1943). Mutations of Bacteria from Virus Sensitivity to Virus Resistance. Genetics 28(6), 491–511; published primary research

References · continued

Lingchen He et al. (2024). Tail assembly interference is a common strategy in bacterial antiviral defenses. Nature Communications 15, 7539; journal article

Juraj Gottweis et al. (2026). Accelerating scientific discovery with Co-Scientist. Nature 655, 487–496; journal article

Daniil A. Boiko, Robert MacKnight, Ben Kline and Gabe Gomes (2023). Autonomous chemical research with large language models. Nature 624, 570–578; journal article