



Multi-Agent Drug Discovery (MADD)

A provider-neutral system for evidence-audited, multi-agent drug design

VARIANT RESEARCH GROUP

Whitepaper · October 2026

Software version 0.1.0 (pre-release) · License: Apache-2.0 · Source: <https://github.com/VariantResearch/multi-agent-drug-discovery>

Executive summary

Multi-Agent Drug Discovery (MADD) designs drugs against a target you name: a disease, a receptor, a genetic disorder, or a molecular target in a disease context. A panel of AI research agents studies the target from evidence retrieved for it, a structured debate decides whether that evidence supports locking a design specification, and only then do designer models propose candidate compounds. Candidates are audited, computed on, and ranked, and each ranked candidate receives a production sheet stating what would be made, its computed identity, its production route, and, named explicitly, whatever the designer did not specify.

The system exists because of a failure mode. Language models produce fluent research narratives, and fluency is not evidence: models invent citations, quote sources that do not say what is claimed, and, given the chance, grade their own homework. MADD's answer is architectural rather than rhetorical. Its rule is that no claim counts unless it rests on a real, checkable primary source, and no model gets to grade its own homework. Mechanical auditors, not the models, decide whose evidence survives. Every claim in a run must name a source in a public registry, carry an evidence tier, and provide a supporting quote. Audits strike claims that are unsourced, malformed, retracted, or unsupported by the source they cite. Scoring is deterministic and computed from audited evidence only: model confidence, prose quality, and vote counts are never inputs.

MADD is provider-neutral and open source (Apache-2.0). The package contains the complete working application: the target-first pipeline, the research and debate engine, the evidence auditors, the design stage, a retrospective test set of eight real drug programs (Gate 0), run orchestration, adapters for live models, a command line interface, and a local web console. The only thing not included is the models themselves: choose a roster, supply API keys, and it runs.

This whitepaper describes how the system works, the evidence rules it enforces, how it has been tested, and, plainly, where it stands. That standing includes its test results: the pipeline has run end to end in owner-directed validation runs, two target runs have produced ranked, gate-passing candidates, and four calibration runs against Gate 0 each failed exactly one of the pre-approved pass thresholds, for reasons the record names. No scored production run has taken place.

1. The problem: fluent, unsourced, self-graded

Three habits make AI research output hard to trust.

First, citation invention. A model asked for supporting literature will produce plausible references, some of which do not exist, and others that exist but do not support the sentence they are attached to. The failure is invisible in the prose; it only appears when someone checks.

Second, self-grading. A pipeline that lets the same model propose a claim and judge the claim has not tested anything. Agreement between a model and itself, or between near-identical models prompted alike, is weak evidence presented as strong evidence.

Third, confidence as currency. Systems that score outputs by model-reported confidence, or by how polished the writing is, reward the wrong thing. A wrong answer stated beautifully outranks a right answer stated plainly, and nothing in the score records which claims actually rested on evidence.

MADD is built so that each of these habits fails mechanically. A claim without a resolvable source is struck, not debated. The auditors that strike claims are code, not models. A model may judge whether a cited passage supports a claim, under a constrained verdict scheme, but it cannot invent the passage, and the final scoring arithmetic takes no input from any model's opinion of itself.

2. Design principles

Six principles govern the system, and each is enforced in code rather than requested in a prompt.

1. **No claim without a checkable primary source.** Every load-bearing claim names a registry identifier (a PubMed ID, a DOI, a trial registration, or an identifier in one of five other registries), an evidence tier, and a supporting quote. What cannot be checked does not count.

2. **Mechanical auditors decide.** Claim and citation audits are deterministic programs. They strike claims for stated reasons, and every strike is recorded in the run bundle with its reason.
3. **Blinding.** Research agents work independently, without seeing each other's dossiers, and retrospective test cases are stripped of outcome-identifying material. Agreement that emerges under blinding means more than agreement produced by imitation.
4. **Dissent is data.** When a run cannot lock, the disagreements are preserved in a mandatory dissent log instead of being averaged away. A no-go is a valid outcome, and a stopped run is a valid outcome.
5. **Deterministic scoring.** Ranking and scoring are fixed arithmetic over audited claims, weighted by evidence tier. The same inputs produce the same scores, and the bundle records everything needed to recompute them.
6. **Honest labelling.** Every output is labelled by the highest validation gate it has actually passed. Agent agreement and model confidence are not validation gates and are never presented as one. What a designer did not specify is recorded as unspecified, never filled in.

3. How a target run works

A target run (`madd target-run`) takes one named target through eight steps.

1. Frame and retrieve. The operator names the target (a disease, receptor, genetic disorder, or molecular target) and its disease context. Evidence is retrieved for it from Europe PMC, and the frame carries those sources as the evidence set. When retrieval restriction is enabled, panel citations are held to the retrieved set: a real source outside it is still struck, because the run's evidence is the frame, not the open literature.

2. Blinded dossiers. The frame goes to several research agents at once. Each works blind and returns a structured dossier: claims with sources, tiers, and quotes; a proposed focus; and an explicit list of what the agent could not verify.

3. The evidence call. A claim ledger audit and a citation audit examine every claim. Claims are struck when they are unsourced, sourced to model opinion, malformed against the claim schema, or, when live resolution is enabled, pointing at a registry record that does not exist. Struck claims take no further part, and every strike is recorded with its reason. Section 4 details the audit machinery.

4. Debate. Surviving dossiers are cross-examined: each agent states the strongest objection it can support against each rival dossier and answers objections to its own. If every dossier agreed on the focus in the blinded round, an extra red-team round runs in which each agent must argue against the agreed focus using surviving claims only; instant agreement is treated as a warning sign, not a strength. Dossiers are then ranked pairwise on audited evidence scores only, and one evolution round merges the best material.

5. Consensus tests. Four tests decide whether the debate may lock a design specification: agreement (all dossiers propose the same focus, compared by canonical identifier), survival (every load-bearing claim in the proposed specification passes the claim ledger audit), the evidence floor (the specification rests on at least one audited claim of adequate tier), and, where the run's protocol mode includes a judge, stability under a judge swap (Section 5). Every applied test must pass.

6. Lock or no-go. If the tests pass, the specification locks and design may begin. If any test fails, the run ends in a no-go, the dissent log records why, and no design work starts.

7. Design. Designer models propose candidate compounds against the locked specification. A real candidate must carry a structure, a SMILES string or a sequence; structureless proposals are dropped and the drop is recorded. Structures are validated and enriched computationally, every claim a candidate carries passes the same audits as the research claims, and candidates are ranked by a fixed deterministic rule (Section 7). A separate design gate decides whether the run's output counts as designed at all.

8. Production sheets and the run bundle. Each ranked candidate receives a production sheet (Section 7). The whole run (dossiers, strikes, transcripts, candidates, production sheets, and a manifest with per-call prompt hashes) is frozen into a hashed, verifiable run bundle with a plain-language report, so any later reader can check what was claimed, what was struck, and why.

4. The evidence policy

The evidence policy is the system's constitution, and it is implemented as a set of audits that run on every run that enables them.

Sources and registries. Claims cite identifiers resolved against the registry that owns them: PubMed (PMID), Crossref (DOI), ClinicalTrials.gov (NCT), Europe PMC (PMCID and preprint identifiers), the GWAS Catalog (study accessions and rsIDs), ChEMBL (molecule IDs), PubChem (compound IDs), and UniProt (accessions). A registry that cannot be reached never strikes anything; an unreachable registry proves nothing either way.

Evidence tiers. Each claim carries a tier reflecting the kind of evidence behind it. In descending order of weight: tier 1, human causal genetics; tier 2, human clinical or human phenotype data; tier 3, human tissue expression data; tier 4, human cell or organoid data; tier 5, animal data, or in-vitro data not from a human system; tier 0, model opinion. (This is the mapping in the project's evidence policy document, docs/EVIDENCE_POLICY.md; the tier-assignment rules for specific study types are recorded there as not yet established.) Tier 0 material can never be load-bearing, and no agent may cite another agent as evidence.

The audits. The claim ledger audit enforces the structural rules: a load-bearing claim must carry a source identifier, must not be sourced to a model or agent opinion, and must not cite a tier-0 source. The citation audit checks identifier formats and requires claims marked quote-

checked to carry a supporting quote; it makes no network calls and never upgrades a claim's status. When live resolution is enabled, identifiers the registries report as not found are struck, and a DOI whose Crossref record carries a retraction or withdrawal notice is struck as retracted. When the full-text quote check is enabled, a quote-checked claim whose supporting quote appears in neither the source's abstract nor its retrieved open full text is struck; the strike requires the full text to have been retrieved, so an abstract alone can clear a quote but never strike one. When the entailment check is enabled, a judge compares each surviving quote-checked claim with the passage it cites; only a refuted verdict strikes. Partial support is recorded as a warning grade, not a strike, and a judge that cannot answer strikes nothing. Duplicate claim identifiers inside a dossier are struck in every copy.

Deterministic focus scoring. A dossier's focus score counts audited claims only, weighted by tier (tier 1 counts 100, tier 2 counts 70, tier 3 counts 40, tier 4 counts 25, tier 5 counts 10, tier 0 counts 0), each claim counted once no matter how often repeated. Model confidence, prose quality, agent seniority, and vote counts are never inputs.

Budget enforcement. A run configuration may cap total cost and total calls; at least one cap is required when a budget is given, the cap is shared across the whole run, and the run stops when it is reached. The budget stop is a hard stop in code, not a guideline.

5. Protocol modes

Runs execute under one of three protocol modes, chosen in the run configuration. The modes count distinct models, not agents: every mode fields a research panel of at least two agents, runs the same debate steps, and puts every dossier through the same evidence call. What changes is how many models the panel spans and whether a judge model sits over it.

- **Simple.** Every research agent runs one and the same model. There is no judge, and the judge-swap stability test is recorded as not applied; a specification locks on agreement, survival, and the evidence floor. This is the cheapest configuration, and the one to validate a setup with.
- **Dual.** The research agents span two models, adding a second model's independent read of the same evidence. Still no judge; stability is likewise recorded as not applied.
- **Dual judge.** The research agents span two models, and a judge model adds the stability test, so all four consensus tests apply. This is the full protocol, and the mode a scored Gate 0 run uses, because the approved thresholds judge judge-swap stability.

The shape of a mode is enforced when the configuration loads and again in the runner before the run starts: a judge declared in simple or dual mode, a missing judge in dual judge, or the wrong number of distinct models on the panel is refused with the reason stated, before any call is made. A configuration that declares no mode keeps the original behaviour: a panel of two or more providers of any composition, with a judge optional; without a judge, stability cannot be

established and debates end in no-go. Where threshold evaluation meets a run in which no judge swap was performed, it records stability as not evaluable rather than as a pass.

6. Gate 0: the retrospective test

Before a system like this is trusted on a new question, it should be measured on questions whose answers are known. Gate 0 is that measurement: eight real, completed drug programs, four that succeeded and four that failed, run blinded against evidence frozen before each program's pivotal outcome was public.

Case	Program	Outcome class	Evidence as of
evolocumab-pcsk9-cvd	Evolocumab / PCSK9, cardiovascular disease	Positive	2017-03-16
ivacaftor-cftr-cf	Ivacaftor / CFTR, cystic fibrosis	Positive	2011-10-31
semaglutide-glp1r-obesity	Semaglutide / GLP-1 receptor, obesity	Positive	2021-02-09
erenumab-cgrpr-migraine	Erenumab / CGRP receptor, migraine prevention	Positive	2017-11-28
verubecestat-bace1-alzheimer	Verubecestat / BACE1, Alzheimer disease	Negative	2017-02-01
pf05089771-nav17-dpn	PF-05089771 / Nav1.7, painful diabetic neuropathy	Negative	2018-02-28
ben2293-pantrk-atopic-dermatitis	BEN-2293 / pan-Trk, atopic dermatitis	Negative	2023-04-03
rec994-ccm	REC-994, superoxide-mediated oxidative stress in cerebral cavernous malformation	Negative	2025-05-04

Nothing in the case set is synthetic: every input claim was quote-checked by hand against its primary source at construction, with a verification ledger kept in the repository. Input packs carry a list of outcome-revealing terms, and a leakage scan checks that no outcome leaks into what the agents see; any case whose outcome did leak would be removed and recorded. Outcomes are sealed in separate files that agents never receive and the input builder never reads; they are opened only at scoring time. One honest qualification travels with the set: these outcomes are famous, and frontier models may remember them despite the blinding and the cutoff dates. Gate 0 is therefore labelled a calibration and regression set, not a pristine blind test.

The pass thresholds were fixed and approved on 2026-10-22, before any run took place, and they are stated here in full:

1. **Failure kill rate:** the debate system must call at least 3 of the 4 known negative cases a no-go.
2. **Positive advance rate:** the debate system must advance at least 3 of the 4 known positive cases.
3. **Baseline margin:** the debate system must score at least 1 case more correct, in total, than a single-model baseline on the same eight cases.
4. **Audit integrity:** zero load-bearing claims in any final dossier may rest on a struck or unverified citation. One violation fails the phase regardless of the case score.
5. **Stability:** swapping the ranking judge to a different model family must not change the winning call on more than 1 of the 8 cases.
6. **Budget:** a full eight-case run, debate plus baseline, must complete within the per-run budget recorded in the run manifest. The figure itself is set by the owner when API access is arranged, and recorded before the first scored run.

A run must also be complete, every case called, to pass at all.

Gate 0's current status is precise. Validation runs have been performed on the set and are reported in Section 8. No scored run has taken place: a scored run first requires the model roster and the budget figure to be recorded in the specification, and that has not happened. By the owner's decision of 2026-10-25, Gate 0 validation is an option rather than a requirement: the production program (live target runs under any protocol mode, and design on locked specifications) may proceed without a scored or passing Gate 0 run. The thresholds are unchanged, and validation results are still reported against them, plainly. What the decision removes is the gate, not the measurement.

7. Phase 2: design from a locked specification

Design starts only from a locked consensus specification; a specification that is not locked is refused, and no design work starts from a killed or unresolved target. The brief is built mechanically from the locked spec: the target, the disease context supplied by the operator, the surviving load-bearing claims as evidence constraints, and any free-text constraints the operator adds.

Three kinds of designer sit behind one interface. A deterministic mock designer exists to exercise the pipeline: its candidates are synthetic, clearly labelled, structure-free, and can never pass the design gate. A model panel proposes candidates under a strict contract: a real structure (SMILES or sequence), claims ruled by the same ledger as the research claims, and predictions labelled as predictions. External structure tools run through a license-gated runner: a tool whose license terms are unverified is blocked, not assumed fine, and a blocked tool run fails the design gate. Tools cleared in the registry include RDKit (structure validation and descriptors), ADMET-AI

(predicted ADMET endpoints, recorded as unverified predictions, because a trained model's prediction is information, not a measurement), AutoDock Vina (docking), PoseBusters (pose geometry checks), and AiZynthFinder (retrosynthesis search). Two structure-prediction tools, Boltz-2 and Chai-1, are registry-cleared but not installed in the development environment, so no structure-prediction run has happened yet; two others (AlphaFold 3 and ESM3-open) are blocked by the registry for commercial use under their terms.

Every claim a candidate carries passes through the same audit machinery as the research claims, and a candidate whose load-bearing claim carries any audit error, or which records a failed constraint, is excluded from the ranking, with its reasons kept. Ranking is a fixed deterministic rule: audited load-bearing claim count, then fewer unknown constraints, then candidate identifier. Mock candidates are stamped SYNTHETIC in the text report, the HTML report, and the console, so a ranked list is never mistaken for real candidates.

The design gate passes only when the brief came from a locked spec, at least one candidate is ranked, at least one ranked candidate is real rather than a synthetic mock or fixture, no tool run was license-blocked, and, when tool runs were requested, at least one completed. A mock-only run fails the gate by design.

Each ranked candidate receives a production sheet, built mechanically: the structure exactly as proposed; its computed identity (canonical SMILES, molecular formula, molecular weight, and InChIKey for small molecules; length, composition, and an alphabet check for sequences); the production route its modality implies (chemical synthesis, solid-phase peptide synthesis, or biologic production); the retrosynthesis result when a route search ran; and a not-specified list naming the production-critical details the designer did not supply, such as a sequence candidate's modification pattern, unassigned stereochemistry, or the confirmation assay. A sheet is marked production-ready only when the thing to make is fully determined: a valid structure (readiness requires positive structure validation), an assigned route, and, for sequences, a specified modification pattern.

One boundary is stated in the specification itself and repeated here: a candidate record is not evidence that the candidate binds the target, works in an assay, or is safe. Docking, ADMET prediction, and retrosynthesis search are computational steps; none of them validates a candidate biologically.

8. Validation record

MADD's validation record is published in the repository (docs/VALIDATION_RUN_2026-10-25.md) and summarized here without varnish. Two facts frame everything in this section. First, no run so far has used a commercial model account: the provider adapters for live models are tested against fake transports shaped from the documented wire formats, and have not been run against a real account. Second, the validation runs below were staffed by independent agents of a single model family, served through the real provider interface, so the production pipeline (debate engine, evidence call, scoring, threshold code) executed exactly as shipped, but the

multi-family hypothesis the scored run is designed to test was not tested. What these runs validate is the process.

The four Gate 0 validation runs (2026-10-25)

Run	Configuration	Debate	Baseline	Threshold failed
1	Full panel, no declared mode	6 of 8	5 of 8	Kill rate: 2 of 4 killed, bar is 3
2	Full panel, quote check repaired	6 of 8	5 of 8	Advance rate: 2 of 4 advanced, bar is 3
3	Full panel, canonical focus fix	7 of 8	7 of 8	Baseline margin: 0, bar is at least 1
4	Simple mode	7 of 8	8 of 8	Baseline margin: -1, bar is at least 1

Run 1 failed on the kill rate alone: the panel advanced verubecestat and PF-05089771, the two calls the field itself made before the outcomes were known, on tier 1 genetics evidence. Seven claims were struck during the run, five of them in a single case, which exposed a defect in the quote check: it searched the deposited full text only, while the case packs quote abstracts, and the two do not always match. The check was repaired to search the record's abstract as well (a quote found in either clears; a strike still requires the quote to be absent from both, with the full text actually retrieved), and run 1's results stand as executed under the rule as it then was.

Run 2 passed the kill rate at 4 of 4 and failed on the advance rate for a mechanical reason: the agreement test compared proposed focus labels as exact strings, and two panels that voted unanimously to advance spelled the target two ways ("CFTR-G551D" against its expanded form; "GLP-1 receptor" against "GLP-1R"), so agreement failed on spelling. The repair gave each case a canonical focus identifier that the engine maps proposals onto before the consensus tests read them. The live audit struck nothing in run 2.

Run 3, with both repairs in, scored 7 of 8 against the baseline's 7 of 8 and failed the margin. The audit struck nothing; every citation resolved and every checked quote was found. The one remaining miss, verubecestat, was advanced by panel, judge, and baseline alike.

Run 4 was the first run under a declared protocol mode (simple). The panel called all four negatives correctly and locked three of four positives, losing erenumab to a split panel; the single-model baseline, being the same model working alone, called all eight correctly, so the margin failed at -1 and stability was recorded as not evaluable (no judge, no swap). Across the four runs the debate scored 6, 6, 7, 7 while the baseline swung 5, 5, 7, 8.

The record's own conclusion is structural: in a single-model staffing, the baseline is the same mind working alone, and the debate machinery can only lose calls to panel disagreement, never gain one the baseline cannot make. The missing ingredient is the roster itself: different model families in the panel seats, which the scored run has always required. The report's verdict on the first run states the standing plainly: this panel configuration would not be cleared for production.

The simple-mode configuration, separately, stands validated as a working protocol whose panel calls 7 of 8 on this set, and stands not passed against the full threshold battery.

Target runs

Two target runs show the full pipeline, research to ranked candidates, working end to end. Both were run in simple mode with agent-staffed providers, and both are labelled in the record as pipeline validation, not candidate claims.

KRAS G12C in non-small cell lung cancer (2026-10-25). Evidence was retrieved live from Europe PMC (12 sources) and citations were held to that set. Three research agents returned 19 claims; live resolution, live quote checking, and entailment review produced zero audit strikes. The debate locked the target with 5 surviving load-bearing claims as the design brief's evidence constraints. Two designer agents proposed four original covalent-inhibitor candidates with valid structures; RDKit validated all four, ADMET prediction completed, all four ranked, and the design gate passed (outcome: designed). The record's characterization is quoted because it is the correct one: this validates the pipeline, not the candidates; they are model proposals, validated computationally in the run and not made or assayed, against a target that already has approved drugs.

MTCH2 inhibition (2026-10-25). A demonstration run (MTCH2 was the demonstration subject; no target is locked in or selected by it) that locked and passed its design gate with four ranked candidates: two small molecules (formulas C17H11F3N2OS and C18H13F3N2O2S, both marked production-ready on their sheets) and two antisense oligonucleotides (both not production-ready, because their modification patterns were recorded only in designer prose, not specified as the sheets require). The run also shows the audit costing a panel its strongest material: 12 of 24 claims were struck, all for quotes not found in the full text, including the claims carrying the three strongest directional papers, because the panel had marked title-level quotes as quote-checked. The target locked on the six surviving load-bearing claims, and the run's own README states the consequence: the design rests on a thinner, partly direction-neutral audited base than the outcome line alone suggests. Build guides were written for the two small-molecule candidates only; they are labelled proposal work for compounds not yet made or assayed, and no retrosynthesis tool ran for them. A third target run, MTCH2 upregulation, ended in a no-go when the agreement test failed on direction, and its dissent is preserved in its report.

Test suite

The package carries a test suite of 297 tests, run with pytest (`python3 -m pytest tests/`), covering the auditors, the debate engine, Gate 0 dataset integrity over the real case set, the design stage, the console, and documentation coverage (every module, class, and function must carry a docstring; the README must name every package module). The suite needs pytest and, for its structure tests, the optional RDKit package; without RDKit those tests fail rather than skip (measured 2026-10-26: 279 pass, 9 fail, 9 skip), while everything else passes. (The most recent

full record validation, on 2026-10-26, recorded 288 passing before nine regression tests for that day's audit fixes brought the suite to its current 297.) The package itself declares no dependencies; that claim is package-scoped and does not extend to the test suite or to the optional design enrichments (RDKit, ADMET-AI), nor to live runs, which need provider API keys.

9. The MADD Console

A local web console ships with the package (`python3 -m madd gui`, serving `http://127.0.0.1:8765` by default). It provides run control (check a configuration, dry-run it, run Gate 0 and watch per-case progress live), the Gate 0 scoreboard, and a run library with per-run scores, threshold checks, styled HTML reports, and debate transcripts. A locked run's view offers one-click design, and design runs appear in the library with their ranked candidates, synthetic candidates marked as such.

For a completed design, the run view renders a 3D structural view: the candidate as a shaded WebGL model (ball-and-stick, sticks, space-filling, or lines) beside the receptor structure on file for the target. The honesty labels are part of the feature: the molecule geometry is a generated conformer for visualization, not a measured structure; the receptor shown for MTCH2 is the AlphaFold predicted model for human MTCH2 (AF-Q9Y6C9-F1, mean pLDDT 86.3), labelled predicted everywhere it appears; the candidate has not been docked, and no binding pose is shown or claimed. A build guide panel renders each ranked candidate's production sheet as steps (make it by its route, confirm identity against the sheet, run the confirmation assay), naming anything the sheet records as unspecified rather than inventing it.

The console binds to localhost by default and has no authentication. It is a local tool, and running a configuration from it executes that configuration exactly as the command line would; the README says so, and so does this paper.

10. Responsible use

MADD is a research-design and evidence-audit scaffold, and its responsible-use rules (RESPONSIBLE_USE.md in the repository) are part of the design, not optional guidance.

- **Scope exclusions.** No pathogen targets. No toxin targets. No work aimed at increasing harm, transmissibility, immune evasion, or environmental persistence of any organism, and no design of agents intended to harm humans, animals, or crops. No human dosing recommendations: nothing the software produces is a dose, a dosing schedule, or advice to administer anything to a person. No clinical advice: nothing here diagnoses, treats, or recommends treatment for any person.
- **No autonomous external action.** The software must not submit papers, filings, orders, messages, or applications to any external body on its own, and must not order synthesis, assays, materials, or services. Any wet-lab step is a handoff to named humans operating

under their own approvals and procedures. The debate protocol ends at a recommendation to a human, never at an external action.

- **Human sign-off gate.** A named human must sign off before a run moves from validation to any external step, including sharing a candidate with a laboratory partner. The sign-off, the artifact hashes signed off on, and the gate reports relied on must be recorded in the run manifest.
- **Honest labelling.** Every output is labelled by the highest validation gate it has actually passed, using the project's gate ladder (retrospective test, citation audit, independent-tool cross-check, computational design checks, human review sign-off, partner-lab experimental handoff). An output that has passed only computational checks must say so in the output itself; it must not be described, in any document generated from it, as tested, validated in vitro, validated in vivo, or effective.

11. Status and open items

Status as of October 2026, stated as the record states it:

- The application is complete and runnable end to end: target runs, the debate engine, the audits, the design stage, Gate 0, the console, and the CLI are built, tested, and documented in the repository.
- Validation to date is process validation: the pipeline executed exactly as shipped in agent-staffed runs, including two target runs that produced ranked, gate-passing candidates. Those candidates are computational proposals. None has been synthesized or assayed, and nothing in this paper or in the repository claims otherwise.
- No scored Gate 0 run has taken place, and the single-model panel configuration used in the validation runs would not be cleared for production under the approved thresholds.
- The provider adapters for live model accounts are implemented and tested against fake transports; they have not been run against a real model account.

Three decisions belong to the project owner and are recorded as open in the specification: the commercial model roster for the scored run (the spec requires three frontier model families from three providers, plus one independently runnable open-weight model as the reproducibility baseline, and names none as approved yet), the per-run budget figure, and the API access that both depend on. The scored run's protocol is fixed in advance: record the roster, baseline, and budget; freeze the Gate 0 inputs at their current hashes; run the baseline first and seal its calls; run the debate system in dual_judge mode; score both against the sealed outcomes; publish the full run bundle, including the cases the system got wrong; and write the result into the changelog in plain terms.

12. Availability and license

MADD is open source under the Apache License, Version 2.0. The complete application, its specifications (docs/PHASE0_SPEC.md, docs/PHASE1_SPEC.md, docs/PHASE2_SPEC.md), its evidence policy and debate protocol, the Gate 0 case set with its verification ledger, the validation record, and the run bundles cited in this whitepaper are all in the public repository:

<https://github.com/VariantResearch/multi-agent-drug-discovery>

Requirements: Python 3.11 or newer. The package itself uses only the Python standard library. Optional components (pytest for the test suite, RDKit and ADMET-AI for design enrichment) and live model runs (provider API keys, held in environment variables, never in the repository) are documented in the README.

Variant Research Group · VRG Labs. This whitepaper describes software and its validation record as of October 2026. It is a technical document about a research tool; it is not medical, clinical, or investment advice, and nothing in it is a claim that any candidate compound is safe or effective.