Supporting Information

Anyone Can Dock: An Online Molecular Docking Tool for Everyone

Kowit Hengphasatporn[1], Thanthapatra Bunchuay[2]*, Lian Duan[1], Yasuteru Shigeta[1,3]*

[1]Center for Computational Sciences, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan

[2]Department of Chemistry and Centre of Excellence for Innovation in Chemistry (PERCH-CIC), Faculty of Science, Mahidol University, Bangkok 10400, Thailand

[3]Center of Excellence in Structural and Computational Biology, Department of Biochemistry, Faculty of Science, Chulalongkorn University, Bangkok, Thailand

*Correspondence: kowith@ccs.tsukuba.ac.jp, thanthapatra.bun@mahidol.ac.th

KEYWORDS: molecular docking, web application, AutoDock Vina, drug design, protein–ligand interaction, virtual screening

Table S1 Supported protein and target types in the platform. The table summarizes target classes currently supported by the docking workflow and notes practical limitations relevant to receptor preparation, grid definition, visualization, and interaction analysis.

Protein or target type	Support	Notes
Standard single-chain proteins	Full	Primary use case.
Multi-chain / homo-oligomers	Full	Duplicate chains can be deduplicated; when the same ligand appears in equivalent chains, the chain A ligand is auto-selected without unnecessary dropdowns.
Heme proteins (CYP450, peroxidases, Hb, Mb)	Full	Fe-porphyrin is handled separately; the grid can be auto-centered on Fe and the heme group is shown in the viewers.
Metal-binding proteins (zinc fingers, carbonic anhydrase)	Full	Common metal ions such as ZN, MG, CA, MN, FE, and CU are re-injected with assigned formal charges after receptor preparation.
MD simulation outputs (GROMACS, AMBER)	Full	Blank chain IDs are automatically assigned to chain A.
Non-standard ligand names (MOL, LIG, UNL, INH)	Full	HETATM keyword handling bypasses potential ProDy misclassification.
Modified amino acids (CYP, MSE, TPO, SEP)	Full	Backbone atom checks retain modified residues as part of the receptor when appropriate.
Multiple co-crystal ligands	Full	Ligand-like HETATM records are auto-scanned. A dropdown is shown only when multiple reference-ligand candidates exist; the selected ligand defines the grid center and is removed from the receptor.
Cofactor-binding proteins (FAD, NAD, ATP, CoA)	Full	Cofactors are classified separately from ligand candidates and can be kept or stripped independently.
Glycoproteins	Partial	Glycans are kept in the receptor, but they are not currently shown in the 2D interaction diagram.
Antibodies / very large proteins	Partial	Supported, but 3D visualization and interaction detection may be slow for very large systems.
Membrane proteins	Partial	Dockable without lipids; lipids are not automatically filtered.
RNA / DNA targets	Partial	Hydrogen-bond and hydrophobic-contact detection can run, but nucleic-acid-specific interactions are not currently represented in the 2D diagram.
Covalent docking	Not supported	AutoDock Vina is designed for non-covalent docking.
Dimer interface binding sites	Not supported	Automatic chain deduplication can remove the second chain, making interface-site docking unsuitable in the current workflow.

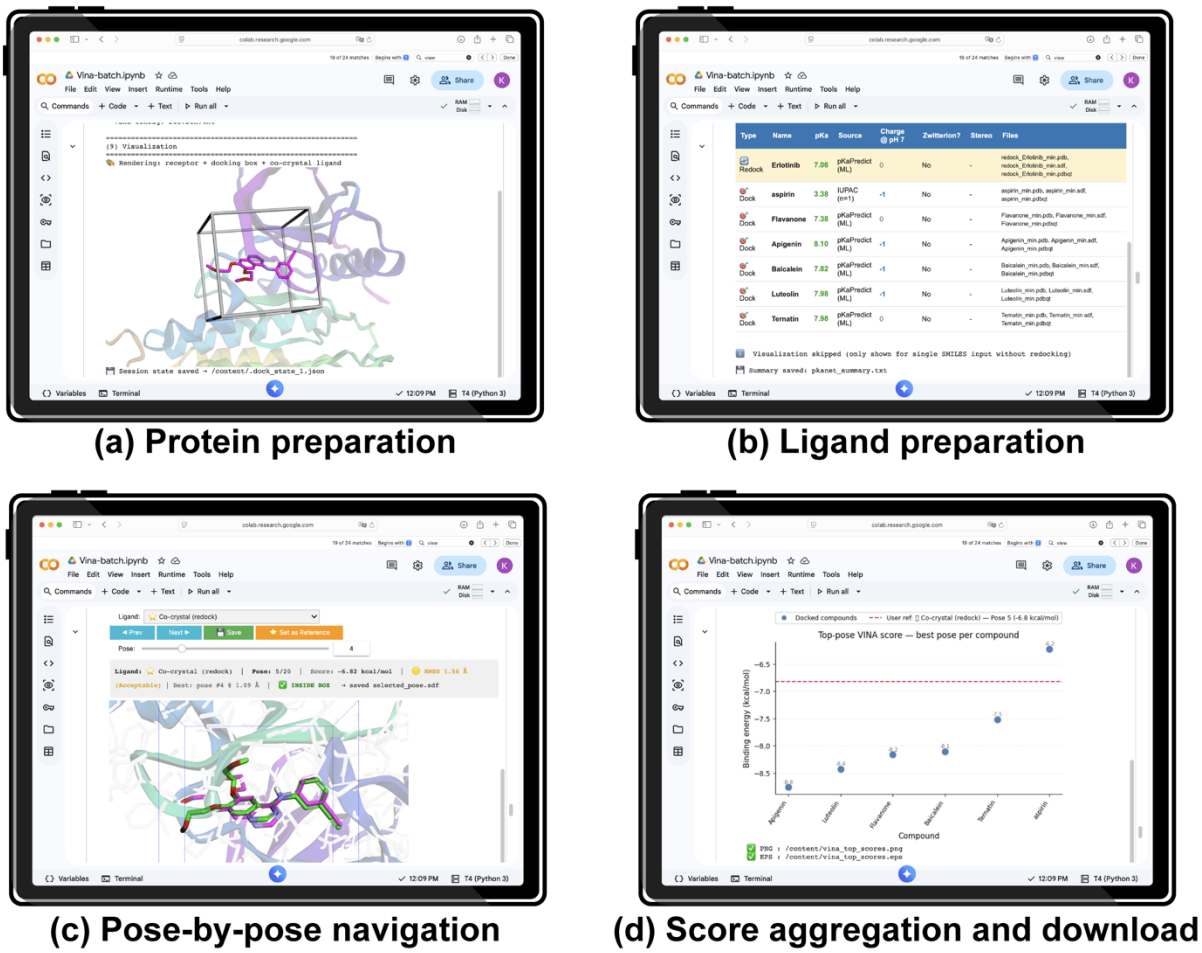

Figure S1. Google Colab notebook interface running on iPad. (a) Receptor preparation with metal-safe processing. (b) Batch ligand preparation. (c) Interactive pose browser with RMSD validation. (d) Score aggregation and download. No installation was required.

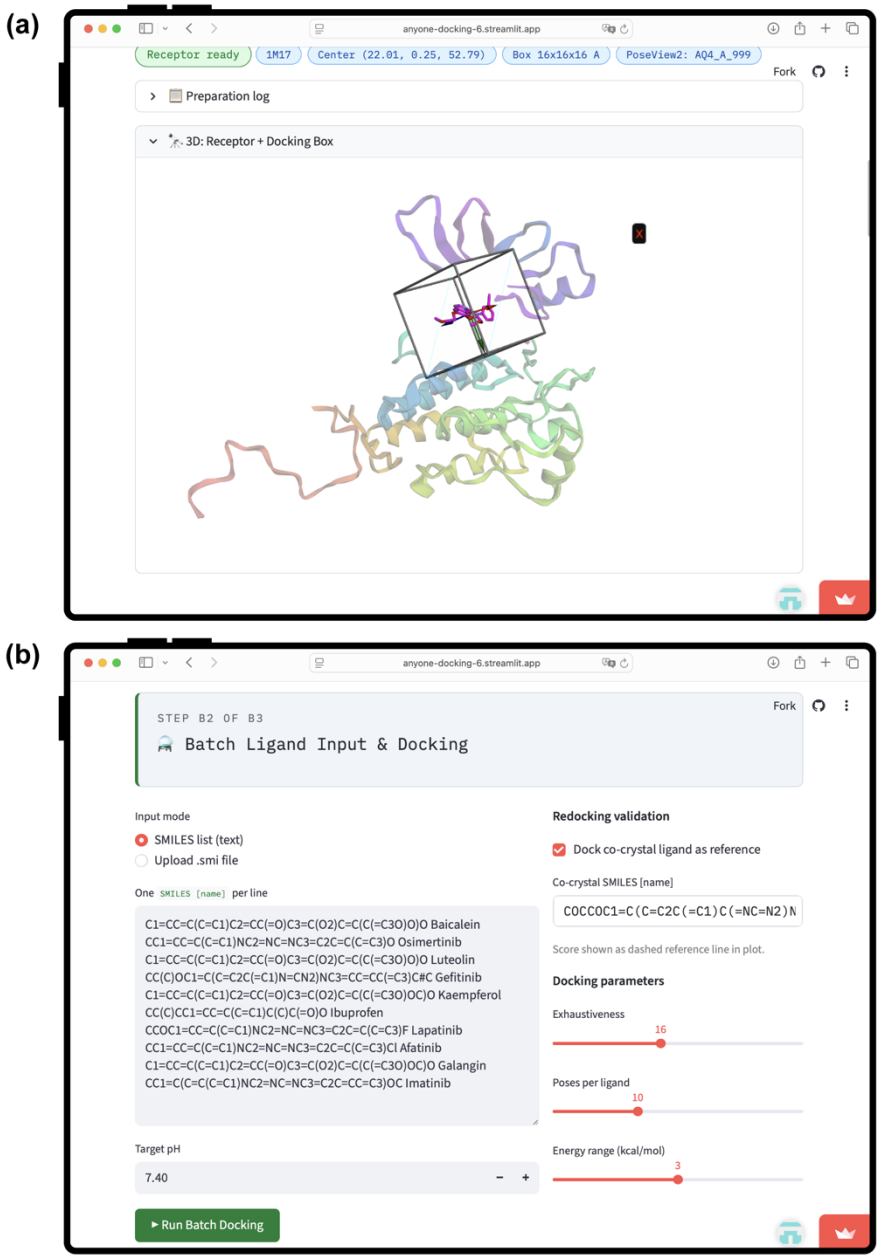

Figure S2. Worked example of the complete batch docking workflow using EGFR kinase (PDB: 1M17) with gefitinib as the test ligand and erlotinib as the co-crystal reference. (a) Receptor preparation: the EGFR kinase domain loaded by PDB accession code with automatic grid centring on the co-crystallised erlotinib ligand (AQ4); the 16×16×16 Å docking box is displayed in the integrated py3Dmol viewer with spectrum cartoon colouring. (b) Batch ligand input and docking parameters: ten ligands entered as SMILES strings with names, target pH set to 7.4, erlotinib entered as the co-crystal redocking reference, and docking executed at exhaustiveness 16 with 10 poses per ligand.

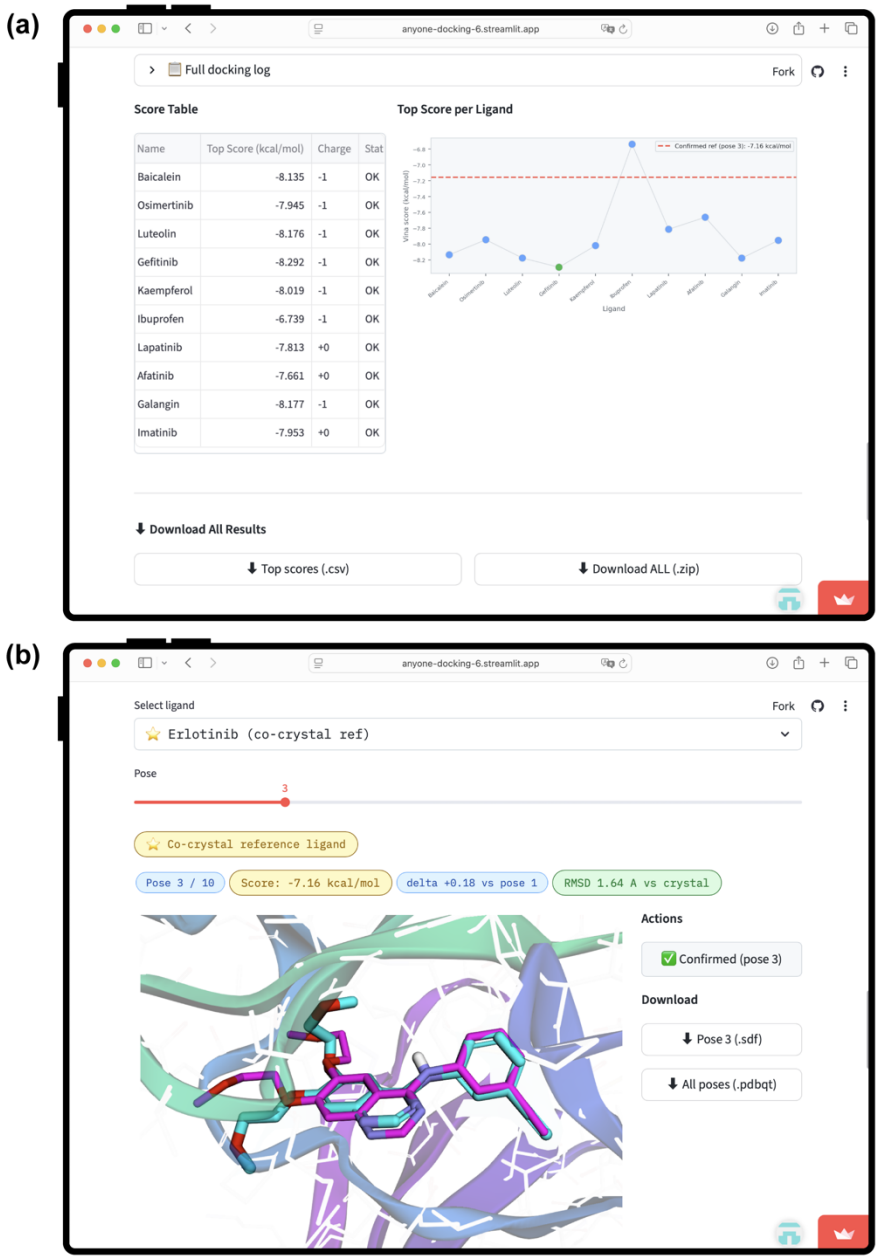

Figure S3. Batch docking results for ten ligands against EGFR kinase (PDB: 1M17). (a) Score table (left) ranking all ten ligands by top predicted binding affinity, and affinity scatter plot (right) with the confirmed erlotinib reference pose 3 (−7.16 kcal/mol, RMSD 1.64 Å) shown as a dashed red line. (b) Pose browser showing erlotinib redocking pose 3 (cyan) overlaid with the co-crystal reference (magenta) in the py3Dmol viewer; RMSD 1.64 Å relative to the crystal structure confirms successful redocking within the conventional 2.0 Å threshold.

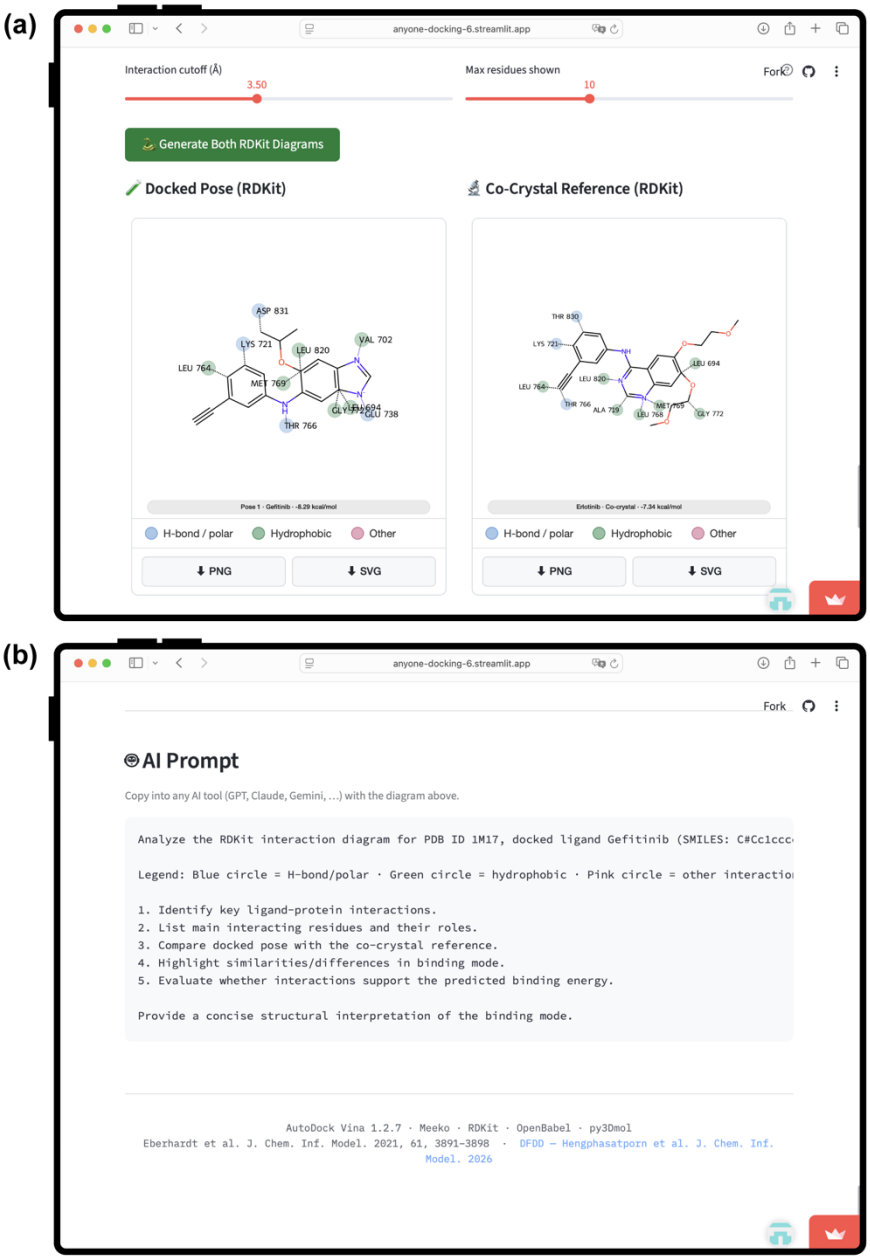

Figure S4. Two-dimensional interaction diagrams and AI-assisted analysis prompt. (a) RDKit-generated 2D interaction diagrams for the top-ranked gefitinib docked pose (left, −8.29 kcal/mol) and the erlotinib co-crystal reference (right, −7.34 kcal/mol), colour-coded by interaction type (blue = H-bond/polar, green = hydrophobic, pink = other). Diagrams were generated locally without external server dependency using the protonated SMILES from Dimorphite-DL. (b) Automatically generated AI analysis prompt, incorporating ligand name, SMILES, binding energy, PDB code, and interaction legend, formatted for direct submission to any large language model to support binding mode interpretation.

Interface Overview: Streamlit Web App, Google Colab Notebook, and Anyone Can Dock GPT

Anyone Can Dock is delivered through three complementary interfaces that implement the same docking pipeline but serve different user profiles (Figure 3 in main text). The Streamlit web application prioritises zero-friction access: four guided steps, no account, no file conversion, and results within minutes from any browser on any device. The Google Colab notebook is a fully self-contained, standalone iPython notebook in which every stage of the pipeline is written as an editable code cell that users can read, modify, copy, and extend directly without any dependency on an external library module. Anyone Can Dock GPT provides a conversational interface for docking setup assistance, workflow guidance, and interpretation of docking results through natural language interaction; it does not execute docking calculations independently but operates as a support layer alongside the Streamlit and Colab interfaces. Both computational interfaces are compared in Table S1.

The web app is appropriate when a result is needed quickly from any device, when the user has no Python background, or when docking 1–20 ligands. The Colab notebook is appropriate when screening more than 20 ligands, GPU acceleration is needed (Uni-Dock), access to additional docking engines (VinaXB, GNINA, Uni-Dock) is required, or when the user wishes to modify, inspect, or reuse pipeline code. Anyone Can Dock GPT is appropriate when the user requires guidance on workflow decisions, interpretation of docking scores, or plain-language explanation of results.

For users who require offline access, institutional data privacy, or repeated high-throughput runs without network dependency, both computational interfaces can alternatively be run on a local machine. The Streamlit app can be launched locally by cloning the repository and running streamlit run app.py in a Python environment with the listed dependencies installed; the Colab notebook can likewise be executed in any local Jupyter environment after installing the same dependency set via the provided requirements.txt. In either case, all docking logic, receptor preparation, and scoring remain identical to their hosted counterparts.

Switching between interfaces, whether hosted or local, does not affect docking accuracy: both use the same receptor preparation logic, the same pKaNET Cloud+-based protonation-state assignment, and the same AutoDock Vina 1.2.7 scoring function.

Table S2 Comparison of the two Anyone Can Dock interfaces. The bottom row lists pipeline components that are identical in both interfaces.

	Streamlit web app	**Google Colab notebook**
Primary user	Any scientist — no coding needed	Computational researchers, developers
Interface type	Point-and-click GUI, 4-step workflow	Standalone iPython notebook, fully transparent
Installation	None — browser only	Cell-based install per session
Account required	No	Google account
Mobile / tablet	Full — tested iOS and Android	Limited
Docking engines	AutoDock Vina 1.2.7	Vina 1.2.7, VinaXB, GNINA 1.3, Uni-Dock (GPU)
Batch size (recommended)	≤ 20 ligands per session	Unlimited
Code visibility	Hidden (black-box interface)	100% visible — every step editable
Copy code for own project	Via GitHub repository only	Directly from any cell
Reproducible notebook format	No	Yes — shareable .ipynb file
Structure drawing	Yes (Ketcher)	No
PoseView 2D diagrams	Yes — auto-generated, side by side	Via API call
AI analysis prompt	Auto-generated	Manual
One-click ZIP download	Yes	Manual from files panel
Shared pipeline (both interfaces)	Metal-preserve receptor prep · Dimorphite-DL → ETKDGv3 → MMFF94 → Meeko · AutoDock Vina 1.2.7 · bond-order correction · RMSD validation	

Extended Ligand Preparation Methodology

This section provides full methodological details for the ligand preparation pipeline referenced in the main text. The pipeline executes four sequential automated stages: protonation state assignment, 3D conformer generation, energy minimisation, and PDBQT conversion.

Stage 1 — Protonation State Assignment (Dimorphite-DL)

Protonation state assignment is performed by Dimorphite-DL,[8] an open-source empirical tool that identifies ionisable functional groups via SMARTS-based substructure matching against a curated internal database of 1,938 experimentally characterised small molecules. Each functional group class carries an empirically derived mean pK_a (μ) and standard deviation (σ), together defining a characteristic pK_a range:

$$range\_pKa = [\ \mu - n\sigma,\quad \mu + n\sigma\ ]$$

where the precision factor *n* (default 1.0) governs the breadth of chemical uncertainty captured. For each ionisable moiety, this range is compared against the user-specified target pH window [pH_{min}, pH_{max}], and one of three outcomes is applied:

- **Deprotonated:** $\mu + n\sigma < pH_{min} \Rightarrow$ moiety is deprotonated
- **Protonated:** $pH_{max} < \mu - n\sigma \Rightarrow$ moiety is protonated
- **Both forms enumerated:** range_pKa $\cap$ range_pH $\neq \emptyset \Rightarrow$ both forms generated

This design allows a single molecule to yield multiple protomers in one pass across all ionisable sites independently, a capability absent in tools such as OpenBabel, which return one output state per input molecule regardless of pH uncertainty. Where multiple protomers are produced, the pipeline selects the single dominant species (`max_variants=1`, `ph_min = ph_max = target_pH`), the protomer whose empirical pK_a range most unambiguously resolves to one side of the target pH, for docking. This AUTO mode matches the behaviour of the pKaNET Cloud Colab implementation. Dimorphite-DL operates from its built-in empirical database without requiring an external pK_a prediction step, distinguishing it from the pKaPredict-dependent workflow of pKaNET Cloud.[6] Accuracy benchmarks across a range of precision factors and ionisable moiety classes are reported in the original publication.[8]

Two limitations merit explicit acknowledgement. First, each ionisable site is evaluated in isolation; long-range electronic coupling between adjacent ionisable groups, such as mutual pK_a perturbation in polyprotic acids or molecules bearing nearby electron-withdrawing substituents, lies outside the scope of the algorithm. Second, Dimorphite-DL addresses ionisation states only and does not enumerate prototropic tautomers, which arise from intramolecular proton transfer rather than net proton gain or loss. Users requiring exhaustive enumeration of tautomers, stereoisomers, or co-existing protonation states are directed to pKaNET Cloud[6] for full protomer preparation prior to docking.

Stage 2 — 3D Conformer Generation (RDKit ETKDGv3)

A three-dimensional starting conformation is generated using the ETKDGv3 distance geometry method[9] implemented in RDKit. ETKDGv3 combines experimentally derived torsional angle distributions from the Cambridge Structural Database with distance geometry embedding and a knowledge-based torsion correction potential, providing an appropriate starting point for force-field refinement. A fixed random seed (`randomSeed=42`) is used for reproducibility. If ETKDGv3 is unavailable in the installed RDKit build, ETKDG is used as a fallback. If embedding fails with default parameters, a second attempt is made with `useRandomCoords=True` to handle molecules where distance geometry fails due to complex ring systems or unusual connectivity. Molecules for which 3D embedding fails after both attempts are flagged with an error and excluded from docking.

Stage 3 — Energy Minimisation (MMFF94 / UFF)

The embedded conformer is energy-minimised using the Merck Molecular Force Field 94 (MMFF94)[10] as implemented in RDKit, with a maximum of 500 iterations. MMFF94 is applied when all force-field parameters are available for the molecule (`AllChem.MMFFHasAllMoleculeParams()` returns True), relaxing steric clashes and refining bond lengths and angles to yield a docking-ready geometry. For molecules outside the MMFF94 parameterisation scope, primarily those containing atoms or bonding patterns not covered by the original training set, the Universal Force Field (UFF) is applied automatically as a fallback to ensure pipeline continuity without user intervention. Both force fields are applied to the fully hydrogenated molecule.

Stage 4 — PDBQT Conversion (Meeko)

The minimised RDKit molecule is converted to PDBQT format using Meeko, the ligand preparation library developed by the AutoDock Vina team. Meeko assigns AutoDock atom types, sets rotatable bonds, handles ring systems, and writes a PDBQT file compatible with AutoDock Vina 1.2.7. The `PDBQTWriterLegacy.write_string()` API is used, with automatic fallback to `prep.write_pdbqt_string()` for older Meeko builds. The formal charge of the prepared ligand, derived from the Dimorphite-DL-assigned protonation state, is calculated and presented to the user alongside 2D and 3D structure previews, enabling chemical verification of the protonation outcome before docking proceeds. The SDF file of the minimised conformer is retained as the SMILES template for bond-order correction after docking.

Accepted Input Formats and Ligand Scope

The pipeline accepts three input modalities:

(i) **SMILES string** — typed or pasted directly. Any valid canonical or isomeric SMILES parseable by RDKit is accepted.

(ii) **Structure file** — SDF, MOL2, or PDB format. The first molecule is extracted, converted to canonical SMILES via RDKit (SDF) or OpenBabel (MOL2, PDB), then passed through the four-stage pipeline. Only the largest fragment is retained if multiple disconnected fragments are present.

(iii) **Ketcher structure drawing** — the in-browser Ketcher editor (Streamlit app only) returns a SMILES string passed directly to the pipeline.

The pipeline is validated for standard drug-like small molecules (MW < 700 Da, Lipinski-compliant chemical space, ≤ 15 rotatable bonds). Reliability for macrocycles, organometallic compounds, prodrugs requiring metabolic activation, and molecules with unusual valence is not guaranteed. Covalent inhibitors and metal-binding pharmacophores where explicit coordination geometry is important require additional manual preparation outside the automated pipeline.

Deployment Instructions

Streamlit Cloud (No Installation Required)

The recommended route for most users is the hosted Streamlit deployment, accessible at the URL below. No account, local installation, or system configuration is required. AutoDock Vina 1.2.7 (ca. 4 MB) is retrieved automatically on first use and cached for the duration of the session.

Live application: https://nyelidl.github.io/anyone-docking/

Status indicator: A green “Vina 1.2.7 ready” pill in the application header confirms a successful environment. A red error banner and application halt indicate a missing OpenBabel installation.

Local Deployment

Linux (Ubuntu / Debian)

```
sudo apt install python3.11 python3.11-venv openbabel libcairo2-dev
libpangocairo-1.0-0 && \
git clone https://github.com/nyelidl/anyone-docking-local.git && \
cd anyone-docking-local && \
python3.11 -m venv venv && \
source venv/bin/activate && \
pip install -r requirements.txt && \
streamlit run app.py
```

macOS

```
sudo apt install python3.11 python3.11-venv openbabel libcairo2-dev
libpangocairo-1.0-0 && \
git clone https://github.com/nyelidl/anyone-docking-local.git && \
cd anyone-docking-local && \
python3.11 -m venv venv && \
source venv/bin/activate && \
pip install -r requirements.txt && \
streamlit run app.py
```

Google Colab

A Colab-compatible notebook is provided in the repository. Open the notebook in Google Colab and execute the installation cell at the top of the notebook prior to running any other cells.

Notebook repository: https://github.com/nyelidl/anyone-docking

Verification of Dependencies

Once all checks pass, the Streamlit application header displays a green “Vina 1.2.7 ready” status pill confirming that AutoDock Vina 1.2.7 has been located and is executable.

Table S3 Representative docking test cases used to evaluate the applicability of the workflow across different target classes. The table summarizes selected PDB structures, corresponding ligands, reported AutoDock Vina 1.2.x or Vina-compatible docking scores from the literature, and their intended use as either validation or demonstration cases. Native-ligand redocking cases were used for direct pose-reproduction assessment, whereas non-native ligand examples were used to demonstrate applicability to broader protein classes, including metalloproteins, cofactor-containing enzymes, and GPCR/membrane-protein targets.

PDB ID	Ligand	Other platform (Vina 1.2)	Anyone Can Dock (Vina 1.2.7)	RMSD vs crystal (Å)	Runtime (minute)
1IEP	Imatinib	−12.71[1]	−12.75	0.82	~2
5LX6	Veliparib	−8.40[2]	−8.68	0.76	~2
4O2B	Colchicine	−10.07[2]	−7.45	1.00	~1
6M2N	Baicalein	−6.70[3]	−6.00	0.67	~1
6O0K	Venetoclax	−10.60[4]	−9.99	1.87	~4
7CFN	Mogrol	−9.52[5]	−8.04	1.89	~2

Table S4 Troubleshooting

Error / symptom	Solution
"Zzzz This app has gone to sleep due to inactivity. Would you like to wake it back up?" in streamlit web application	To wake the application, simply click "*Yes, get this app back up!*"
"OpenBabel not found" red banner on startup	Install OpenBabel: conda install -c conda-forge openbabel or sudo apt-get install openbabel. Restart the Streamlit server after installation.
"Could not download Vina binary"	Check internet connection. If behind a firewall, download vina_1.2.7_linux_x86_64 manually from the AutoDock-Vina GitHub releases page, place at /tmp/vina_1.2.7, and run chmod +x /tmp/vina_1.2.7.
Receptor preparation fails for a metal-containing PDB	Verify the PDB file is valid and not truncated. The metal-safe protocol should handle all standard HETATM metal records. Report the PDB code as a GitHub issue if the problem persists.
PoseView returns error or times out	PoseView is an external REST API (Proteins.plus). Errors indicate temporary service downtime — retry after a few minutes. 3D viewing and all downloads remain available regardless of PoseView status.
"RDKit could not parse SMILES"	Verify the SMILES using an external validator. Common issues: mismatched parentheses, invalid valence, unsupported element symbols. Use the Ketcher drawing tool as an alternative input mode.
Bond-order correction warning flag	Occurs when MCS coverage between docked pose and input SMILES falls below 60%. Common causes: symmetric scaffolds, tautomers, or fragments. The uncorrected pose is returned, inspect the raw SDF before downstream use.
Batch job terminates partway through	On Streamlit Cloud, jobs exceeding session CPU limits are terminated. Reduce batch size to ≤ 20 ligands or lower exhaustiveness to 8. For large batches, use the Google Colab notebook or local deployment.
py3Dmol viewer shows blank or black screen	Browser extensions (ad-blockers, script-blockers) may block WebGL. Try in an incognito window or a different browser. All viewers require WebGL support.
RMSD shows "no reference" indicator	The co-crystal ligand was not detected at the receptor preparation step. Use Manual XYZ or ProDy selection mode for grid centering. RMSD display requires a co-crystal reference identified during receptor preparation.

Reference

1. Polák L, Škoda P, Riedlová K, Krivák R, Novotný M, Hoksza D: **PrankWeb 4: a modular web server for protein–ligand binding site prediction and downstream analysis**. *Nucleic Acids Res* 2025, **53**(W1):W466-W471.
2. Boualia B, Sandeli Ae-K, Evren E, Özdemir İ, Karci H, Dündar M, Özdemir İ, Gürbüz N, Koç A, Boulebd H: **Synthesis, characterization, and anticancer potential of novel NHC ligands and their selenium complexes: a combined in vitro and in silico investigation**. *RSC Advances* 2026, **16**(1):409-424.
3. Sandalli C, Bostancioglu SM, Sandalli A, Turumtay EA, Almohazey D, Alsaeed M, Alhamid G, Rabaan AA, Turumtay H, Ami E-i *et al*: **Targeting SARS-CoV-2 main protease (3CLpro) with Paeonia-derived phytochemicals**. *In Silico Pharmacology* 2026, **14**(2):122.
4. Almansour NM, Allemailem KS, Abd El Aty AA, Ismail EIF, Ibrahim MAA: **In Silico Mining of Natural Products Atlas (NPAtlas) Database for Identifying Effective Bcl-2 Inhibitors: Molecular Docking, Molecular Dynamics, and Pharmacokinetics Characteristics**. *Molecules* 2023, **28**(2).
5. Tanaka C, Harada N, Teraoka Y, Urushizaki H, Shinmori Y, Onishi T, Yotsumoto Y, Ito Y, Kitakaze T, Inui T *et al*: **Mogrol stimulates G-protein-coupled bile acid receptor 1 (GPBAR1/TGR5) and insulin secretion from pancreatic β-cells and alleviates hyperglycemia in mice**. *Scientific Reports* 2024, **14**(1):3244.