Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Master Thesis

Ligand Transplantation and Optimization in Protein Models

Submitted by

Marius Rüve
Student number 7657327

in the study program

Master of Science Bioinformatics
at the Center for Bioinformatics Hamburg
and the Department of Informatics
of the MIN Faculty

September 13, 2025

1. Examiner: Prof. Dr. Matthias Rarey
2. Examiner: Prof. Dr. Andrew Torda

Abstract

Todo: Fill a 200–250 word abstract summarizing problem, method (FoldFusion), key results (quantitative), limitations, and contributions.

Todo: Ensure advisor names/titles and matriculation number are correct and match university records.

Todo: Decide on final chapter structure: currently Introduction, Methods, Results, Conclusion. Remove references to non-existent chapters (Background, Discussion) or add those chapters.

Todo: Move final figures into thesis/figures as PDF/SVG (preferred) or 300 DPI PNG; reference them in Methods/Results.

Todo: Add an Appendix section with exact reproducibility details: environment (Python version, key packages), commands to run pipeline, git commit hash, dataset versions, and config.toml snapshot.

Todo: Run full LaTeX compile and resolve all warnings: no undefined refs/citations, consistent figure/table numbering, proper hyphenation.

Test text

Contents

1 Chapter 1. Introduction

Proteins are the primary effectors of cellular function, and understanding their three-dimensional (3D) structures is essential for mechanistic insight, target identification, and rational design in biotechnology and drug discovery. For decades, experimental structure determination by X-ray crystallography, nuclear magnetic resonance (NMR) spectroscopy, and cryo-electron microscopy (cryo-EM) has provided atomic-level information, but at significant cost and with notable coverage gaps [1]. Recent breakthroughs in machine learning, most prominently *AlphaFold* and *RoseTTAFold*, have transformed the landscape by enabling accurate in silico structure prediction directly from amino-acid sequences, dramatically increasing the number of available models across many proteomes [2, 3, 4]. These advances have catalyzed a shift from "structure scarcity" to a new phase in which the central challenges concern structural context, functional annotation, and downstream usability of predicted models [5].

In this thesis, a distinction is made between *experimental structural models* (determined by X-ray, NMR, or cryo-EM and deposited in the PDB [6]) and *predicted models* (e.g., AlphaFold, RoseTTAFold). Experimental structures frequently capture proteins in a *holo* state, i.e., with bound small molecules such as metal ions, cofactors, or ligands; by contrast, predicted models are provided as *apo* proteins without non-polymer entities by design. The term *ligands* is used to refer broadly to small molecules in binding sites, including organic cofactors (e.g., heme, FAD, NAD(P)H) and mono- or polyatomic ions (e.g., Zn^{2+}, Mg^{2+}). Because many proteins require such moieties to stabilize their fold or to realize catalysis, the absence of these entities in predicted structures can obscure function, binding-site chemistry, and downstream computability. Moreover, predicted models include per-residue confidence scores (e.g., pLDDT in AlphaFold) that correlate with

local reliability and often flag flexible or poorly constrained regions, reinforcing the need to reason about context before analysis [2, 6, 7].

1.1. Motivation

Predicted single-chain protein models typically represent the polypeptide backbone and side chains under canonical residue chemistry [8]. However, many proteins only adopt their native fold or functional state in the presence of small molecules: metal ions that stabilize architecture, organic cofactors that mediate catalysis, and physiologically relevant ligands such as ATP, heme, or NAD(P)H [9, 10]. In predicted models, these moieties are absent by design, which complicates functional interpretation, binding-site analysis, and computational follow-ups such as molecular docking or molecular dynamics. Moreover, the conformational state of a predicted model is often not annotated, and flexible regions may be modelled with lower confidence, further widening the gap between prediction and experimental use in structure-guided workflows. Current (2025) advances such as AlphaFold 3 extend the predictive scope to joint modeling of proteins, nucleic acids, ions, small molecules, and certain modifications. However, no publicly downloadable, proteome-scale repository of such context-rich AF3 models exists, and routine large-scale workflows still rely on AlphaFold 2–derived apo predictions [11]. Consequently, practical enrichment of predicted structures with biochemically sensible ligands remains an unmet need.

As a concrete example, protein kinases and many ATPases require Mg^{2+}-ATP in the active site to position catalytic residues and neutralize charge; in an apo predicted model, the nucleotide pocket may appear collapsed or incorrectly polarized, hampering binding-site recognition and leading to unrealistic docking poses. Similarly, heme enzymes without the porphyrin cofactor or zinc-finger domains without Zn^{2+} can appear destabilized or ambiguous with respect to their functional geometry. Restoring these entities from homologous experimental structures often reveals the correct local architecture and plausible interaction networks, enabling more faithful functional hypotheses and more realistic computational experiments [7].

1.2. Problem Statement

A central question is how to systematically enrich predicted protein structures with plausible, biochemically sensible small molecules and ions to make them more useful for functional reasoning and downstream computation, without

requiring de novo quantum mechanics or extensive experimental input for each target. The central idea explored in this thesis is to exploit the wealth of biophysical knowledge already contained in experimentally determined structures deposited in the Protein Data Bank (PDB) [6] by transferring ("transplanting") ligands and cofactors from suitable homologues into corresponding predicted models.

1.3. Prior Work and Opportunity

A closely related line of work demonstrates that large-scale ligand transplantation from experimentally solved homologous structures into AlphaFold models is both feasible and useful. For example, the *AlphaFill* resource applies sequence and structure similarity to transplant common ligands, cofactors, and metal ions from curated experimental models, validating quality with metrics such as local RMSD and a transplant-clash score. At scale, AlphaFill reported over twelve million transplants across nearly one million AlphaFold models, exposing binding sites, restoring essential cofactors (e.g., heme, Zn^{2+}, Mg^{2+}), and enabling hypothesis generation about function [7]. While this demonstrates the promise of homology-driven enrichment, there remains a practical need for open, lightweight, and extensible pipelines that researchers can adapt to bespoke datasets, integrate into modern data/ML workflows, and evaluate end-to-end on specific biological questions.

This opportunity is addressed with a binding-site-guided, modular pipeline that differs from prior resources in several ways. First, candidate donor structures are prioritized via binding-site detection to focus alignment on functionally relevant pockets (using *DoGSite3* [12]). Second, site-centric structural alignment is performed to position donor sites relative to the predicted target (via *SIENA* [13]). Third, transplanted ligands are optionally refined with a simple pose optimization/scoring step (*JAMDAScorer* [14]) to reduce clashes and improve geometry. Finally, each placement is annotated with quality indicators, *Local RMSD* in the ligand environment and a *Transplant Clash Score (TCS)*, to support downstream triage and decision-making. These choices emphasize transparency, configurability, and ease of integration with scripting- and batch-oriented research workflows.

1.4. Thesis Aim

This thesis presents a pipeline that automates ligand and cofactor transplantation from the PDB into AlphaFold structures to produce *context-enriched* models suitable for exploratory analysis and computational follow-up.[1] At a high level, the pipeline (i) identifies homologous donor structures with relevant non-polymer entities, (ii) performs global and local structural alignments to position candidate ligands, (iii) applies simple but effective filters to prioritize biochemically sensible placements, and (iv) emits enriched models and metadata for quality control and downstream use.

Throughout, *Local RMSD* (a measure of structural agreement in the protein environment around the ligand) and a *Transplant Clash Score* (a measure of steric overlap between ligand and protein) are reported as lightweight indicators of placement quality; full definitions appear in Chapter 2. The pipeline is evaluated on a representative set of UniProt targets assembled for this study, and outcomes are summarized in Chapter 3.

1.5. Contributions

The main contributions of this thesis are:

1. **A modular, open pipeline for ligand transplantation.** The pipeline implements an end-to-end workflow that ingests predicted structures, retrieves homologous experimental entries, and transplants non-polymer entities (ligands, cofactors, metal ions) with provenance tracking and reproducible configuration.

2. **Quality indicators and metadata for downstream trust.** The pipeline annotates each transplant with alignment measures and simple clash checks, enabling users to stratify placements by confidence and decide when refinement is warranted.

3. **Empirical evaluation on representative targets.** The evaluation illustrates use cases where enrichment adds value for functional interpretation (e.g., revealing cofactor requirements or likely substrate preferences) and for computation (e.g., seeding docking with realistic binding-site chemistry).

[1] A public code repository accompanies this work (`https://github.com/mariusrueve/foldfusion`).

4. **Engineering for reproducibility and integration.** The codebase is designed for batch execution, scripted analysis, and integration with common structural bioinformatics tools, easing adoption in research settings.

1.6. Research Questions

The following questions guide this thesis:

- **RQ1:** To what extent can homolog-based ligand transplantation reliably restore biochemically plausible small-molecule context in predicted structures?
- **RQ2:** Which alignment and filtering criteria most influence transplant quality, and how should they be configured in practice?
- **RQ3:** How does structural enrichment affect downstream tasks such as docking preparation, site annotation, or hypothesis generation about protein function?

1.7. Scope and Limitations

The pipeline focuses on non-polymer ligands and metal ions commonly represented in the PDB. It does not perform full flexible-receptor docking or quantum refinement, and it does not attempt to model post-translational modifications or glycans. As with any homology-based approach, transplant reliability depends on the availability and quality of structurally similar donors, as well as on the conformational compatibility between donor and acceptor. Enriched models should be treated as *qualitative* hypotheses that can guide experiments or more detailed simulations, rather than as final, quantitatively precise holo structures [7].

1.8. Thesis Structure

Chapter 2 details the pipeline design and implementation. Chapter 3 reports evaluation on diverse targets and discusses quality indicators and failure modes. Chapter 4 concludes and outlines avenues for future work, including refinement protocols and multi-state/complex modelling.

Chapter 2.

2 Methods

2.1. Overview

This work implements a modular pipeline for systematic augmentation and assessment of predicted protein structures using experimentally resolved homologous complexes and associated ligands. The architecture consists of (i) structured acquisition of target and donor structural data, (ii) detection and characterization of binding environments, (iii) sequence/structure alignment and geometric mapping, (iv) ligand and pocket feature transplantation with conflict resolution, (v) multi-criteria filtering and scoring, and (vi) standardized evaluation and visualization of resulting models and annotations. This layered decomposition promotes separation of concerns, controlled extensibility, and reproducible execution.

Inputs consist of one or more target protein identifiers (e.g., UniProt accessions) whose corresponding predicted three-dimensional models are retrieved (or validated if cached) and normalized. Donor candidates are gathered from curated structural repositories and homolog databases according to configurable similarity, coverage, and quality constraints. Auxiliary metadata (ligands, binding site residues, cavity descriptors, and experimental provenance) are harvested through dedicated tool adapters. Each external data source is encapsulated behind a dedicated integration component to isolate protocol specifics from core orchestration logic.

Evaluation utilities quantify improvement and reliability through metrics such as structural alignment quality, pocket conservation, ligand retention integrity, and scoring distribution profiles. Visualization components generate comparative plots and summary artifacts to support interpretability and methodological

diagnosis. Configuration management and structured logging ensure parameter traceability, deterministic re-runs, and auditability of intermediate decisions. The overall design facilitates insertion of new data sources, scoring strategies, or analytical endpoints with minimal impact on existing workflow stages.

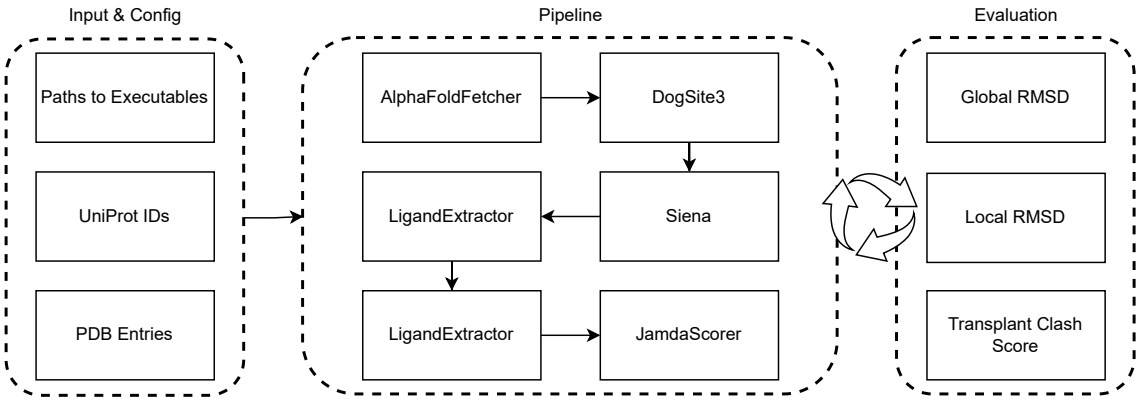

Figure 2.1.: High-level pipeline stages: data acquisition; binding site and ligand characterization; alignment and geometric mapping; transplantation; filtering and scoring; evaluation and visualization.

2.2. Data and Inputs

Todo: Describe the input protein models (AlphaFold/UniProt IDs), how they are fetched, and the benchmark dataset in evaluation/data (size, selection criteria). Include table summarizing proteins used.

Todo: Specify configuration via config.toml: key parameters (alignment thresholds, filters), seeds, and runtime knobs.

2.3. Components and Tools

2.3.1. AlphaFold Fetcher

Predicted target structures are sourced from the public AlphaFold Protein Structure Database [2, 11]. For each UniProt accession U, the fetcher attempts to download the most recent single-chain model file in Protein Data Bank (PDB) format using the canonical URL pattern:

`https://alphafold.ebi.ac.uk/files/AF-`U`-F1-model_v4.pdb`

If version v4 is not available (HTTP 404), a controlled fallback to model_v3 is performed. This ordered fallback (v4 $\rightarrow$ v3) guarantees maximal currency while preserving compatibility with accessions that have not yet been reprocessed in later database releases. Network and HTTP errors are logged with structured messages; only a definitive failure after exhausting all configured versions raises a terminal exception. The module is deliberately side effect free beyond file system writes and thus supports higher-level retry logic if desired.

All AlphaFold downloads for a run are stored under a deterministic cache directory AlphaFold/ inside the tool's designated output root to enable re-use across downstream stages and future executions. File names are preserved verbatim from the source (e.g., AF-<UniProt>-F1-model_v4.pdb) to maintain traceability to the original database asset. A processed companion file suffixed _processed.pdb is generated (see below), allowing both raw and normalized forms to coexist. While the current implementation always overwrites existing files (ensuring freshness), the presence of both raw and processed variants provides an implicit cache that higher orchestration layers may exploit to skip redundant network access.

Confidence-Guided Segment Trimming. AlphaFold supplies per-residue predicted Local Distance Difference Test (pLDDT) scores encoded in the B-factor column of each ATOM record. Rather than discarding individual residues below a threshold (which can produce fragmented backbones and complicate pocket computations), the fetcher applies a conservative *segment-based* pruning strategy: any contiguous stretch of ≥ 5 residues where pLDDT < 50 for all members is removed in its entirety. The threshold of 50 aligns with AlphaFold's published qualitative confidence bands ("low" confidence boundary) and has been widely adopted to demarcate regions likely to be disordered or spuriously modeled; requiring a minimal run length of five suppresses noise from isolated low-scoring residues embedded in otherwise reliable context. This approach preferentially excises flexible termini or unresolved loops while preserving internal residues that might transiently dip below the cutoff.

Implementation-wise, the parser performs two passes over the PDB text: (i) extraction of an ordered residue identifier and associated pLDDT stream (residue identity is defined by concatenated residue name, chain identifier, and sequence number parsed from columns 18–26), and (ii) identification and marking of qualifying low-confidence segments. A second traversal writes out all ATOM/HETATM lines not belonging to flagged segments. Non-coordinate records (HEADER, REMARK, etc.) are retained verbatim to preserve provenance metadata. The output is emitted to AF-<UniProt>-F1-model_vX_processed.pdb (same directory) enabling deterministic downstream referencing.

Normalization and Structural Assumptions. AlphaFold monomer models are emitted as a single chain (commonly chain `A`); alternative location indicators and insertion codes are not expected and are therefore passed through without modification if present. No coordinate re-numbering or chain renaming is performed, preserving direct comparability with original residue numbering and external annotations (e.g., UniProt feature mappings). Because trimming physically removes ATOM records, residue numbering may exhibit gaps; downstream components treat numbering symbolically and do not require continuity. The fetcher does not currently remap sequence indices nor generate masks; omission is intentional to keep a faithful structural subset rather than a padded representation.

Error Handling and Logging. Each retrieval attempt logs: (i) target accession and version, (ii) URL, (iii) success or specific failure classification (404 vs. other HTTP vs. network). Processing anomalies (file read/write errors, malformed ATOM records) are trapped and surfaced with contextual messages while falling back to the unprocessed file if trimming cannot be completed safely. This defensive design prevents a single malformed record from aborting an otherwise usable structure.

Limitations and Future Extensions. The current segment detection logic infers residue boundaries from repeating per-atom records and assumes uniform pLDDT within a residue (fulfilled in AlphaFold outputs). Multi-chain assemblies (e.g., AlphaFold multimer) and partial models are out of scope but could be supported by iterating per chain. Optional future enhancements include: checksum-based cache validation to avoid redundant downloads, configurable pLDDT thresholds and minimal segment length, generation of a residue-level confidence mask, and integration of model confidence metrics (PAE matrices) for region-aware downstream weighting.

Overall, this module provides a reproducible, transparent acquisition and normalization layer that supplies structurally cleaner models to pocket detection and alignment stages while retaining provenance and enabling straightforward parameterization.

2.3.2. Binding Site Prediction (DoGSite3)

Binding pocket detection on the (optionally confidence-trimmed) AlphaFold target model is performed with DoGSite3 [12], a reimplementation of the original DoGSite approach employing a Difference-of-Gaussians (DoG) filtering scheme on a three-dimensional grid discretization of the solvent-accessible protein envelope.

The method locally contrasts a pair of Gaussian-smoothed molecular surface occupancy grids at different scales to accentuate cavity voxels, followed by region growing, connectivity refinement, and physicochemical descriptor calculation. Compared to its predecessor, DoGSite3 emphasizes robustness in the presence of bound ligands and yields reproducible pocket rankings with reduced parameter sensitivity (see publication for benchmarked performance characteristics). We treat its ranked pocket list as a proxy for ligandable micro-environments in subsequent donor alignment.

Invocation and Output Layout. Execution is wrapped by the Dogsite3 tool adapter (see implementation in `foldfusion/tools/dogsite3.py`), which constructs a deterministic command line:

```
<dogsite3> --proteinFile <ABSOLUTE_PDB_PATH> --writeSiteResiduesEDF
```

where `<dogsite3>` denotes the configured executable path and the input PDB path is resolved to an absolute canonical location prior to invocation. All artifacts are written into a run-scoped directory `Dogsite3/` underneath the tool's designated output root (one directory per target). The `--writeSiteResiduesEDF` flag instructs DoGSite3 to emit per-pocket EDF ("Extended Descriptor File") records that include residue membership annotations in addition to geometric and physicochemical attributes.

Pocket Selection and EDF Normalization. For each target we currently retain only the top-ranked pocket as defined by DoGSite3's internal scoring (exposed as index 1 in the output file name pattern). The corresponding EDF is expected at `Dogsite3/output_P_1_res.edf`.

Immediately post-run the adapter performs an idempotent sanitization step: if the EDF header line `REFERENCE <NO-FILE>` is present, it is replaced in-place with the absolute path to the analyzed PDB file. This guarantees downstream tools (notably the SIENA binding-site similarity search) have a resolvable pointer without relying on relative directory context. No other EDF fields are modified, and the operation is skipped if a valid `REFERENCE` already exists.

Extracted Features Consumed Downstream. The EDF encapsulates a mixture of scalar descriptors and residue-level membership annotations. While DoGSite3 enumerates an extended descriptor set, the current pipeline utilizes the following subset in later decision functions and reporting:

- Geometric: pocket volume, surface area, effective radius / diameter, and asphericity—used to characterize cavity size and shape when comparing donor pockets or rationalizing ligand accommodation.
- Physicochemical composition: hydrophobic surface fraction, polar surface fraction, hydrogen bond donor/acceptor counts, aromatic residue count—informing plausibility of transplanted ligand physicochemical complementarity.
- Accessibility: mouth opening area and enclosure metrics—qualitatively referenced when interpreting steric clash filtering outcomes.
- Ranking / score: the native DoGSite3 pocket score employed only implicitly (selection of rank 1 pocket) but retained for audit.
- Residue membership list: chain identifiers and residue numbers forming the pocket environment; projected onto alignment results to compute overlap and to guide ligand clash resolution.

Additional descriptors present in the EDF (e.g., curvature-derived terms) are stored but not presently incorporated into filtering heuristics; preservation ensures future extension does not require regeneration of pockets. All EDF files are version-stamped implicitly through inclusion of the upstream executable version (recorded in the structured log) facilitating reproducibility and cross-run comparison.

In preliminary internal tests (evaluating downstream SIENA hit quality, ligand transplantation success, and final filtering yield) inclusion of additional pockets beyond rank 1 (evaluated up to the top 5) produced no measurable improvement while increasing wall-clock time. Consequently, to avoid unnecessary runtime expansion and complexity, the pipeline currently restricts processing to the single top-ranked pocket.

Overall, this stage converts a raw target structure into a normalized pocket descriptor and residue environment bundle with explicit provenance, enabling reproducible, tool-version aware propagation of binding-site context into alignment and ligand transfer stages.

2.3.3. Donor Databases and Retrieval (PDB/UniProt/SIENA DB)

Donor structures (experimental templates and their cognate ligands) are sourced from a locally mirrored slice of the Protein Data Bank (PDB) [6]. Mirroring is performed via `rsync` using the template script distributed by RCSB (adapted in `scripts/rsyncPBD.sh`); only the classic PDB coordinate hierarchy `data/structures/divided/pd`

is synchronized to reduce footprint. We employ the RCSB public rsync endpoint (port 33444) and capture the mirror date/time stamp in the run metadata. The mirror path (default `/home/marius/Data/pdb`) is treated as read-only by the pipeline. Periodic refresh (manually triggered) ensures currency while preserving deterministic snapshots between updates. Each run logs the PDB snapshot date plus the total number of mirrored entries, enabling exact reconstruction of the donor corpus.

UniProt accessions serve as the primary identifiers for targets (AlphaFold models) and provide linkage to experimental structures through canonical mapping tables (implicit in AlphaFold naming conventions). While a separate UniProt flat file is not parsed here, consistency of residue numbering and chain semantics is preserved by deferring any sequence-based reconciliation to later alignment stages; provenance of donor sequences thus remains anchored to their deposited PDB records.

To accelerate binding-site similarity searches, a SIENA structure database is built (or re-used if present) using the SienaDB tool wrapper (implementation: `foldfusion/tools/siena_db.py`), referencing the method described in [13]. Database creation indexes residue-kmer encodings across all mirrored PDB entries, enabling sub-second retrieval of alternative binding site conformations that share local sequence/structural motifs with the query pocket. The generator is invoked once per run with arguments:

```
<siena_db_build> --database <NAME> --directory <MIRROR_PDB_DIR>
                          --format <f>
```

where `<f>` is `1` for plain `.pdb` files (current configuration) or `0` if operating on compressed legacy `.ent.gz` layouts. The wrapper accepts an optional explicit path; otherwise it defaults to `<output>/SienaDB/siena_db`. Post-build, the code accommodates ambiguity in produced filenames (with or without a `.db` suffix) by detecting and adopting the existing variant to avoid downstream path mismatches.

Regeneration and Validity Checks. Prior to invocation, the wrapper performs existence and plausibility validation: (i) file presence, (ii) regular file type, and (iii) a minimal size threshold ($< 1KB$ flagged as invalid). If these criteria are satisfied, regeneration is skipped and the existing index reused, ensuring stable identifiers and minimizing unnecessary I/O. Otherwise, a fresh index is created in-place. This behavior enforces idempotence: repeated runs over an unchanged PDB mirror are constant-time with respect to database preparation.

Directory Layout and Provenance. All generated index artifacts reside under `SienaDB/` inside the run's output root (or a user-specified directory if an absolute database path is given). Structured logs record: executable path and (if available) version string, build timestamp, input directory hash (optional future extension), database file size, and number of processed PDB entries. These annotations collectively permit reproducibility assessments and forensic reconstruction of search scope.

Donor Structure Filtering. A light-weight, pre-alignment filtering regime is applied conceptually (some criteria enforced implicitly by tool behaviour, others by configuration) to exclude low-informative or incompatible donors:

- **Experimental resolution:** Structures exceeding a configured maximal resolution (e.g., $\geq 3.0\mathring{A}$; crystallography only) are candidates for exclusion to reduce noise from poorly resolved active sites.
- **Polymer type:** Only protein chains (standard amino acid polymers) are considered; nucleic acid or hybrid complexes contribute coordinates but are not indexed as donors for binding site transplantation.
- **Ligand presence:** Preference is given to entries containing at least one non-solvent, non-polymer ligand within the pocket vicinity (enabling later ligand transfer); purely apo conformations remain permissible when needed for conformational diversity.
- **Model completeness:** Grossly truncated chains or entries dominated by unresolved (missing) residues can be pruned (future automated metric—currently manual oversight via log summaries).
- **Redundancy control:** Intrinsic redundancy is implicitly mitigated by SIENA's k-mer indexing; optional explicit sequence identity clustering (not yet enabled) is noted as a scalable extension.

At this stage, filtering is conservative to preserve conformational breadth; more stringent quality and conflict assessments occur after pocket-level alignment and ligand transplantation.

Ligand Data. Ligands are leveraged directly from donor PDB HETATM records; no external ligand database normalization is imposed in this stage. Retained hetero groups exclude water and common crystallization additives (handled via a configurable exclusion list, future explicit parameterization). Ligand identifiers and residue numbering propagate unchanged, enabling traceable mapping during transplantation and optimization (JAMDA stage).

Versioning and Snapshot Notation. Each results set (alignments, transplanted ligands, scores) references: (i) PDB snapshot date, (ii) SIENA database file checksum (future extension), and (iii) AlphaFold model version for the target. This triad forms the minimal provenance key required to reproduce donor retrieval decisions. Inclusion of exact rsync command options (logged verbatim from `rsyncPBD.sh`) further aids auditability.

In summary, this component assembles a high-quality yet breadth-preserving donor search space through a reproducible mirror, a lazily materialized SIENA index, and light-touch structural filtering, establishing the foundation for subsequent pocket similarity search and targeted ligand transplantation.

2.3.4. Binding-Site Similarity Search and Alignment (SIENA)

Todo: Describe inputs (EDF pocket from DoGSite3) and the SIENA search/alignment workflow; include citation and tool version.

Todo: Document command parameters used: `--edf`, `--database`, `--output .`, **identity cutoff (**`--identity 0.85`**); justify chosen thresholds.**

Todo: Explain result parsing: reading `resultStatistic.csv` **(semicolon separated), column cleanup, sorting primarily by Active site identity (desc), then Backbone and All-atom RMSD (asc).**

Todo: Clarify use of `ensemble/.pdb` **files for downstream ligand extraction/transplantation and how many top alignments are retained (configurable** `siena_max_alignments`**).**

2.3.5. Ligand Optimization and Scoring (JAMDA)

Todo: Summarize JAMDA's role: energy-based pose optimization/scoring of transplanted ligands; include citation and tool version.

Todo: Describe invocation pattern per ligand: inputs (AlphaFold PDB, ligand SDF), outputs (optimized SDF), and flags used (`--optimize`**); output organization (**`JamdaScorer/<PDB>/ligand_id.sdf`**).**

Todo: State what scores/outputs are captured and how they inform filtering (e.g., clash reduction, score improvements). Cross-reference concrete thresholds in Filtering and Scoring.

2.3.6. Alignment

Todo: Describe global/local alignment methods used; define RMSD/score variants and parameters.

2.3.7. Transplantation

Todo: Explain coordinate mapping, handling of alternate locations, protonation states, and chain/residue mapping.

2.3.8. Filtering and Scoring

Todo: Define clash checks, distance cutoffs, and quality indicators; justify chosen thresholds. Reference how SIENA metrics (identity, RMSDs) and JAMDA outputs contribute to accept/reject decisions.

2.4. Implementation Details

Todo: Summarize code structure (packages in foldfusion/), logging, CLI entry points, and dependency management (pyproject/uv.lock).

Todo: Discuss error handling and retries for external resources; note offline/cached mode.

2.5. Reproducibility

Todo: Document exact environment (Python version, key packages), compute hardware, and commands to reproduce results. Reference Appendix for full config and commit hash.

2.6. Ethical and Licensing Considerations

Todo: Note database/tool licenses and usage restrictions; acknowledge limitations of homology-based inference.

3 Chapter 3. Results

3.1. Evaluation Setup

Todo: Briefly restate dataset and metrics. Point to Methods for details; specify commit hash and config used for final runs.

3.2. Overall Transplantation Performance

Todo: Report counts: number of targets, donors found, ligands considered, successful transplants. Include a summary table.

3.3. Quality Indicators

Todo: Present distributions of RMSD, clash scores, and confidence indicators. Add figure placeholders.

3.4. Alignment Quality Analysis

Todo: Insert analysis and figure from `scripts/evaluation_visualisations.py`. Explain trends and outliers.

3.5. Optimization and Filtering Effects

Todo: Show before/after metrics; explain trade-offs. Include relevant figures (`optimization_*`).

3.6. Case Studies

Todo: Pick 2–3 representative proteins; show visualizations (ligand placement), discuss plausibility and caveats.

3.7. Failure Modes

Todo: Describe common errors (misaligned donors, steric clashes, metal coordination errors) with examples and how filters mitigate them.

3.8. Ablations (optional)

Todo: If available, compare alternative alignment thresholds or scoring components. Summarize in a small table.

Chapter 4. Conclusion

4.1. Summary of Findings

Todo: Summarize key results and what they imply for ligand transplantation into predicted protein models. Reference main figures/tables.

4.2. Limitations

Todo: State major limitations: dependence on donor availability/quality, alignment sensitivity, metal coordination accuracy, protonation/tautomer issues.

4.3. Future Work

Todo: List concrete next steps: refinement with energy minimization, multi-state modelling, better scoring for metals, integration with docking/MD pipelines.

4.4. Concluding Remarks

Todo: Provide a concise take-home message and broader impact.

References

[1] Gabriel Monteiro da Silva et al. "High-Throughput Prediction of Protein Conformational Distributions with Subsampled AlphaFold2". In: *Nature Communications* 15.1 (Mar. 27, 2024), p. 2464. ISSN: 2041-1723. DOI: 10.1038/s41467-024-46715-9. URL: https://www.nature.com/articles/s41467-024-46715-9 (visited on 09/13/2025).

[2] John Jumper et al. "Highly Accurate Protein Structure Prediction with AlphaFold". In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2. URL: https://www.nature.com/articles/s41586-021-03819-2 (visited on 09/05/2025).

[3] Minkyung Baek et al. "Accurate Prediction of Protein Structures and Interactions Using a Three-Track Neural Network". In: *Science* 373.6557 (Aug. 20, 2021), pp. 871–876. DOI: 10.1126/science.abj8754. URL: https://www.science.org/doi/10.1126/science.abj8754 (visited on 09/05/2025).

[4] Kathryn Tunyasuvunakool et al. "Highly Accurate Protein Structure Prediction for the Human Proteome". In: *Nature* 596.7873 (Aug. 2021), pp. 590–596. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03828-1. URL: https://www.nature.com/articles/s41586-021-03828-1 (visited on 09/05/2025).

[5] Eduard Porta-Pardo et al. "The Structural Coverage of the Human Proteome before and after AlphaFold". In: *PLOS Computational Biology* 18.1 (Jan. 24, 2022), e1009818. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1009818. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009818 (visited on 09/13/2025).

[6] Stephen K Burley et al. "RCSB Protein Data Bank: Biological Macromolecular Structures Enabling Research and Education in Fundamental Biology, Biomedicine, Biotechnology and Energy". In: *Nucleic Acids Research* 47.D1 (Jan. 8, 2019), pp. D464–D474. ISSN: 0305-1048. DOI: 10.1093/nar/gky1004. URL: https://doi.org/10.1093/nar/gky1004 (visited on 09/05/2025).

[7] Maarten L. Hekkelman et al. "AlphaFill: Enriching AlphaFold Models with Ligands and Cofactors". In: *Nature Methods* 20.2 (Feb. 2023), pp. 205–213. ISSN: 1548-7105. DOI: 10.1038/s41592-022-01685-y. URL: https://www.nature.com/articles/s41592-022-01685-y (visited on 09/05/2025).

[8] *AlphaFold Server*. URL: https://alphafoldserver.com/faq#what-is-the-difference-between-alphafold-server-and-the-alphafold-database (visited on 09/13/2025).

[9] Jia Tang et al. "Characterization of Cofactor-Induced Folding Mechanism of a Zinc Binding Peptide Using Computationally Designed Mutants". In: *Journal of molecular biology* 389.1 (May 29, 2009), pp. 90–102. ISSN: 0022-2836. DOI: 10.1016/j.jmb.2009.03.074. PMID: 19361525. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2792901/ (visited on 09/13/2025).

[10] Wusheng Xiao et al. "NAD(H) and NADP(H) Redox Couples and Cellular Energy Metabolism". In: *Antioxidants & Redox Signaling* 28.3 (Jan. 20, 2018), pp. 251–272. ISSN: 1523-0864. DOI: 10.1089/ars.2017.7216. PMID: 28648096. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5737637/ (visited on 09/13/2025).

[11] Josh Abramson et al. "Accurate Structure Prediction of Biomolecular Interactions with AlphaFold 3". In: *Nature* 630.8016 (June 2024), pp. 493–500. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07487-w. URL: https://www.nature.com/articles/s41586-024-07487-w (visited on 09/05/2025).

[12] Joel Graef, Christiane Ehrt, and Matthias Rarey. "Binding Site Detection Remastered: Enabling Fast, Robust, and Reliable Binding Site Detection and Descriptor Calculation with DoGSite3". In: *Journal of Chemical Information and Modeling* 63.10 (May 22, 2023), pp. 3128–3137. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.3c00336. URL: https://doi.org/10.1021/acs.jcim.3c00336 (visited on 09/05/2025).

[13] Stefan Bietz and Matthias Rarey. "SIENA: Efficient Compilation of Selective Protein Binding Site Ensembles". In: *Journal of Chemical Information and Modeling* 56.1 (Jan. 25, 2016), pp. 248–259. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.5b00588. URL: https://doi.org/10.1021/acs.jcim.5b00588 (visited on 09/05/2025).

[14] Florian Flachsenberg et al. "A Consistent Scheme for Gradient-Based Optimization of Protein–Ligand Poses". In: *Journal of Chemical Information and Modeling* 60.12 (Dec. 28, 2020), pp. 6502–6522. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.0c01095. URL: https://doi.org/10.1021/acs.jcim.0c01095 (visited on 09/05/2025).

A Appendix A. Reproducibility Details

Todo: Insert environment details: OS, Python version, key packages with versions, hardware.

Todo: Insert exact commands (Makefile targets or CLI) used to generate results and figures.

Todo: Record git commit hash and config snapshot used for final evaluation.

B Appendix B. Supplementary Tables and Figures

Todo: Add tables summarizing dataset proteins, donors, ligands, and transplant success by category.

Todo: Include any extended figures not in the main text.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich vorliegende Masterarbeit im Studiengang Bioinformatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, den 13. September 2025

_____

Marius Rüve

Ich bin mit einer Einstellung der Masterarbeit in den Bestand der Bibliothek des Departments Informatik einverstanden.

Hamburg, den 13. September 2025

_____

Marius Rüve