REFRAMING RETRIEVAL-AUGMENTED GENERATION FOR IN SILICO OPTIMIZATION OF ANTIBODY SOLUBILITY

Lena Erlach
Department of Biosystems Science and Engineering
ETH Zürich
Basel, Switzerland
lena.erlach@bsse.ethz.ch

Rohit Singh
Department of Biostatistics & Bioinformatics
Duke University
Durham, NC, US
rohit.singh@duke.edu

Bonnie Berger
CSAIL
Massachusetts Institue of Technology
Cambridge, MA, US
bab@csail.mit.edu

Sai T. Reddy
Department of Biosystems Science and Engineering
ETH Zürich &
Botnar Institue for Immune Engineering
Basel, Switzerland
sai.reddy@bsse.ethz.ch

ABSTRACT

Antibodies are successful biotherapeutics used for the treatment of various diseases. Throughout their therapeutic development, antibody candidates require optimization for drug developability, while retaining their functionality. This task remains a significant challenge as it is constrained by low-throughput experimental measurements. Retrieval Augmented Generation (RAG) was developed in natural language processing to generate more accurate text responses combining a retriever, a generator and a knowledge database. Here, we present a novel adaptation of this framework for the developability optimization of antibodies. Using solubility as a proof-of-concept, we demonstrate that this framework generates optimized antibody sequences with improved solubility scores, when evaluated *in silico*. This RAG framework allows precise control over the optimization process with the aim of preserving functionality of the antibody candidate. Moreover, the modular design enables adaptability across diverse optimization campaigns using a generalizable knowledge database, which has the potential to substantially reduce experimental efforts required for antibody developability optimization.

1 INTRODUCTION

Developing a safe and effective antibody therapeutic not only requires optimization for functionality, but also other properties, such as immunogenicity, manufacturability, solubility, and stability, collectively referred to as developability (Jain et al., 2017). Most developability properties are measured through low-throughput assays, which has led to substantial efforts being directed towards developing computational tools to assess developability based on amino acid (aa) sequence or protein structure (Raybould et al., 2019; Wolf Pérez et al., 2019; Khetan et al., 2022; Waight et al., 2023). However, these tools are limited when developability parameters are not directly linked to the biophysical properties of amino acids and accurate antibody structures are unavailable. Protein language models (PLMs) leveraged for protein engineering have emerged as promising tools to address these challenges. PLMs, trained on extensive protein sequence datasets, can learn biologically meaningful representations and have been applied to a variety of tasks including structure prediction, functional annotation and mutant design (Bepler & Berger, 2021; Brandes et al., 2022; Lin et al., 2023), but also to improve antibody affinity (Hie et al., 2023; Shanker et al., 2024; Jiang et al., 2024; Singh et al., 2025). Nevertheless, in a zero-shot setting PLMs are limited when discrepancies between evolutionary fitness and desired functional outcomes exist. Therefore, they may not be able to generalize and optimize diverse tasks without additional fine-tuning (Hie et al., 2023; Shanker

et al., 2024; Ding & Steinhardt, 2024). Alternative approaches have addressed the complex antibody optimization problem by combining generative modeling and Bayesian optimization (Stanton et al., 2022; Zeng et al., 2024; Amin et al., 2024). These methods optimize antibody properties in the latent space of a generative model that is trained on a large database of antibody sequences. Hence, these approaches rely on compute-intensive models that lack transparency in their generative process.

In this work, we present a novel framework based on RAG (Lewis et al., 2020) for the optimization of antibody solubility as quantified by the CamSol measure (Sormanni et al., 2015), which is a developability parameter critical in drug development. Recent studies have investigated the utility of RAG in the context of protein structure and function prediction and to enhance protein generation with diffusion models (Ma et al., 2023; Li et al., 2024; Huang et al., 2024; Shaw et al., 2024). However, to our knowledge, this is the first study that uses RAG for *in silico* antibody optimization. Originally developed in natural language processing, the RAG architecture was developed to enhance the correctness and relevance of generated text responses. It typically consists of three main components: (1) a retriever, often based on vector similarity (e.g., using embeddings from language models), (2) an external knowledge database and (3) a generator, which incorporates the retrieved information into its output response. Based on an input query, relevant documents are retrieved from the knowledge database and are used as additional context for the generator to produce a more informed and contextually accurate response. Developability optimization is a constrained optimization problem, as retaining the functionality of the antibody lead candidate, such as antigen-binding, is critical throughout the optimization process. To this aim, we repurposed RAG and developed a modular and adaptable approach that allows control over the optimization process aimed at balancing retention of functionality and optimization (Figure 1).

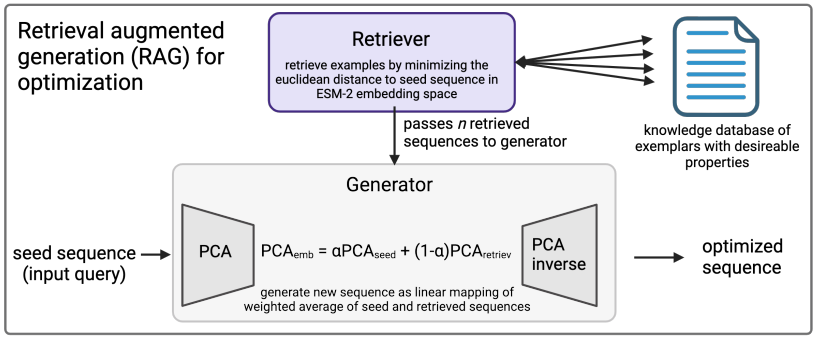

Figure 1: Visualization of a RAG framework for the generation of optimized antibody amino acid sequences.

2 Results

2.1 Overview of the RAG framework for antibody optimization

RAG for antibody optimization consists of three key components: 1) a retriever that fetches functionally similar sequences from 2) a curated knowledge database, and 3) a generator that produces optimized sequences (Figure 1). The retriever module is based on a PLM, Evolutionary scale model (ESM)-2 (Lin et al., 2023), specifically its last hidden layer representation which is a continuous embedding of the aa sequence capturing evolutionary and biologically meaningful information. For a given seed sequence, the retriever identifies n=3 sequences from a knowledge database that minimize the Euclidean distance in the ESM-2 embedding space. The knowledge database contains exemplars of antibody sequences represented as ESM-2 embeddings with desirable, optimal target solubility. Minimizing this distance is aimed at maximizing the probability of functional similarity between the seed and retrieved sequences, which are then passed to the generator. The generator is based on the principal component analysis (PCA) algorithm and linearly projects the seed and retrieved sequences into a lower-dimensional latent space of principal components (PCs) and computes a weighted average of the PCs of the seed and retrieved sequences. The hyperparameter α , also referred to as seed

ratio, weighs the contribution of the seed versus the retrieved sequences. This hyperparameter is aimed at controlling the balance between optimization and retention of functionality, enabling systematic extrapolation and optimization. The optimized PCs are then mapped back to sequence space via the PCA inverse, producing a new candidate sequence. This framework, employing the ESM-2-based retriever, is referred to as ESM-RAG. Given the computational requirements of generating PLM representations, simpler retriever modules were benchmarked against the ESM-2 retriever. Since the generator is based on PCA, we assessed a retriever module that fetches sequences based on similarity in the PCA embedding space (PCA-RAG). Moreover, a random retriever (RANDOM-RAG) was investigated to test the advantage of retrieving similar sequences rather than random sampling from the optimization set.

In order to test RAG, an exemplary antibody optimization scenario inspired by an *in vivo* (mouse immunization) antibody discovery campaign was designed. Following the identification of functional, antigen-specific lead candidates, developability parameters have to be optimized. Hence, a single-cell antibody repertoire dataset generated by Erlach et al. (2024) was used and the antigen-specific sequences of the variable heavy chains (VH) identified in this study represent the lead candidates for this optimization study (Supplementary Table 1). Solubility was selected as a proof-of-concept parameter quantified by the CamSol score as it can be directly computed from the aa sequence and has been extensively experimentally validated (Sormanni et al., 2015; Wolf Pérez et al., 2019). While it's acknowledged that optimizing a developability metric which can be computed computationally reduces the relevance of this scenario, it allowed for a systematic and comprehensive evaluation of the optimization framework without experimental validation. For evaluation, the sequence dataset was partitioned into three subsets, a training, optimization and test set based on ranked, calculated solubility scores (Supplementary Figure A.1). The training and optimization set were used to fit the PCA-based generator. The optimization set served as the knowledge database for sequence retrieval and the test set contained sequences with the highest CamSol solubility scores and was held out to evaluate the generator's ability to produce optimized, functional sequences. The optimization performance was evaluated as the improvement in solubility score, which was defined as the difference in CamSol score of the generated sequence and the seed sequence, providing a direct quantifiable metric for optimization. In addition, sequence generation was evaluated, quantified as the percentage of generated sequences that are identical to 1) the seed sequence or 2) retrieved sequences, 3) sequences in the training and optimization set, or 4) the test set. Details about the evaluation of the sequence generation of the RAG frameworks can be found in Appendix A.2. For one optimization step, a seed sequence was randomly sampled from the training dataset with solubility scores $<$ -0.1 to simulate a scenario of a lead candidate with low solubility (Supplementary Figure A.1). Subsequently, three sequences based on similarity of ESM-2 embeddings from the optimization set were retrieved to generate an optimized sequence based on the retrieved sequences and the seed sequences.

2.2 Evaluation of optimization in solubility

To quantify optimization CamSol solubility scores were computed for the optimized sequences generated across 50 optimization rounds with the three RAG frameworks, ESM-RAG, PCA-RAG, and RANDOM-RAG. Additionally, each framework was tested under varying seed ratios (α). Any generated sequence that was identical to the seed or any of the retrieved sequences was excluded from this evaluation. The solubility score improvement was defined as the difference between the optimized and seed sequence solubility scores. While all RAG frameworks demonstrated significant improvements in solubility scores, ESM-RAG achieved the greatest improvement (Supplementary Table 2, Figure 2A). Although overall differences among all frameworks were minor, these results indicate that retrieval based on similarity of ESM-2 representations is beneficial while PCA-RAG performed slightly worse than RANDOM-RAG. Overall, these results demonstrate the capability of RAG in solubility optimization. The seed ratio (α) had the largest impact on the improvement of solubility demonstrating the critical influence of this hyperparameter across all frameworks (Supplementary Figure A.5). With seed ratios $\alpha < 0.5$, nearly every optimized sequence exhibited improved solubility compared to its seed (Figure 2B, Supplementary Figure A.4A, B). These findings highlight that the seed ratio can provide precise control over this process and enables systematic optimization. Notably, the optimization framework was capable of generating optimized sequences with solubility scores up to 1.0 (Figure 2B). These scores even exceed the maximum solubility scores in the optimization set of 0.55 CamSol score (Supplementary Figure A.1) demonstrating the potential of this

framework to extrapolate beyond the optimization set, even though no data in this range of solubility scores was observed by any of the RAG frameworks.

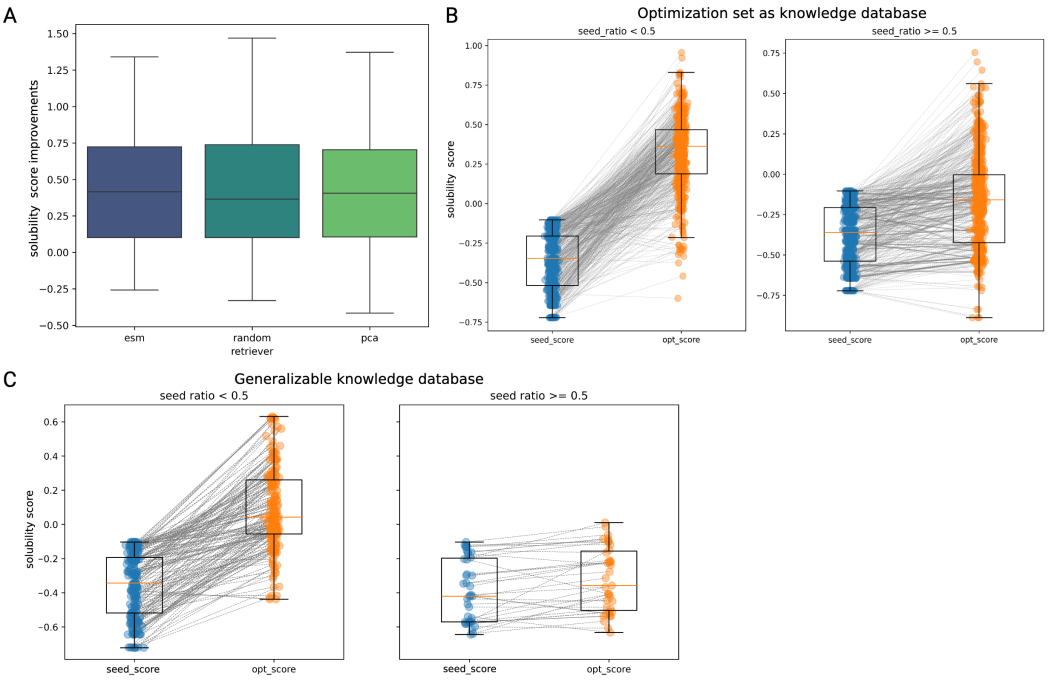

Figure 2: Evaluation of solubility optimization of antibody sequences. (A) Box plot comparing the solubility score improvements quantified as the difference between the solubility scores of the optimized and the seed sequences across 50 sampled seeds when using RAG with the optimization set as knowledge database. The solubility scores were grouped by retrievers. (B) Box plots of paired solubility scores of the seed and optimized sequences summarized across retrievers (ESM-RAG, PCA-RAG, RANDOM-RAG). (C) Evaluation of solubility scores optimized with OAS-RAG using a generalizable knowledge database. Box plots of paired solubility scores of the seed and optimized sequences are grouped by seed ratio, α, < 0.5 and ≤ 0.5. If the generated sequences were identical to the seed sequence, they were excluded for this visualization.

2.3 Optimization utilizing a generalizable knowledge database

By employing the proposed RAG optimization framework with a reusable and target-independent knowledge database that supports optimization for multiple development campaigns, this approach has the potential to reduce the overall experimental efforts for antibody developability optimization. Knowledge databases could be derived from public databases, therapeutic antibody datasets, or internally generated datasets. By covering a sufficiently large sequence space we envision this generalizable knowledge database to enable its re-usability for diverse antibody development campaigns across different targets. To evaluate RAG in such a context, we aimed to construct a generalizable database by randomly sampling 10,000 sequences from the Observed Antibody Space (OAS) database (Olsen et al., 2022) and computing their CamSol solubility scores (Sormanni et al., 2015). Based on ranked solubility scores the top 25%, 2500 sequences, with the highest solubility scores were used as the generalizable knowledge database. Seed sequences from the training dataset with low solubility ($<$0.1 CamSol score) were sampled and optimized using RAG with the ESM-2 retriever as described above, but using this generalizable database (OAS-RAG). Again, the evaluation focused on both improvements in solubility score and sequence generation. When using OAS-RAG to generate optimized sequences, a high proportion of the generated sequences was identical to the seed sequence, particularly at seed ratios (α) > 0.5 and at $\alpha = 0.9$ all sequences were identical to the seed (Supplementary Figure A.6A). Nevertheless, at seed ratio = 0.5, OAS-RAG was able to generate two sequences that are identical to sequences in the training set, indicating its capability

to produce functional antigen-binding sequences. At lower α values, most of the generated sequences were novel and none of the generated sequences were identical to the retrieved sequences. These results are in contrast to those of the other RAG frameworks using the optimization set as knowledge database, which produced fewer seed-identical sequences, but a higher proportion of retrieved sequences. When assessing the solubility score improvements, sequences identical to the seed were excluded. Sequences generated with OAS-RAG exhibited significantly higher solubility scores than the seed sequences, though the overall improvements were lower compared to the other RAG framework evaluations (ESM-RAG, PCA-RAG, RANDOM-RAG) (Supplementary Table 2. Moreover, OAS-RAG demonstrated greater sensitivity to the seed ratio, with improvements in solubility scores decreasing sharply at $\alpha \geq 0.5$, while the other RAG frameworks showed declines at $\alpha \geq 0.6$ (Supplementary Figure A.6B and A.3B). Paired solubility scores of seed and optimized sequences at seed ratios < 0.5 reveal that OAS-RAG still achieved substantial solubility improvements and even extrapolated to solubility scores as high as 0.6, exceeding solubility scores in the in the training and optimization set (Supplementary Figure A.1A, Figure 2C). However, at seed ratios $\geq$ 0.5 improvements were minor and it becomes clear that fewer optimized sequences were generated that are not identical to the seed. These findings again highlight the impact of the hyperparameter α balancing seed sequence (retention of functionality) and the retrieved sequence (optimization) in the generation of new antibody sequences.

3 DISCUSSION

In this study we present a novel computational framework for antibody developability optimization based on RAG (Lewis et al., 2020). The focus was specifically on solubility quantified by the CamSol score (Sormanni et al., 2015) which offers an efficient measure for the computational evaluation of this optimization algorithm as it can be solely computed from aa sequences. While this setup provides a simple, yet illustrative example of the framework's utility, detailed experimental validation of our findings and benchmarks against alternative approaches such as Bayesian optimization (Stanton et al., 2022; Zeng et al., 2024; Amin et al., 2024) are essential. The RAG framework we tested is based on a PLM-based retriever and a PCA-based generator and demonstrates significant improvements in solubility score of the generated antibody sequences (Figure 2, Supplementary Table 2). While using a linear PCA-based generator suggests that it may simply interpolate between sequences and their solubility scores, the RAG optimization framework could successfully generate sequences with solubility scores exceeding those in the optimization set. Moreover, optimized solubility scores reached values comparable to scores from sequences in the test set, to which none of the RAG modules had access to. These findings challenge this assumption of simplistic interpolation and highlight the framework's capability to extrapolate to some extent, which is particularly amenable for the generation of optimized antibody candidates. While the simplicity of this framework yields transparency and interpretability, it may also limit its capacity particularly with developability metrics that are more complex and require antibody structure context. The optimization of metrics such as stability, poly-specificity or immunogenicity may be more challenging and RAG in this setup might not capture enough complexity of the relationship between aa sequence and developability property. More detailed (experimental) evaluations are required to investigate how this framework performs across a broader range of developability parameters. The modular design of the framework facilitates replacement of individual components with more sophisticated models which would allow more complex sequence generation. For instance, utilizing a retriever based on explicit or implicit structural similarity could improve the framework's capacity to retrieve functionally similar antibody variants. Moreover, the PCA-based generator could be substituted with more advanced generative architectures, such as (variational) autoencoders, PLMs or diffusion models (Friedensohn et al., 2020; Shuai et al., 2023). Joint training or fine-tuning of deep learning-based retriever and generator components could further enhance sequence generation, optimization performance and adaptability.

A generalizable knowledge database that can be reused for multiple optimization campaigns offers the potential to significantly reduce overall experimental efforts. However, the feasibility of generating and curating such a database and determining its required size and diversity need further investigation to ensure broad applicability across different antibody targets. Our findings demonstrated effective solubility optimization when using a generalizable knowledge database that was sampled from the OAS database. However, without experimental validation we cannot confirm whether the generated sequences retained antigen-binding, which is crucial for real-world applicability. Sim-

ilarly, the ideal seed ratio (α) requires careful investigation for balancing the trade-off between functionality and developability optimization. Experimental validation of generated sequences will be essential to select the optimal α value that ensures that generated antibodies maintain functional relevance while reaching desired target developability.

In conclusion, our study introduces a novel approach for optimizing a developability parameter of therapeutic antibody candidates based on RAG. Despite its success for *in silico* solubility optimization, RAG's utility in real-world antibody development will depend on further refinement, testing, and adaptation to more complex metrics and tasks. However, this modular framework provides a promising foundation for incorporating more advanced models underscoring its potential for more effective and efficient antibody optimization strategies.

REFERENCES

Alan Nawzad Amin, Nate Gruver, Yilun Kuang, Lily Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. Bayesian optimization of antibodies informed by a generative model of evolving sequences. *arXiv [stat.ML]*, December 2024.

Tristan Bepler and Bonnie Berger. Learning the protein language: Evolution, structure, and function. *Cell Systems*, 12(6):654–669.e3, June 2021.

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, April 2022.

Frances Ding and Jacob Steinhardt. Protein language models are biased by unequal sequence sampling across the tree of life. *bioRxiv*, March 2024.

James Dunbar and Charlotte M Deane. ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, January 2016.

Lena Erlach, Raphael Kuhn, Andreas Agrafiotis, Danielle Shlesinger, Alexander Yermanos, and Sai T Reddy. Evaluating predictive patterns of antigen-specific B cells by single-cell transcriptome and antibody repertoire sequencing. *Cell Syst.*, 15(12):1295–1303.e5, December 2024.

Simon Friedensohn, Daniel Neumeier, Tarik A Khan, Lucia Csepregi, Cristina Parola, Arthur R Gorter de Vries, Lena Erlach, Derek M Mason, and Sai T Reddy. Convergent selection in antibody repertoires is revealed by deep learning. *bioRxiv*, pp. 2020.02.25.965673, February 2020.

Brian L Hie, Varun R Shanker, Duo Xu, Theodora U J Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nat. Biotechnol.*, pp. 1–9, April 2023.

Zhilin Huang, Ling Yang, Xiangxin Zhou, C Qin, Yijie Yu, Xiawu Zheng, Zikun Zhou, Wentao Zhang, Yu Wang, and Wenming Yang. Interaction-based retrieval-augmented diffusion models for protein-specific 3D molecule generation. *ICML*, 235:20348–20364, 2024.

Tushar Jain, Tingwan Sun, Stéphanie Durand, Amy Hall, Nga Rewa Houston, Juergen H Nett, Beth Sharkey, Beata Bobrowicz, Isabelle Caffry, Yao Yu, Yuan Cao, Heather Lynaugh, Michael Brown, Hemanta Baruah, Laura T Gray, Eric M Krauland, Yingda Xu, Maximiliano Vásquez, and K Dane Wittrup. Biophysical properties of the clinical-stage antibody landscape. *Proc. Natl. Acad. Sci. U. S. A.*, 114(5):944–949, January 2017.

Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, Samantha R Sgrizzi, Lukas Villiger, Alisan Kayabolen, B J Kim, Josephine K Carscadden, Masahiro Hiraizumi, Hiroshi Nishimasu, Jonathan S Gootenberg, and Omar O Abudayyeh. Rapid in silico directed evolution by a protein language model with EVOLVEpro. *Science*, pp. eadr6006, November 2024.

Rahul Khetan, Robin Curtis, Charlotte M Deane, Johannes Thorling Hadsund, Uddipan Kar, Konrad Krawczyk, Daisuke Kuroda, Sarah A Robinson, Pietro Sormanni, Kouhei Tsumoto, Jim Warwicker, and Andrew C R Martin. Current advances in biopharmaceutical informatics: guidelines, impact and challenges in the computational developability assessment of antibody therapeutics. *MAbs*, 14(1):2020082, January 2022.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-Tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv [cs.CL]*, May 2020.

Pan Li, Xingyi Cheng, Le Song, and Eric Xing. Retrieval AuGmented protein language models for protein structure prediction. *bioRxiv*, December 2024.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, March 2023.

Chang Ma, Haiteng Zhao, Lin Zheng, Jiayi Xin, Qintong Li, Lijun Wu, Zhihong Deng, Yang Lu, Qi Liu, and Lingpeng Kong. Retrieved sequence augmentation for protein representation learning. *arXiv [q-bio.BM]*, February 2023.

Tobias H Olsen, Fergus Boyles, and Charlotte M Deane. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Sci.*, 31(1):141–146, January 2022.

Matthew I J Raybould, Claire Marks, Konrad Krawczyk, Bruck Taddese, Jaroslaw Nowak, Alan P Lewis, Alexander Bujotzek, Jiye Shi, and Charlotte M Deane. Five computational developability guidelines for therapeutic antibody profiling. *Proc. Natl. Acad. Sci. U. S. A.*, 116(10):4025–4030, 2019.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.*, 118:e2016239118, 2021.

Varun R Shanker, Theodora U J Bruun, Brian L Hie, and Peter S Kim. Unsupervised evolution of protein and antibody complexes with a structure-informed language model. *Science*, 385(6704): 46–53, July 2024.

Peter Shaw, Bhaskar Gurram, David Belanger, Andreea Gane, Maxwell L Bileschi, Lucy J Colwell, Kristina Toutanova, and Ankur P Parikh. ProtEx: A retrieval-augmented approach for protein function prediction. *bioRxiv*, pp. 2024.05.30.596539, June 2024.

Richard W Shuai, Jeffrey A Ruffolo, and Jeffrey J Gray. IgLM: Infilling language modeling for antibody sequence design. *Cell Systems*, 14:979–989.e4, 2023.

Rohit Singh, Chiho Im, Yu Qiu, Brian Mackness, Abhinav Gupta, Taylor Joren, Samuel Sledzieski, Lena Erlach, Maria Wendt, Yves Fomekong Nanfack, Bryan Bryson, and Bonnie Berger. Learning the language of antibody hypervariability. *Proc. Natl. Acad. Sci. U. S. A.*, 122(1): e2418918121, January 2025.

Pietro Sormanni, Francesco A Aprile, and Michele Vendruscolo. The CamSol method of rational design of protein mutants with enhanced solubility. *J. Mol. Biol.*, 427(2):478–490, January 2015.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. *arXiv [cs.LG]*, March 2022.

Andrew B Waight, David Prihoda, Rojan Shrestha, Kevin Metcalf, Marc Bailly, Marco Ancona, Talal Widatalla, Zachary Rollins, Alan C Cheng, Danny A Bitton, and Laurence Fayadat-Dilman. A machine learning strategy for the identification of key in silico descriptors and prediction models for IgG monoclonal antibody developability properties. *MAbs*, 15(1):2248671, January 2023.

Adriana-Michelle Wolf Pérez, Pietro Sormanni, Jonathan Sonne Andersen, Laila Ismail Sakhnini, Ileana Rodriguez-Leon, Jais Rose Bjelke, Annette Juhl Gajhede, Leonardo De Maria, Daniel E Otzen, Michele Vendruscolo, and Nikolai Lorenzen. In vitro and in silico assessment of the developability of a designed monoclonal antibody library. *MAbs*, 11(2):388–400, January 2019.

Yimeng Zeng, Hunter Elliott, Phillip Maffettone, Peyton Greenside, Osbert Bastani, and Jacob R Gardner. Antibody design with constrained bayesian optimization. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, April 2024.

A Appendix

A.1 Method details

A.1.1 Dataset

To evaluate the optimization RAG framework, the variable regions of the heavy chain (VH) of an antibody sequence dataset from mice immunized with OVA generated by Erlach et al. (2024) was used (VH dataset). The dataset was processed and aligned as described (Erlach et al., 2024). The CamSol solubility scores of the 573 OVA-specific VH aa sequences were calculated on the web server `https://www-cohsoftware.ch.cam.ac.uk//index.php` and based on the ranked solubility scores the data was split in a training, optimization and test set. The test set consists of 10% of the sequences with the highest solubility scores and was used for evaluation of the framework and completely held out from any of the fitting or optimization. The optimization set represents the database that is used for optimization of solubility. 25% of the remaining sequence again with the largest solubility scores were assigned to the optimization set. The remaining sequences were the test set from which the seed sequences were sampled with CamSol solubility scores $<$ -0.1.

A.1.2 Retriever module

The retriever module should fetch related antibody sequences from the knowledge database, which consists of antibody examples with good solubility scores, which we refer to as optimization set. Three retriever modules were tested, ESM-2-based, PCA-based retrievers are based on Euclidean distance in the ESM-2 embedding space and PCA space. The random retrieval was used to benchmark the similarity based retrievers. To enable comparison of VH sequences of different length, the VH sequences were aligned to the same length with the ANARCI numbering system (Dunbar & Deane, 2016). Sequence gaps were represented as '-' which led to a total of 573 sequences of aligned length 139.

ESM-2 Retriever
To generate residue-based ESM-2 embeddings, the aligned sequences were passed through the ESM-2 model. ESM-2 (esm2_t33_650M_UR50D) is a variant of the ESM model with 33 layers and 650 million parameters (Rives et al., 2021), which was trained on the UR50/D 2021_04 dataset, as detailed in the ESM GitHub documentation: `https://github.com/facebookresearch/esm?tab=readme-ov-file#available`. The resulting last hidden layer representation was a vector of sequence length + 2 that includes the classification and end of sequence token added by ESM-2, which results in an embedding of dimension 141 x 1280 per sequence. The ESM-2 retriever fetches three sequences with the lowest Euclidean distance to the seed sequence's ESM-2 embedding.

PCA Retriever
The PCA retriever utilizes PCA for the generation ot the sequence embeddings. A PCA with 250 components is fitted with the one-hot encoded sequences of the training and optimization set. The sequences were transformed to a PC vector of 1x250. The number of PCs was evaluated based on the percentage of sequences that were recovered correctly based on a subsample of 300 sequences of the VH dataset, as well as explained variance of the PCs (Supplementary Figure A.2). Hence, the number of PCs was set to 250 as all 300 sequences were recovered correctly.

Similarly to the ESM-2 retriever, three sequences with the lowest Euclidean distance based on the PCA embeddings are retrieved.

Random Retriever
The random retriever serves as a baseline and randomly samples three sequences from the knowledge database.

A.1.3 GENERATOR MODULE

The linear generator module employs PCA with the number of components set to 250 and is fitted on a one-hot encoding of the sequences in the training and optimization set, without access to the test set. The generator combines the PCA embedding of the seed and the average embedding of the retrieved sequences to generate a new PCA embedding. The hyperparameter α balances the contribution of seed sequences in the new PCA embedding. Higher alpha values emphasize retaining original functionality, while lower values favor optimization, weighing the retrieved sequences more.

$$PC_{opt} = \alpha PC_{seed} + (1 - \alpha)\frac{1}{N}\sum_{i=1}^{N} PC_{retrieved,i} \tag{1}$$

, where N is a the number of retrieved sequences, which was defined to be 3 and PC_{seed}, PC_{opt} and $PC_{retrieved}$ are the PC embeddings.

The scikit-learn PCA function was used to create a continuous embedding, and PCA inverse was applied to invert the PCA embedding to a one-hot encoded sequence.

A.1.4 CREATING GENERALIZABLE KNOWLEDGE DATABASE FOR OPTIMIZATION

The generalizable knowledge database we created for solubility optimization was based on a randomly sampled subset of 10,000 unpaired VH sequences of species 'mouse_C57BL/6', which is the same mouse strain from which the training, optimization and test set was derived. The CamSol solubility score was calculated for all the 10,000 sequences and 2500 sequences with the highest solubility score were utilized as generalizable knowledge database. The sequences were aligned and processed as described above.

A.2 EVALUATION OF SEQUENCE GENERATION

To evaluate the sequence generation capabilities of the optimization framework, we analyzed the sequences that were generated from 50 randomly sampled seed sequences (Supplementary Figure A.3A). In addition, the influence of the hyperparameter α, which determines the balance between the PCs of the seed sequence and the retrieved sequences was evaluated. The RAG frameworks (ESM-RAG, PCA-RAG, RANSOM-RAG) were evaluated across a range of α values (0.1–0.9) by examining the composition of the newly generated sequences. For each of the 50 optimization runs, a seed sequence with low solubility (-0.1 CamSol score) was sampled, three sequences were retrieved, and an optimized sequence was generated. All frameworks showed the tendency to reproduce the seed sequence with higher values of the seed ratio α. At lower α values ($<$0.5), differences between the retriever frameworks were observed. All RAG frameworks could reproduce a small number of sequences from the training set, proving its capability to produce functional, antigen-specific sequences. However, given the small size of the dataset ($<$600 sequences), evaluating this framework solely on its ability to reproduce sequences within the dataset is limited. Retriever frameworks based on similarity more frequently generated sequences that are identical to the retrieved sequences, with the highest number of such sequences observed from ESM-RAG. This likely occurs because the PCs of the seed and retrieved sequences are aligned more closely, resulting in an optimized PC that is not sufficiently dissimilar to the retrieved PCs, leading to the reproduction of one of the retrieved sequences. While this behavior may not be advantageous in the current setup, it is likely less pronounced when using a generalizable database containing sequences from diverse repertoires that are overall more diverse in their antibody sequence.

A.3 SUPPLEMENTARY TABLES

Table 1: Number of sequences from used to evaluate RAG antibody optimization

Dataset	**Number of sequences**	**%**
Training set	386	65
Optimization set	130	25
Test set	57	10

Table 2: Summary of optimization evaluation

Framework	**Average improvement**	**Standard deviation**	**Significance level**
ESM-RAG	**0.422**	0.363	***
PCA-RAG	0.410	0.358	***
RANDOM-RAG	0.415	0.368	***
OAS-RAG (ESM-3 retreiver)	0.381	0.266	***

A.4 SUPPLEMENTARY FIGURES

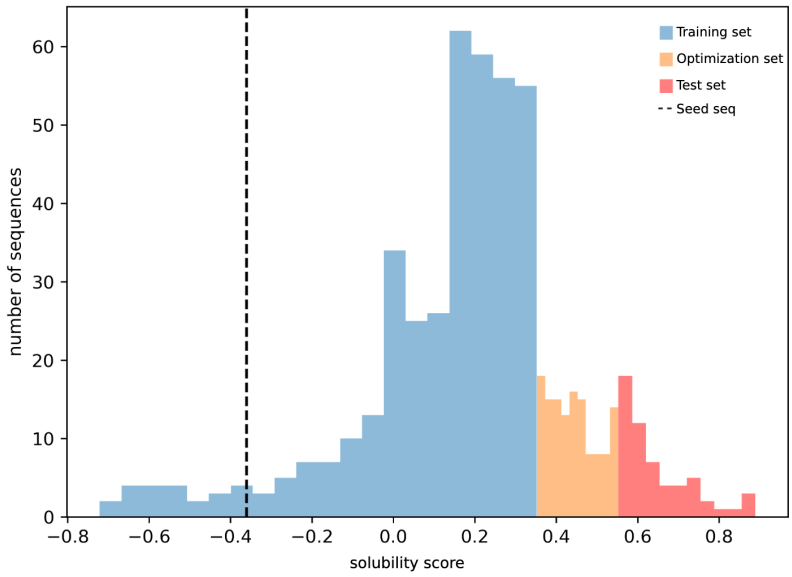

Figure A.1: Histogram of the distribution of sequences ranked by solubility values split in training, test, and optimization sets. For each optimization step a sequence with low solubility (solubility score < 0.1, dashed line in black) was randomly sampled from the training dataset and represented the seed sequence for optimization.

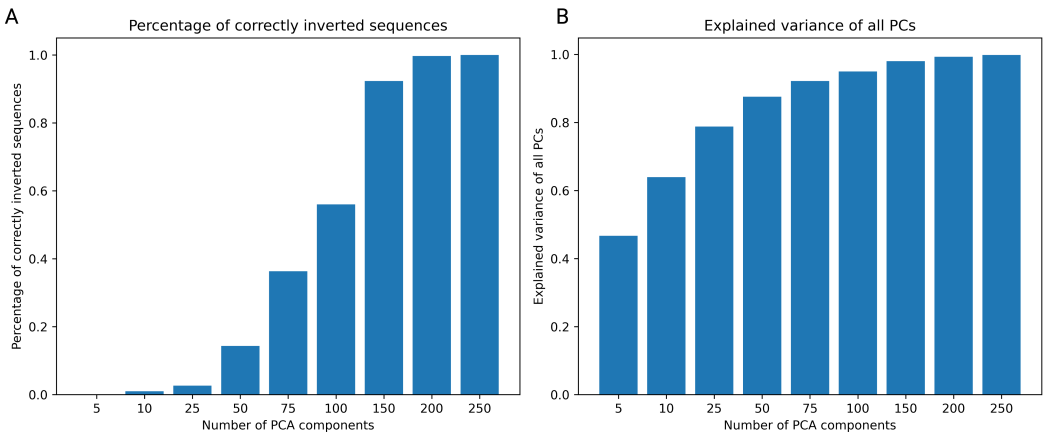

Figure A.2: Principle components evaluation. (A) Bar plot of number of PCs and the percentage of correctly inverted sequences, which were 300 sequences randomly sampled from the training set of the VH dataset. (B) Bar plot of number of PCs and the sum of explained variance with 300 sequences randomly sampled from the training set.

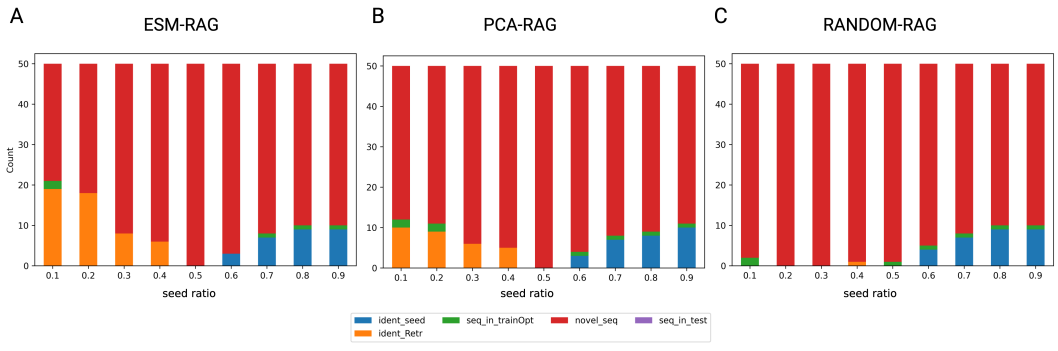

Figure A.3: Evaluation of sequence generation using different retriever modules. Bar plots of the composition of sequences generated for 50 sampled seed sequences with (A) ESM-RAG, (B) PCA-RAG and (C) random retriever. The colors indicate whether the generated sequences are identical to the seed sequence (blue), to a sequence in the training or optimization set (green), to a sequence in the held out test set (purple), identical to one of the retrieved (orange) sequences or completely novel (red).

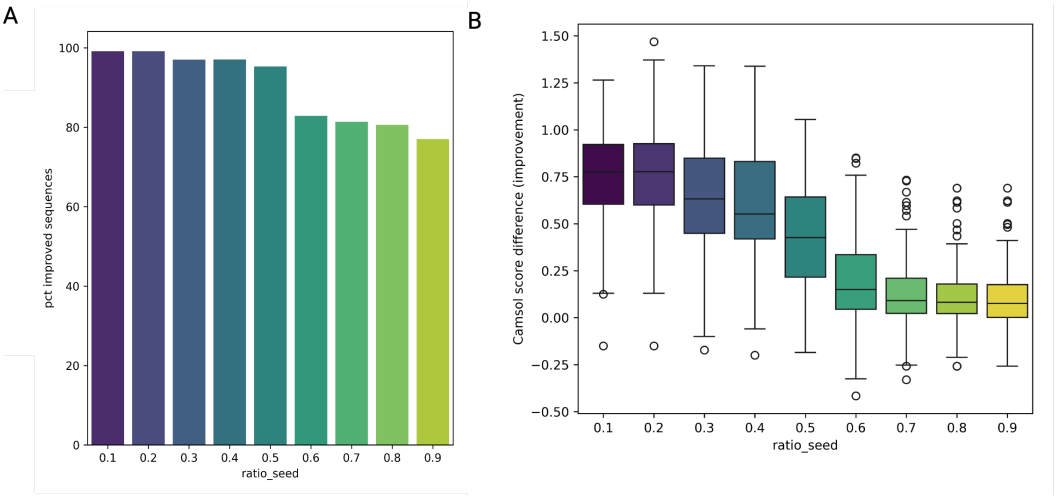

Figure A.4: Evaluation of optimized solubility scores with varying seed ratios. (A) Bar plot of the percentages of sequences for which an improvement in solubility was observed, grouped by seed ratio, α. The data was summarized across the different retrievers. (B) Box plots comparing the solubility score improvements quantified as the difference between the solubility scores of the optimized and the seed sequences across 50 sampled seeds. The solubility scores were grouped by seed ratios, α. If the generated sequences were identical to the seed sequence, they were excluded for these visualizations.

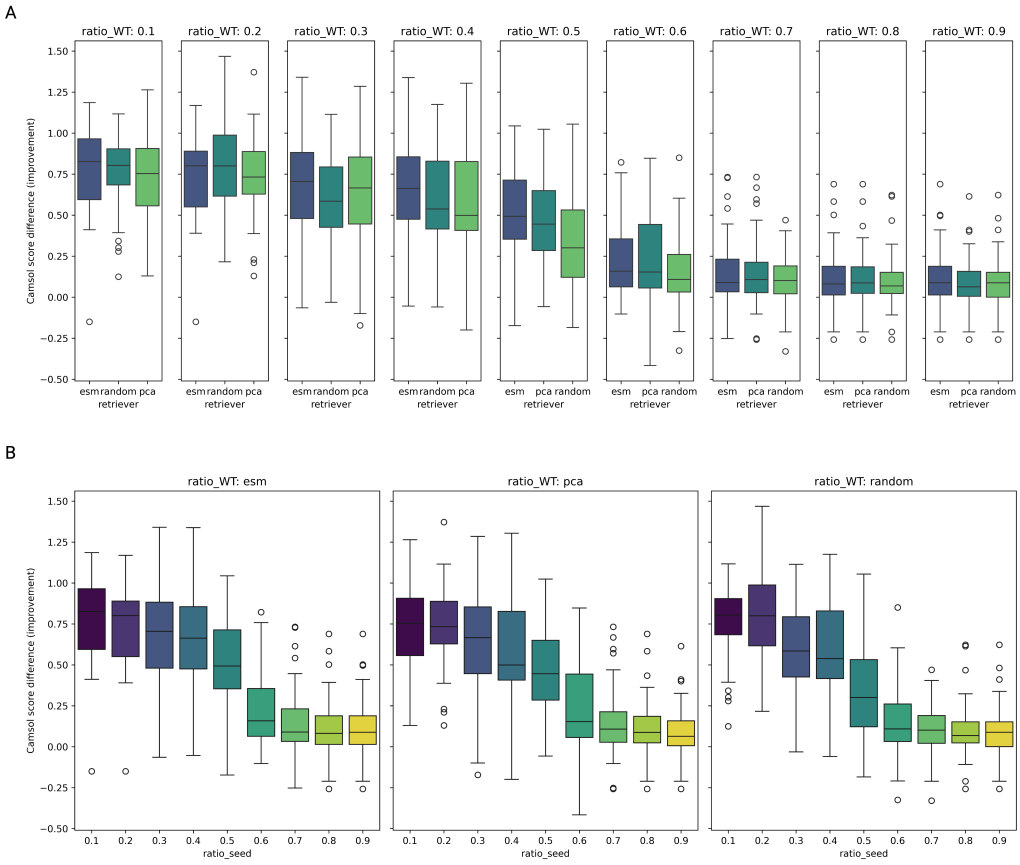

Figure A.5: Box plots of solubility score improvements as difference between score of the optimized and seed sequence. Improvements were averaged for the different retrievers and seed ratios (hyper-parameter, α). (A) Box plots are grouped by seed ratio and (B) by retriever for improved visibility of comparisons.

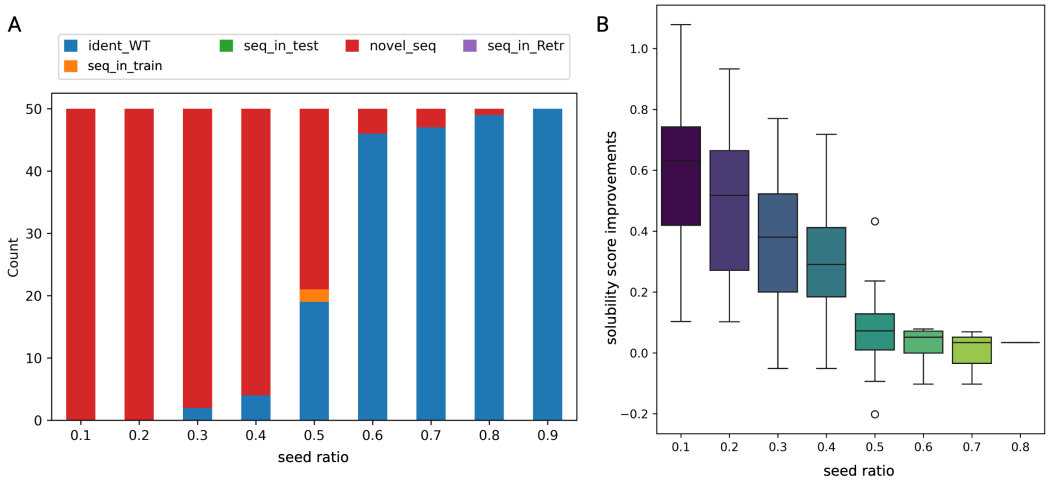

Figure A.6: Evaluation of solubility optimization and sequence generation using the generalizable knowledge database. (A) Bar plots of 50 seed optimizations with the ESM-2 retriever leveraging the generalizable knowledge database sampled from the OAS. The colors indicate whether the generated sequences are identical to the seed sequence (blue), to a sequence in the training or optimization set (green), to a sequence in the held out test set (purple), identical to one of the retrieved (orange) sequences or completely novel (red). (B) Box plot comparing the solubility score improvements quantified as the difference between the solubility scores of the optimized and the seed sequences across 50 sampled seeds. The solubility scores were grouped by seed ratios, α. Generated sequences that were identical to the seed sequence were excluded for this visualization.