DMPedia Lecture Notes in Computer Science & Engineering
Volume 1, pp. 274-282, 2026

International Conference on Multidisciplinary Perspectives in Advanced Computing and Technology (IMPACT 2026)
G. B. Pant University of Agriculture and Technology, Uttarakhand, India. Jan. 10-11, 2026

End-to-End In-Silico Antibody Humanization Pipeline for Google Colab

Lokaveenasri D, Arun Arumugaperumal
Department of Biotechnology, Rajalakshmi Engineering College, Thandalam 602105, Chennai, Tamil Nadu
dlokaveenasri@gmail.com, arun.a@rajalakshmi.edu.in

Abstract
Antibodies are Y-shaped proteins produced by the immune system to recognise and tag foreign substances for destruction. Therapeutic antibodies have become an important class of drugs for treating infectious diseases, autoimmune disorders, and cancers, but many promising candidates cannot be used directly in humans due to their immunogenicity. The immune system may recognise these non-human antibodies as foreign and attack them, rendering them unsafe for clinical use. Antibody humanization tries to solve this problem by modifying the amino acid sequence so that the antibody looks more human while still keeping its structure, binding function and basic developability properties.

In this project, I built an end-to-end, fully in-silico antibody humanization pipeline in Google Colab. The workflow starts from raw antibody FASTA sequences, applies standardized numbering to define framework and CDR regions, and then uses protein language model embeddings and a supervised classifier to estimate how "human-like" each sequence is. Guided by this humanness score, the pipeline proposes small mutations mainly in the framework regions, keeping CDRs fixed, and accepts only those changes that improve the score. Finally, ColabFold is used to check that the humanized sequence still preserves a reasonable antibody fold at the structural level. By combining AI-based sequence representations with simple, transparent design rules in a Colab notebook, this pipeline aims to make antibody humanization faster and more accessible to students and small labs, while still keeping the steps understandable and easy to modify.

Keywords: Antibody humanization, immunogenicity reduction, protein language models, sequence embeddings, humanness scoring, framework mutations, in-silico antibody design, ColabFold structure prediction.

1. Introduction

Therapeutic antibodies have become one of the main classes of modern biologic drugs, especially for treating cancers, autoimmune disorders, and some infectious diseases [2,7]. In many real projects, however, the antibody that binds well in the lab is first identified in a non-human system, such as a mouse or a rat, because it is easier to immunise those animals and screen large panels of candidates [1,4]. These non-human sequences are usually not taken "as they are" into the clinic, as the human immune system often recognises them as foreign, mounts an anti-drug antibody (ADA) response, and either neutralises the therapy or raises safety concerns [1,7]. Antibody humanization tries to solve exactly this problem by modifying the original non-human antibody so that it looks more similar to natural human antibodies, but still keeps its ability to bind the same antigen with acceptable stability and developability [1,3]. Classical humanization workflows typically rely on complementarity-determining region (CDR) grafting, where the CDR loops from the original antibody are transplanted onto a human framework (FR) scaffold [1,4]. After that, people often add back key framework residues by trial and error to restore binding, which requires several rounds of design and experimental screening [1]. This approach has worked for many approved antibodies, but it is usually slow, labour-intensive, and heavily dependent on expert structural intuition and wet-lab capacity, which are not always available to students or small labs [7,12]. In the last few years, the situation has changed because of the growth of large-scale antibody repertoire databases, standardized numbering schemes, and deep learning models trained on protein sequences [10]. These resources make it possible to define “humanness” more systematically and to design more data-driven humanization strategies that can be done fully in silico before any bench experiment [2,5].

Google Colab is a practical platform for building such workflows because it combines a Python notebook interface with free access (at least at a basic level) to GPUs and popular machine-learning libraries [8,10]. Tools for antibody numbering, sequence analysis, and even structure prediction through ColabFold can be integrated without needing a local high-performance cluster [6]. For many students, including me, Colab is also simply the easiest environment to start with, because everything runs in the browser. In this work, we describe an end-to-end in-silico antibody humanization pipeline implemented entirely in Google Colab. The pipeline takes raw FASTA sequences as input and outputs humanized candidate sequences together with both sequence-level and structure-level evaluation. The workflow combines balanced antibody training data, consistent numbering, feature engineering using protein-language model embeddings, supervised humanness classification, framework-focused mutation design, and basic structural checks via ColabFold [10,12]. The overall idea is to make a pipeline that is transparent and modular enough so that each component can be inspected, modified, or even replaced, making it useful both as a practical humanization tool and as a teaching/learning resource for people who are new to computational antibody design.

2. Pipeline Overview and Schematic

This pipeline is divided into eight major stages, each structured to maintain clarity and reproducibility. The overall workflow includes (1) data acquisition, (2) preprocessing, (3) sequence numbering, (4) feature engineering, (5) classifier training, (6) mutation design, (7) structural validation, and (8) final evaluation. Each module is implemented as an independent step in the Google Colab notebook, allowing users to substitute models, adjust parameters, or introduce additional validation stages without changing the entire workflow.

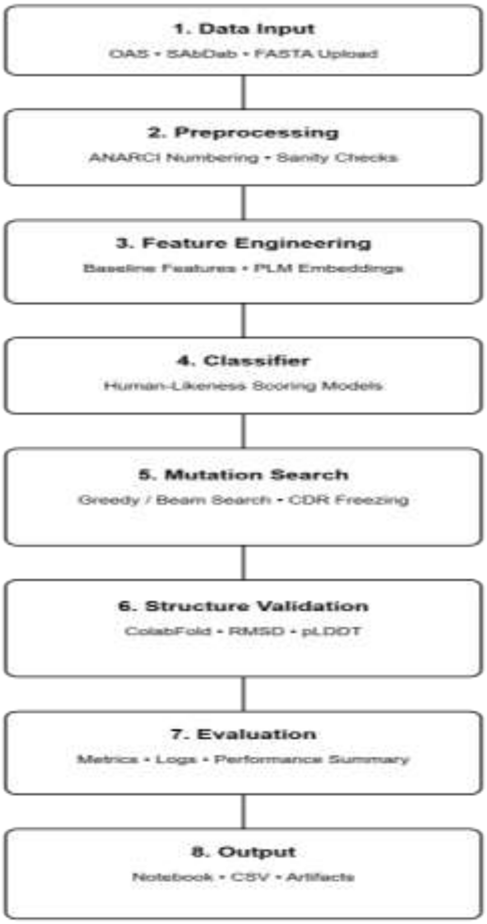

Figure 1: Eight-stage antibody humanization schematic.

3. Detailed Methods

The workflow starts by collecting antibody sequences from either public databases or user-supplied FASTA files and putting them into a standard input format. At this point, some simple quality-control checks are applied: the pipeline screens for non-standard amino acids, internal stop codons, and unusual sequence lengths that might correspond to truncated or mis-annotated entries [5,6]. Sequences that fail these checks are filtered out, and the remaining sequences are grouped by chain type (e.g., heavy vs. light), which simplifies plotting and later model training.

After quality control, each sequence is assigned consistent residue numbers using a standardized antibody numbering scheme [6]. This mapping enables the pipeline to define framework regions (FRs) and CDRs reproducibly across different antibodies, even when their raw lengths do not match exactly [6]. Working in this positional reference frame is important, because it makes it easier to separate the antigen-binding loops (CDRs) from the more conserved structural framework, which is typically the main target for humanization [5].

With the sequences numbered, the next step is feature engineering. For each antibody, the pipeline computes basic sequence-level descriptors, such as length, amino acid composition, and positional statistics, and then concatenates them with embeddings from a protein language model [2,5,8,10]. These embeddings are meant to capture higher-level patterns in antibody sequences that may correlate with whether a sequence looks “human-like” or “non-human-like” [2,5,8]. The combined feature vector serves as input to several classifiers trained to output a humanness score, i.e., the probability that a given sequence comes from a human repertoire [10,11].

Once the humanness classifier is trained on a balanced dataset of human and non-human sequences, it can be used to guide humanization. For a new non-human antibody sequence, the pipeline proposes stepwise point mutations, mainly targeting FR positions while keeping the CDRs unchanged, because CDRs are presumed to be critical for antigen binding [1,3,5]. After each proposed mutation, the updated sequence is re-scored by the humanness classifier, and only mutations that increase the predicted humanness probability are accepted. This iterative procedure continues until no further improvement is observed or a maximum of 6 framework mutations per antibody is reached, a somewhat conservative limit chosen here to avoid overmutating the antibody [1,7].

For structural validation, both the original and the humanized sequences are submitted to ColabFold for structure prediction [13]. The resulting models are compared visually and by simple metrics (such as backbone RMSD over frameworks) to check whether the overall fold is preserved after humanization. This does not replace laboratory biophysical assays, but it serves as a quick structural sanity check to spot obviously disruptive mutations early [4]. In this way, the workflow moves from raw sequences to humanized candidates, with sequence-level and structure-level information available at the end.

Table 1. Simple overview of the in-silico antibody humanization pipeline.

Step	What we do	What it gives us
1. Collect sequences	Load antibody sequences from public databases or user FASTA files and remove obvious errors.	Clean antibody sequences ready for analysis
2. Number the residues	Apply a standard antibody numbering scheme so that FR and CDR regions are clearly marked.	Numbered sequences with FR and CDR positions defined
3. Build features	Calculate basic sequence properties and protein language model embeddings for each antibody.	Feature set that describes each sequence
4. Score humanness	Use a trained classifier to predict how human-like each sequence is.	Humanness score for every antibody
5. Design mutations	Propose small point mutations mainly in framework regions; keep CDRs fixed and accept only changes that increase humanness.	Humanized sequence that is more human-like but keeps its CDRs
6. Check structure	Model original and humanized antibodies with ColabFold and compare their folds.	Structural check that the fold is preserved after humanization

4. Results

The pipeline produced a balanced and interpretable setup for in-silico antibody humanization. We started with a synthetic training dataset of 300 antibody sequences, split evenly between non-human (class 0, n = 150) and human (class 1, n = 150) examples. After adding a curated set of additional antibodies, the final training set had 400 sequences, again balanced at 200 per class. This balanced design was chosen deliberately because label imbalance can otherwise lead the classifier to be overly confident in the majority class and underperform on the minority class [2,5,10]. The class-distribution plots in the notebook confirm that both the initial synthetic dataset and the final combined dataset remain perfectly balanced.

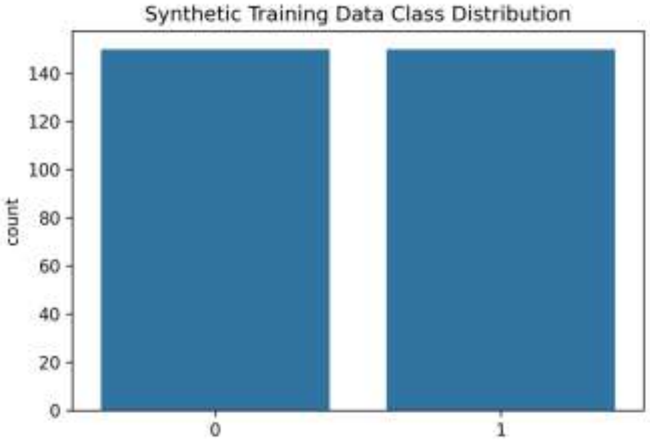

Figure 2: Synthetic training data class distribution.

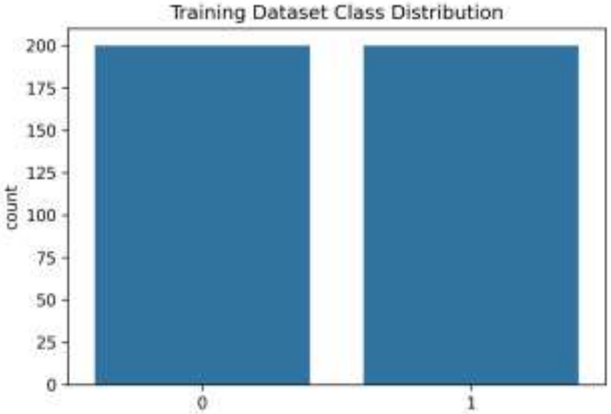

Figure 3: Final training dataset class distribution.

All sequences in the training and test sets were renumbered with a consistent antibody numbering scheme. A residue-coverage plot from positions 1–95 showed continuous indexing without gaps or duplicates, suggesting that FR and CDR regions were defined consistently across sequences [6]. This positional consistency is important for any method that wants to target mutations to framework residues while protecting CDRs.

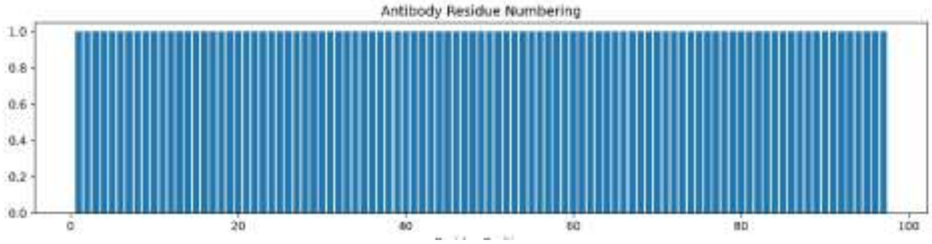

Figure 4. Antibody residue numbering coverage

Within this standardized representation, the pipeline treated CDRs as protected regions and focused mutation proposals on framework residues. This design aims to increase humanness while avoiding direct changes to the main antigen-binding loops [1,3,5]. The Colab notebook implements all components as modular steps: FASTA loading, numbering, feature extraction, classifier training, humanness scoring, mutation proposal, and re-scoring, following the schematic shown in Figure 1.

For a representative non-human antibody example, the baseline classifier first assigned an intermediate humanness score to the original sequence. The pipeline then iteratively proposed up to six framework mutations, accepting only those that increased the humanness probability as determined by the classifier. The final humanized variant reached a higher humanness score while preserving its original CDR composition. ColabFold modelling indicated that the overall fold and backbone of the humanized antibody remained similar to the original model, which suggests that the selected mutations are compatible with the native architecture and not obviously destabilising [12,13].

```
Some weights of EsmModel were not initialized from the model checkpoint at facebook/esm2_t6_8M_UR50D and are newly initialized: [
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
100%|██████████| 300/300 [00:15<00:00, 19.14it/s]

Original sequence:
 EVQLVESGGGLVQPGGSLRLSCAASGFTFSSYAMSWVRQAPGKGLEWVSAISWGGSTYYADSVKGRFTISRDNAKNTLYLQMNSLRVEDTAVYYCAR

Humanized sequence:
 EVQLVESGGGLVQPGGSLRLSCAASGFTFSSYAMSWVRQAPGFGLEWVSAISWGGSTYYADSVKGRFTISRDNAKNTLYLQMNSLRVEDTAVYYCAR

Original score: 0.515
Improved score: 0.575
Delta: 0.06
```

Figure 5. Example Antibody Humanization Outcome

5. Discussion

This work illustrates that a full in-silico antibody humanization workflow can be implemented in a reproducible and reasonably transparent way using only Google Colab and public data/resources [2,5,7,10,12]. By explicitly balancing the training data between human and non-human antibodies, the classifier avoids one of the common pitfalls in machine learning, where an imbalanced dataset makes the model appear very accurate on paper but is actually biased toward the majority class [2,5,10]. In our case, both the initial synthetic set and the enriched final dataset show equal representation of the two classes, which gives a more trustworthy basis for humanness scoring. A central strength of the pipeline is the use of consistent residue numbering before any feature extraction or mutation design is carried out [6]. The continuous coverage across positions 1–95 allows FR and CDR regions to be defined in a manner that is stable across sequences. Treating CDRs as protected regions and focusing mutations on frameworks is a pragmatic compromise that mirrors many experimental humanization strategies, where the goal is to make the antibody more human-like but still keep its binding intact [1,3–5]. In the demonstration example, this strategy allowed up to six framework mutations to be introduced while improving the humanness score and leaving the CDRs unchanged. The modular organization of the Colab notebook is another deliberate design choice. Each stage, FASTA import, numbering, feature engineering, classifier training, scoring, mutation proposal, and structural evaluation, is exposed as a separate block that users can inspect or modify. For instance, a user could plug in a different protein language model, try an alternative classifier, such as gradient-boosted trees or transformers, or implement a more advanced mutation search strategy, such as energy-based ranking [13]. Because of this modularity, the pipeline can serve as both a practical tool and a teaching example for students who want to see how sequence-based models, antibody numbering, and structure prediction can be linked in a single workflow [2].

At the same time, several limitations need to be acknowledged. First, even with balanced classes, the humanness classifier is limited by the size and diversity of the training data, as well as by any hidden biases in the underlying repertoires [11]. Synthetic sequences help to balance labels but may not capture all the features of real human and non-human antibodies [5]. Second, the structural evaluation relies on ColabFold predictions, which, while powerful, do not replace experimental measurements of stability, aggregation behaviour, or binding affinity [4]. The fact that the humanized structure looks similar to the original fold is reassuring but should be interpreted as a qualitative check, not as a guarantee of developability. Finally, the current framework-focused mutation strategy assumes that preserving CDRs is sufficient to preserve binding; however, some antibodies depend on specific framework CDR interactions, and even conservative framework mutations might affect their function [1,3,4].

Despite these caveats, the pipeline offers a concrete starting point for computational antibody humanization that is accessible to students and early-career researchers. It lowers the barrier to exploring humanness-focused design, testing different models, and connecting sequence changes to predicted structural outcomes [12]. In our experience, this kind of hands-on pipeline in Colab also makes it easier for learners to “see” how each design decision, balanced data, numbering, feature choice, model type, and mutation limit affects the final humanised sequence.

6. Conclusion

In summary, we present an end-to-end, Google Colab-based pipeline for in-silico antibody humanization that starts from raw FASTA sequences and produces humanized candidates with both sequence- and structure-level evaluation. Each input sequence is first standardized through antibody numbering, then represented by a combination of simple biological descriptors and protein language model embeddings, and finally scored by a supervised classifier that estimates its humanness [2,5,8,10,11]. Guided framework mutations are introduced stepwise, and each substitution is accepted only if it increases the predicted humanness score, while CDRs are kept fixed to protect the putative antigen-binding regions [1,7].

The use of class-balanced training data and a consistent positional scheme supports more reliable humanness predictions and helps target framework residues reproducibly. Structural evaluation with ColabFold provides an additional check that the overall antibody fold is largely preserved after humanization, offering an early indication that the mutated sequence remains compatible with the desired architecture, even if experimental testing is still required [4,7,12,13]. The Colab implementation keeps the workflow transparent: each stage, data loading, numbering, feature engineering, classifier training, scoring, mutation design, and structural evaluation, is organised as an independent, editable module.

Because of this modularity, the pipeline can be used either with default settings or as a scaffold for further methodological development. Users can integrate more expressive models, add stability or developability predictors, or refine the mutation strategy using, for example, energy-based scoring or repertoire-aware constraints [7,10,12,13]. Overall, the workflow provides a practical blueprint for computational antibody humanization that can be explored using only a web browser, making it easier for students and researchers to connect concepts in antibody engineering, machine learning, and structure prediction in a single, reproducible project.

References

[1] J. Apgar et al., “Beyond CDR-grafting: Structure-guided humanization of framework and CDR regions of an anti-myostatin antibody,” mAbs, vol. 8, no. 7, pp. 1302–1318, 2016, doi: 10.1080/19420862.2016.1215786.

[2] M. Chin, C. Marks, and C. Deane, “Humanization of antibodies using a machine learning approach on large-scale repertoire data,” Bioinformatics, vol. 37, no. 21, pp. 4041–4047, 2021, doi: 10.1093/bioinformatics/btab434.

[3] L. Chinery, J. Jeliazkov, and C. Deane, “Humatch: Fast, gene-specific joint humanisation of antibody heavy and light chains,” mAbs, vol. 16, no. 1, 2024, doi: 10.1080/19420862.2024.2434121.

[4] Y. Choi, C. Hua, C. Sentman, M. Ackerman, and C. Bailey-Kellogg, “Antibody humanization by structure-based computational protein design,” mAbs, vol. 7, no. 6, pp. 1045–1057, 2015, doi: 10.1080/19420862.2015.1076600.

[5] A. Clavero-Álvarez, T. Di Mambro, S. Perez-Gaviro, M. Magnani, and P. Bruscolini, “Humanization of antibodies using a statistical inference approach,” Sci. Rep., vol. 8, Art. no. 32986, 2018, doi: 10.1038/s41598-018-32986-y.

[6] M. Dondelinger et al., “Understanding the significance and implications of antibody numbering and antigen-binding surface/residue definition,” Front. Immunol., vol. 9, Art. no. 2278, 2018, doi: 10.3389/fimmu.2018.02278.

[7] G. Gordon, M. Raybould, A. Wong, and C. Deane, “Prospects for the computational humanization of antibodies and nanobodies,” Front. Immunol., vol. 15, Art. no. 1399438, 2024, doi: 10.3389/fimmu.2024.1399438.

[8] B. Hie et al., “Efficient evolution of human antibodies from general protein language models,” Nat. Biotechnol., vol. 42, no. 2, pp. 275–283, 2023, doi: 10.1038/s41587-023-01763-2.

[9] A. Kassardjian, D. Ivanochko, B. Barber, A. Jetha, and J. Julien, “Humanization of Pan-HLA-DR mAb 44H10 hinges on critical residues in the antibody framework,” Antibodies, vol. 13, no. 3, p. 57, 2024, doi: 10.3390/antib13030057.

[10] D. Prihoda et al., “BioPhi: A platform for antibody design, humanization, and humanness evaluation based on natural antibody repertoires and deep learning,” mAbs, vol. 14, no. 1, 2021, doi: 10.1080/19420862.2021.2020203.

[11] Gaddam, M.K., “Architecting Observability for AI-Driven Microservices at Scale.” In 2025 3rd International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI) (pp. 1830-1838). 2025, IEEE.

[12] A. Tennenhouse et al., “Computational optimization of antibody humanness and stability by systematic energy-based ranking,” Nat. Biomed. Eng., pp. 1–15, 2023, doi: 10.1038/s41551-023-01079-1.

[13] Sharma, Varinder Kumar. "Performance Evaluation of Network Slicing in 5G Core Networks." (2022).