Project 11 - AMPACT

Student Project Proposal

AMPACT: Machine Learning for Antimicrobial Peptide Activity Prediction and Profiling on YADAMP

(AntiMicrobial Peptide Activity Classification and acTivity-prediction)

Master area	AI for Drug Discovery; cheminformatics; antimicrobial peptides; machine learning
Project type	Computational project with Python implementation and data analysis
Duration	2-3 months
Supervision	One tutor or co-tutor; weekly or bi-weekly progress meetings
Recommended level	Basic Python and machine learning; basic notions of peptide chemistry are useful
Main output	A documented notebook/workflow, a technical report, and a short final presentation
Dataset	YADAMP (Yet Another Database of Antimicrobial Peptides); tutor-provided spreadsheet

Short description. The project asks the student to build a complete machine learning workflow on the YADAMP database of antimicrobial peptides (AMPs). Starting from molecular descriptors already present in the dataset, the student will explore the data, engineer and select features, and then train and evaluate models for several distinct tasks: predicting the minimum inhibitory concentration (regression), classifying peptides by activity level (binary and multi-class), predicting activity against several bacterial species at once (multi-target), grouping peptides into natural families (clustering), and flagging unusual peptides (anomaly detection). The same dataset is used throughout, so the focus is on comparing what different formulations of the problem reveal about antimicrobial peptides.

1. Background and rationale

Antimicrobial peptides are short amino acid sequences with the ability to kill or inhibit bacteria, fungi and other microorganisms. They are studied intensively as candidate alternatives to conventional antibiotics, because resistance to them tends to develop more slowly. A central practical question is which sequence and physicochemical properties make a peptide active against a given microorganism, and how strong that activity is.

The YADAMP database collects experimentally characterized antimicrobial peptides together with measured activities, expressed as minimum inhibitory concentration (MIC) values against several bacterial species, and a set of molecular descriptors such as length, molecular weight, molecular volume, net charge at different pH values, isoelectric point, hydrophobicity, instability

index and Boman index. This combination of measured activity and computed descriptors makes YADAMP a natural testbed for supervised and unsupervised machine learning.

In the context of AI for drug discovery, this matters because the same descriptors can support very different questions. One may want to estimate the exact potency of a peptide (a regression problem), simply separate strong from weak peptides (a classification problem), predict the activity profile across many species at once (a multi-target problem), discover whether peptides cluster into recognizable families (an unsupervised problem), or detect peptides whose descriptor profile is anomalous and may warrant manual inspection. Treating YADAMP through all of these lenses teaches the student how problem framing, not just the algorithm, shapes the conclusions one can draw.

2. Project objective

The objective of the project is to implement, on the YADAMP dataset, a reproducible machine learning pipeline that moves from raw descriptors to several trained and evaluated models, and to interpret what each task reveals about antimicrobial peptide activity.

The project should remain realistic for a 2-3 month student activity. It is not necessary to obtain state-of-the-art predictive performance or to introduce new experimental data. The student should focus on a clean, well-documented workflow applied consistently to the YADAMP dataset, with honest evaluation and clear discussion of limitations.

3. Specific aims

- Perform exploratory data analysis (EDA) on YADAMP, including missing-value handling, distributions, outliers and correlations among descriptors and activities.
- Engineer additional features from existing descriptors (for example per-residue normalizations and charge-related quantities) and select an informative, non-redundant feature subset.
- Train and evaluate regression models that predict the MIC value of a peptide against a chosen target species.
- Train and evaluate classification models that assign peptides to activity categories, in both a binary and a multi-class formulation.
- Build a multi-target model that predicts activity against several bacterial species simultaneously, and compare it with separate single-target models.
- Apply unsupervised methods (clustering and dimensionality reduction) to discover natural groupings of peptides, and anomaly detection to flag unusual peptides.
- Critically compare the tasks and discuss what each formulation reveals, together with the limitations of the dataset and the models.

4. Proposed methodology

4.1 Input data

The student will start from the YADAMP spreadsheet provided by the tutor. Each row is a peptide; columns include molecular descriptors (length, molecular weight, molecular volume, net charge at pH 5, 7 and 9, isoelectric point, hydrophobicity, instability index, Boman index, and similar) and measured MIC values, in micromolar units, against several bacterial species such as Staphylococcus aureus, Escherichia coli, Bacillus subtilis, Salmonella Typhimurium, Micrococcus luteus and the fungus Candida albicans. The student should first establish a clean, numeric, reproducible version of this table.

4.2 Exploratory data analysis and preprocessing

The student will inspect data types, count and handle missing values, and convert activity and descriptor columns to numeric form. Univariate analysis (histograms, boxplots, summary statistics) and bivariate analysis (scatter plots, correlation matrices) will be used to understand distributions, detect outliers (for example implausible peptide lengths) and identify redundant descriptors. Because MIC values typically span several orders of magnitude, the student should consider a log transform of the targets.

4.3 Feature engineering and selection

From the existing descriptors the student will derive additional features, for example charge density per residue, weight per residue, volume per residue, and the change of charge between pH 5 and pH 9. The student will then reduce redundancy and select an informative feature subset using a combination of correlation filtering, univariate scoring, and model-based importance (for example coefficients of a regularized linear model or feature importances of a tree-based model). The effect of feature selection on model performance should be documented.

4.4 Supervised modelling: regression

For a chosen target species, the student will predict the (log-transformed) MIC value using regression models of increasing complexity, for example linear and regularized regression, k-nearest neighbours, and a tree-based model such as random forest or gradient boosting. Models will be compared with a proper train/test split and cross-validation, reporting metrics such as RMSE, MAE and the coefficient of determination, and inspecting predicted-versus-true plots and residuals.

4.5 Supervised modelling: classification

The continuous MIC will be converted into activity categories using quantile-based thresholds. The student will first train a binary classifier (active versus less active) and then a multi-class classifier (for example highly active, moderately active, less active). Evaluation will use accuracy, precision, recall, F1-score, the confusion matrix and, for the binary case, the ROC curve and AUC. The student should comment on class balance and on the choice of decision threshold.

4.6 Multi-target prediction

Because YADAMP records activity against several species, the student will build a model that predicts activity (as regression or as classification) for several species at once, using a multi-output estimator. This will be compared with the equivalent set of independent single-species models, to assess whether modelling species jointly helps and whether activity profiles across species are correlated.

4.7 Unsupervised analysis and anomaly detection

Using the selected descriptors, the student will apply dimensionality reduction (for example PCA, and optionally t-SNE or UMAP) for visualization, and clustering (for example k-means or hierarchical clustering) to look for natural peptide families. Cluster profiles will be characterized in terms of descriptors and activity. Finally, the student will apply an anomaly detection method (for example Isolation Forest) to flag peptides with unusual descriptor profiles, and will inspect a few of them qualitatively.

5. Indicative work plan

Period	Main task	Expected output
Weeks 1-2	Data understanding and EDA	Clean numeric dataset; notebook with distributions, correlations and outlier handling.
Weeks 3-4	Feature engineering and selection	Engineered descriptors and a justified, non-redundant feature subset.
Weeks 5-6	Regression on a target species	Trained regression models with cross-validated metrics and diagnostic plots.
Weeks 7-8	Binary and multi-class classification	Trained classifiers with confusion matrices, ROC/AUC and per-class metrics.
Weeks 9-10	Multi-target, clustering and anomaly detection	Multi-output model, cluster/embedding visualizations and a list of flagged peptides.
Weeks 11-12	Integration and reporting	Final notebook, short report and presentation comparing all tasks.

6. Expected deliverables

- A documented Python notebook or small code repository implementing the full workflow on YADAMP.
- A clean, reproducible processed dataset together with the engineered and selected features.
- Trained models for regression, binary and multi-class classification, and multi-target prediction, with their evaluation metrics.
- Clustering and dimensionality-reduction visualizations and a short list of anomalous peptides with comments.
- A final technical report of approximately 3-5 pages, covering aims, methods, results, limitations and possible extensions.
- A final presentation of approximately 10-15 minutes.

7. Skills acquired by the student

- Practical handling of a real, partly incomplete chemical-biological dataset.
- Exploratory data analysis, feature engineering and feature selection for molecular descriptors.
- Building and honestly evaluating regression and classification models, including cross-validation and appropriate metrics.
- Understanding multi-target learning and the trade-off between joint and independent models.
- Use of unsupervised learning and anomaly detection to explore structure in molecular data.
- Critical interpretation of how problem framing affects the conclusions in AI-assisted drug discovery.

8. Suggested software and resources

The project can be implemented in Python using standard scientific libraries. Useful packages include pandas and NumPy for data processing, matplotlib and seaborn for visualization, scikit-learn for preprocessing, feature selection, regression, classification, multi-output models, clustering and anomaly detection, and optionally XGBoost or LightGBM for stronger tree-based models and UMAP for dimensionality reduction. The work can be carried out in Jupyter or Google Colab.

The tutor will provide the YADAMP spreadsheet and help the student decide which target species and which descriptor subset to focus on first, so that the workload stays manageable.

9. Evaluation criteria

Criterion	What should be assessed
Scientific clarity	The questions and their biological meaning are clearly defined for each task.
Technical implementation	The code is readable, documented and reproducible on the YADAMP dataset.
Methodological soundness	Train/test splitting, cross-validation, metrics and transforms are used correctly.
Comparative insight	Regression, classification, multi-target and unsupervised results are meaningfully compared.
Critical discussion	The student discusses limitations, assumptions and possible future developments.

10. Optional extensions

For students with stronger computational skills, the project may be extended by adding sequence-derived features (for example amino acid composition) when sequences are available, by trying

nested cross-validation and hyperparameter search, by interpreting models with permutation importance or SHAP values, by predicting a continuous selectivity score between two species, or by testing a simple model that uses the multi-species activity profile to group peptides by spectrum of action.