Bioinformatics

FLAMS-web 2.0: an interactive PTM resource

Alita D’hoore[1], Sieben Mertens[1], William Mwine[1], August Winderickx[1], Hannelore Longin[2], Maarten Langen[2]

[1]KU Leuven, Faculty of Bioscience Engineering, Bioinformatics, Leuven 3001, Belgium

[2]KU Leuven, Department of Microbial and Molecular Systems, Computational Systems Biology, Leuven 3001, Belgium

Abstract

Motivation: FLAMS is a Python3-based command line and web-tool that is used to compare input post-translational modification (PTM) sites with the CPLM, compendium of protein lysine modifications, and dbPTM, a database for exploring regulatory networks and functional associations of protein PTMs. The updated version of FLAMS, FLAMS-web 2.0, introduces several significant improvements, enhancing both its usability and functionality. By incorporating cross-referenced links for CPLM, UniProt, and PubMed IDs. The tool streamlines the process of connecting PTM data to external databases, reducing the manual effort required by researchers. The addition of protein visualization, which highlights PTM sites along with key features such as active and binding sites, further enriches the user experience by offering an interactive and visually informative platform. Moreover, the integration of user statistics tracking provides valuable insights into user behavior, enabling optimization of the tool to prioritize frequently queried PTMs or species. Together, these enhancements position FLAMS-web 2.0 as a more comprehensive and efficient resource for PTM analysis, broadening its appeal to researchers across various disciplines.

Implementation and availability: FLAMS-web 2.0 implementation can be found at https://gitlab.kuleuven.be/csb/aces/flams-web/-/tree/IBP24?ref_type=heads Updates to FLAMS command line tool can be found at https://gitlab.kuleuven.be/csb/aces/flams/-/tree/IBP24?ref_type=heads

1 Introduction

Post-translational modifications (PTMs) are essential biochemical changes that proteins undergo after their synthesis. These modifications can occur immediately after translation or later during the protein's lifespan, often in response to environmental stimuli or cellular needs (Zhong et al., 2023). PTMs involve the covalent addition of chemical groups to the amino acid side chains or the main backbone, enzymatic alterations, proteolytic cleavage of specific protein groups or the formation of covalent bonds between amino acids such as disulfide bridges. While some PTMs are irreversible, others are reversible, enabling proteins to adapt their functions dynamically (Ramazi & Zahiri, 2021). PTMs play a crucial role in determining protein properties and functions. They influence protein folding by altering structural dynamics, affecting processes such as protein-protein interactions, signal transduction, molecular trafficking, and localization. Among the most studied PTMs are phosphorylation, acetylation, and ubiquitination (Ramazi & Zahiri, 2021). Disruptions by PTMs can have profound consequences, often contributing to diseases. For instance, abnormal phosphorylation has been linked to neurodegenerative disorders such as Alzheimer’s disease (Zhong et al., 2023). Understanding PTMs is therefore critical to understanding the complexities of protein regulation and disease mechanisms.

PTM databases contain a lot of experimentally validated PTMs. These databases can be divided into general databases that have a wide range of PTMs and specific databases that are tailored to specific PTMs or amino acid residues. These databases were built to catalog PTMs across proteins and species. dbPTM, a general database, contains over 2.79 million PTM sites, including more than 2.24 million experimentally validated sites and 3,870 disease-associated sites. It also provides PTM substrate information from 10,200 species, enabling comparative studies across organisms (Chung et al., 2024). CPLM (Compendium of Protein Lysine Modifications) is a specialized database that contains PTMs occurring at lysine residues in proteins. CPLM 4.0 includes 592,606 modification lysine sites spanning 463,156 unique lysine residues from 105,673 proteins, representing 29 types of lysine modifications from 219 species (Zhong et al., 2023). Researchers can use these databases not only to compare their identified PTM sites as a quality check but also to functionally annotate them (Longin et al., 2024). However, this process is cumbersome, as the databases don't allow querying protein and site information together, complicating functional annotation.

FLAMS (Find Lysine Acylations and other Modification Sites), a Python3-based command line and webtool, was developed to overcome this challenge. It aids researchers in the identification of previously characterized modification sites in identical or homologous proteins across species. It achieves this by supporting position-based searches within the CPLM database and the experimentally validated subset of dbPTM (Longin et al., 2024). The FLAMS command-line tool generates a tab-separated file (TSV) containing the identified PTMs. Each PTM entry in the output includes key details such as the UniProt ID, modification type, modification site and species. The FLAMS web tool visualizes this TSV file, presenting the results in an intuitive and easily interpretable tabular format.

While FLAMS provides a framework for analyzing PTMs, there are areas where the tool can be improved to enhance usability and functionality while ensuring that its original performance remains uncompromised. One such area is the lack of crosslinked evidence links, which requires users to manually trace supporting literature. Furthermore, adding visualizations of PTMs and protein features, such as active and binding sites, could create a more informative and interactive user experience.

Logging user statistics including IP addresses, frequently queried PTMs and the most represented species, allow for further optimization of the tool and help prioritize commonly queried PTMs that lack functional annotations for targeted annotation efforts. To further improve reliability and scalability, integrating a comprehensive test suite into a continuous integration (CI) pipeline would enable automated testing of future updates, ensuring consistent functionality over time. Updating all dependencies and ensuring compatibility with a more recent Python version would also enhance usability and align the tool with modern software development practices.

2 Implementation

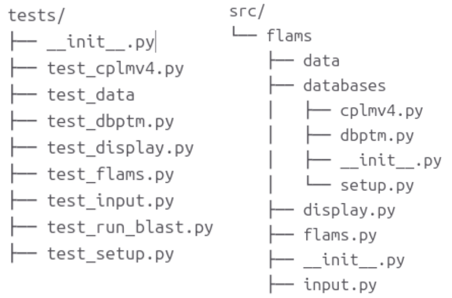
The updated version of FLAMS enhances its core functionality by introducing these new features aimed at improving usability, functionality and scalability. These updates include constructing unit tests for the code, updating dependencies, crosslinking evidence links in the output, visualization of protein structures with highlighted PTM sites and protein features like active and binding sites and tracking user statistics to gain insights into tool utilization. These improvements are seamlessly incorporated into the web tool while preserving its ease of use.

2.1 Testing

The primary objective in developing the testing suite was to verify that the update of FLAMS dependencies did not disrupt its functionality. The test suite for the FLAMS python3 command line tool was constructed using the pytest 8.3.4 framework (Pajankar, 2017), which facilitates comprehensive unit testing. The suite is organized such that each file in the source directory has a corresponding test file in the test directory, ensuring thorough testing of all functions within each file. Test cases were designed to cover user input validation as well as functionality tests to confirm that FLAMS features operate as expected. The structure of the source and test directories is shown in Figure 1.

2.2 Updating dependencies

Using pip-review 1.3.0's interactive mode (Julian & Vincent, 2022), we updated all of FLAMS' dependencies to the latest versions available as of November 11, 2024, except for the requests 2.29.0 (Kenneth, 2024) and urllib3 1.26.18 (Andrey, 2024) packages. These two dependencies were kept at the original



```
tests/                        src/
├── __init__.py               └── flams
├── test_cplmv4.py                ├── data
├── test_data                     ├── databases
├── test_dbptm.py                 │   ├── cplmv4.py
├── test_display.py               │   ├── dbptm.py
├── test_flams.py                 │   ├── __init__.py
├── test_input.py                 │   └── setup.py
├── test_run_blast.py             ├── display.py
├── test_setup.py                 ├── flams.py
                                  ├── __init__.py
                                  ├── input.py
```

Figure 1: Structure of the source and test directories

versions to maintain compatibility and prevent any disruption in the functionality of the FLAMS tool.

2.3 Crosslinking resources

The implementation of cross-linking external resources, including CPLM IDs, UniProt IDs, and PubMed evidence links, utilizes a combination of structured HTML, Jinja templating, JavaScript, and client-side API integration. The feature was developed following a modular approach, with JavaScript logic separated into dedicated files to enhance clarity, maintainability, and scalability. Hyperlinks for CPLM and UniProt IDs are embedded directly into the HTML during the server-side rendering process using Jinja templating. The Jinja template dynamically generates these links by iterating through the relevant data and inserting the corresponding identifiers into the HTML table structure. This approach enables seamless linking of UniProt IDs and CPLM IDs to their respective external databases. Similarly, CPLM IDs are embedded, with hyperlinks directing users to the CPLM database website.

The handling of PubMed IDs required a more sophisticated approach due to the nature of the "Evidence links" column in the results table. Not all evidence links in this column correspond to PubMed IDs, some correspond to experimental data, structural data and other types of evidence. This necessitates validation to differentiate PubMed entries from other types of evidence. This validation is achieved through an API call to the NCBI e-utilities service, which confirms whether a given evidence link is a valid PubMed ID. Additional complexity arises from the API's rate limit, which restricts users to three requests per second per IP address. Since some results pages may contain hundreds of evidence links, rendering such pages would be significantly delayed when individual API calls are made for each ID. To mitigate this limitation, the NCBI API's support for batched requests was utilized, allowing multiple IDs to be validated in a single API call. While this approach addresses the issue for a single client, performing these API calls on the server side introduced the potential for exceeding the rate limit when multiple client pages would be rendered simultaneously. To address this challenge, the validation process is offloaded to the client side. PubMed IDs are included in the HTML generated by the server, and a dedicated JavaScript script executes the API call on the client's side, evidently using the client's IP address. This ensures compliance with the rate limit while maintaining responsiveness for all users. Following the API response, a second dedicated JavaScript parses the returned data and updates the appropriate table cells with validated PubMed links. Each cell's ID attribute was dynamically generated during the server-side rendering process using Jinja, incorporating column and row indices to uniquely identify each cell. This approach enables JavaScript to accurately target and populate the correct table cells with clickable links based on the API response. By structuring the workflow in this manner, the table can be rendered before the API response is received, allowing users to explore other results while the evidence links are being validated in the background. Lastly, evidence links that are not present in the PubMed database are displayed as plain text.

2.4 Protein visualization

For the protein visualization the NGL viewer JavaScript library was used in combination with HTML templates, CSS templates and the Flask Python app as backend (Rose et al., 2018). The structures are obtained by querying the AlphaFold version 4 database (Abramson et al., 2024) with the UniProt ID's given by the output of the FLAMS application. The whole protein is then visualized in cartoon representation and the PTM is accentuated in red with a ball and stick representation to clearly see which one is present. A dropdown menu also gives the possibility to select the desired protein and PTM for visualization. Furthermore, key features of the protein, like the active sites and binding sites, are visualized using the ribbon representation with distinct colors compared to the whole protein. These features are extracted from the UniProt API of the protein and are all processed before the eventual output is generated. This is to guarantee smooth operability, ensuring that all necessary connections and data loading are pre-handled, so the user can inspect the output seamlessly without additional delays or processing during analysis.

2.5 Usage statistics

User activity in FLAMS is tracked by extracting information during the execution of the tool's Flask-based Python components. These components process user inputs, such as query types and parameters, and log relevant operational details. This data is extracted directly from the Flask application's handling of user interactions, this integration into the overall workflow ensures the performance is not impacted. The information is stored into an SQLite3 database which can be found in FLAMS' application support files.

SQLite3 is a serverless database engine that uses a file-based approach to store data. It is lightweight and fast as it only uses a single file to store data. Also, it implements a simple SQL syntax which makes it easy to quickly retrieve data (Sumit, 2023). The database ('stats.db') consists of three tables, each designed to track different aspects of user activity. One table ('ip_queries') stores the IP address of the user alongside the total number of queries they have executed. A second table ('modifications_amino_acids') logs the IP address, the specific modification type being queried, the amino acid of interest, and the frequency of repetitions for that exact combination. The third table ('species_modifications') captures the IP address, the species associated with the queried protein, and the number of times the user has looked up this species. By including the IP address as a common field in all tables, it becomes straightforward to perform cross-table lookups using SQL queries, enabling efficient data analysis and comprehensive tracking of user interactions. A visual representation of the database is provided in Figure 2.

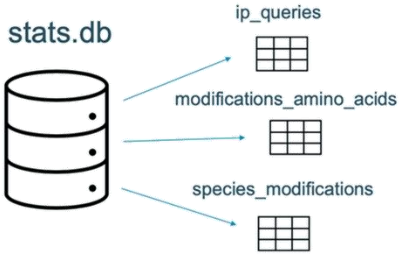
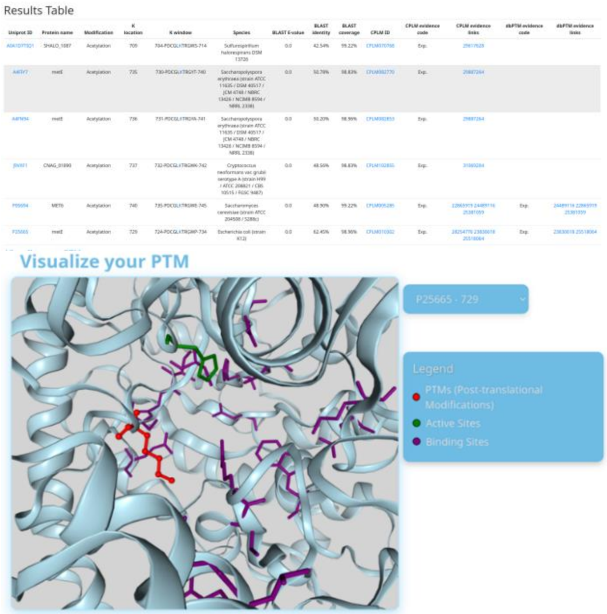


Figure 2: Visual representation of the SQLite3 database and the three tables

3 Usage

The updated version of FLAMS provides users with an enriched platform for analyzing their modification sites. In addition to assessing whether these sites have been previously reported or conserved in similar proteins. The tool now integrates enhanced visualization and cross-referenced resources.

To illustrate its usage, we examined the acetylation of MetE K738 P. aeruginosa with UniProt ID P57703. This is an acetylation in the active site and the acetylation of position 729 of P25665 in Escherichia coli which was part of output is visualized in Figure 3. The implication of this PTM will be further discussed in the discussion section of this paper.



Results Table

Uniprot ID	Protein name	Modification	K location	K window	Species	BLAST E-value	BLAST identity	BLAST coverage	CPLM ID	CPLM evidence code	CPLM evidence links	dbPTM evidence code	dbPTM evidence links
A0A1D7TSQ1	SHALO_1087	Acetylation	709	704-PDCGLKTRGWS-714	Sulfurospirillum halorespirans DSM 13726	0.0	42.54%	99.22%	CPLM070766	Exp.	25617620		
A4FIY7	metE	Acetylation	735	730-PDCGLKTRGYT-740	Saccharopolyspora erythraea (strain ATCC 11635 / DSM 40517 / JCM 4748 / NBRC 13426 / NCIMB 8594 / NRRL 2338)	0.0	50.78%	98.83%	CPLM002770	Exp.	29887264		
A4FN94	metE	Acetylation	736	731-PDCGLKTRGYA-741	Saccharopolyspora erythraea (strain ATCC 11635 / DSM 40517 / JCM 4748 / NBRC 13426 / NCIMB 8594 / NRRL 2338)	0.0	50.20%	98.96%	CPLM002853	Exp.	29887264		
J9VXF1	CNAG_01890	Acetylation	737	732-PDCGLKTRGWK-742	Cryptococcus neoformans var. grubii serotype A (strain H99 / ATCC 208821 / CBS 10515 / FGSC 9487)	0.0	48.56%	98.83%	CPLM102850	Exp.	31069284		
P05694	MET6	Acetylation	740	735-PDCGLKTRGWE-745	Saccharomyces cerevisiae (strain ATCC 204508 / S288c)	0.0	48.90%	99.22%	CPLM005285	Exp.	22865919 24489116 25381059	Exp.	24489116 22865919 25381059
P25665	metE	Acetylation	729	724-PDCGLKTRGWP-734	Escherichia coli (strain K12)	0.0	62.45%	98.96%	CPLM010302	Exp.	28254776 23830618 25518064	Exp.	23830618 25518064

Figure 3. Results page of the FLAMS output for UniProt ID P57703 with an acetylation on postion 738.

4 Discussion

The PTM in E. coli depicted in Figure 3 plays a critical role in influencing acetate metabolism, motility, and acid stress response (Castaño-Cerezo et al., 2014). Using FLAMS-web 2.0, we can visualize the 3D structure of the protein and observe that the PTM is located in close proximity to the binding site. This visualization provides insights into how the PTM might influence the protein's function by altering its interaction with substrates.

Despite these advancements, certain limitations remain. Testing of the new features, particularly the protein visualization and cross-linking of evidence links, was limited to surface-level validation, leaving some potential edge cases unexamined. While the visualizations are functional, they could benefit from additional interactivity, such as the ability to superimpose structures or toggle between different representation modes like cartoon and ball-stick models. Moreover, confidence metrics involving the AlphaFold prediction could make the visualization more transparent. Addressing these gaps through more rigorous testing and expanded visualization options would strengthen the robustness and usability of the tool.

5 Conclusion

The updated version of FLAMS introduces several significant improvements, enhancing both its usability and functionality. By incorporating cross-referenced links for CPLM, UniProt, and PubMed IDs, the tool streamlines the process of connecting PTM data to external databases, reducing the manual effort required by researchers. The addition of protein visualization, which highlights PTM sites along with key features such as active and binding sites, further enriches the user experience by offering an interactive and visually informative platform. Moreover, the integration of user statistics tracking provides valuable insights into user behavior, enabling optimization of the tool to prioritize frequently queried PTMs or species. Together, these enhancements position FLAMS as a more comprehensive and efficient resource for PTM analysis, broadening its appeal to researchers across various disciplines.

Use of AI

AI was utilized to rewrite text for improved clarity and coherence without introducing new content or altering the original meaning. Additionally, AI provided support in ensuring correct syntax and troubleshooting minor issues in the coding process.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A.,…Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, *630*(8016), 493-500. https://doi.org/10.1038/s41586-024-07487-w

Andrey, P. (2024). *urllib3 1.26.18*. Retrieved September 12 from https://pypi.org/project/urllib3/

Castaño-Cerezo, S., Bernal, V., Post, H., Fuhrer, T., Cappadona, S., Sánchez-Díaz, N. C.,…Cánovas, M. (2014). Protein acetylation affects acetate metabolism, motility and acid stress response in Escherichia coli. *Mol Syst Biol*, *10*(11), 762. https://doi.org/10.15252/msb.20145227

Chung, C. R., Tang, Y., Chiu, Y. P., Li, S., Hsieh, W. K., Yao, L.,…Lee, T. Y. (2024). dbPTM 2025 update: comprehensive integration of PTMs and proteomic data for advanced insights into cancer research. *Nucleic Acids Res*. https://doi.org/10.1093/nar/gkae1005

Julian, G., & Vincent, D. (2022). *pip-review 1.3.0*. PyPI. Retrieved November 29 from https://pypi.org/project/pip-review/

Kenneth, R. (2024). *Requests 2.29.0*. PyPI. Retrieved May 29 from https://pypi.org/project/requests/

Longin, H., Broeckaert, N., Langen, M., Hari, R., Kramarska, A., Oikarinen, K.,…van Noort, V. (2024). FLAMS: Find Lysine Acylations and other Modification Sites. *Bioinformatics*, *40*(1). https://doi.org/10.1093/bioinformatics/btae005

Mann, M., & Jensen, O. N. (2003). Proteomic analysis of post-translational modifications. *Nat Biotechnol*, *21*(3), 255-261. https://doi.org/10.1038/nbt0303-255

Pajankar, A. (2017). pytest. In *Python Unit Test automation* (2nd ed., pp. 87-100). Apress.

Ramazi, S., & Zahiri, J. (2021). Posttranslational modifications in proteins: resources, tools and prediction methods. *Database (Oxford)*, *2021*. https://doi.org/10.1093/database/baab012

Rose, A. S., Bradley, A. R., Valasatava, Y., Duarte, J. M., Prlic, A., & Rose, P. W. (2018). NGL viewer: web-based molecular graphics for large complexes. *Bioinformatics*, *34*(21), 3755-3758. https://doi.org/10.1093/bioinformatics/bty419

Sumit. (2023, April 28). *What are the advantages of SQLite database for Storing Data? hat are the advantages of SQLite database for Storing Data?* Perfect E-Learning.

Weinert, B. T., Schölz, C., Wagner, S. A., Iesmantavicius, V., Su, D., Daniel, J. A., & Choudhary, C. (2013). Lysine succinylation is a frequently occurring modification in prokaryotes and eukaryotes and extensively overlaps with acetylation. *Cell Rep*, *4*(4), 842-851. https://doi.org/10.1016/j.celrep.2013.07.024

Zhong, Q., Xiao, X., Qiu, Y., Xu, Z., Chen, C., Chong, B.,…Dai, L. (2023). Protein posttranslational modifications in health and diseases: Functions, regulatory mechanisms, and therapeutic implications. *MedComm (2020)*, *4*(3), e261. https://doi.org/10.1002/mco2.261

Jumper, J., Evans, R., Pritzel, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596, 583–589. https://doi.org/10.1038/s41586-021-03819-2

Varadi, M. et al. “AlphaFold Protein Structure Database in 2024: Providing structure coverage for over 214 million protein sequences.” Nucleic Acids Research, gkad1011 (2023). DOI: 10.1093/nar/gkad1011

Varadi, M. et al. “AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models.” Nucleic Acids Research, 50(D1), pages D439–D444 (2021). DOI: 10.1093/nar/gkab1061