

# <sup>1</sup> Automated Statistical and Machine Learning Platform for Biology Research

<sup>3</sup> **Rimmo Loyer Lego**  <sup>1</sup>, **Samantha Gauthier**  <sup>2</sup>, and **Denver Jn. Baptiste**  <sup>3</sup>

<sup>4</sup> 1 Department of Biomedical Engineering, Charles V. Schaefer, Jr. School of Engineering and Science,  
<sup>5</sup> Stevens Institute of Technology, Hoboken, NJ 07030, USA 2 Department of Computer Science, Charles  
<sup>6</sup> V. Schaefer, Jr. School of Engineering and Science, Stevens Institute of Technology, Hoboken, NJ 07030,  
<sup>7</sup> USA 3 Department of Biology, Charles V. Schaefer, Jr. School of Engineering and Science, Stevens  
<sup>8</sup> Institute of Technology, Hoboken, NJ 07030, USA

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

---

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

## <sup>9</sup> Summary

<sup>10</sup> The Automated Statistical and Machine Learning Platform for Biological Research (ASMLP-BR) software provides a platform that combines machine learning and statistical analysis for <sup>11</sup> biology research. It is deployable as both a browser-based application and a standalone desktop <sup>12</sup> software. Researchers can upload comma separated value (CSV) data files to train Random <sup>13</sup> Forest classification making use of regression models and fully automated hyperparameter <sup>14</sup> optimization. Our software performs comprehensive statistical tests through a unified interface <sup>15</sup> requiring no programming expertise. The platform integrates data preprocessing, model training <sup>16</sup> with version control, feature importance analysis, and interactive visualization, addressing the <sup>17</sup> common workflow challenge of using multiple disconnected tools. Built with React 18.3 and <sup>18</sup> TypeScript, it efficiently handles typical research datasets while allowing researchers to save and <sup>19</sup> iteratively improve models through versioned training sessions. The complete implementation <sup>20</sup> workflow from user interaction through model storage is illustrated herein.

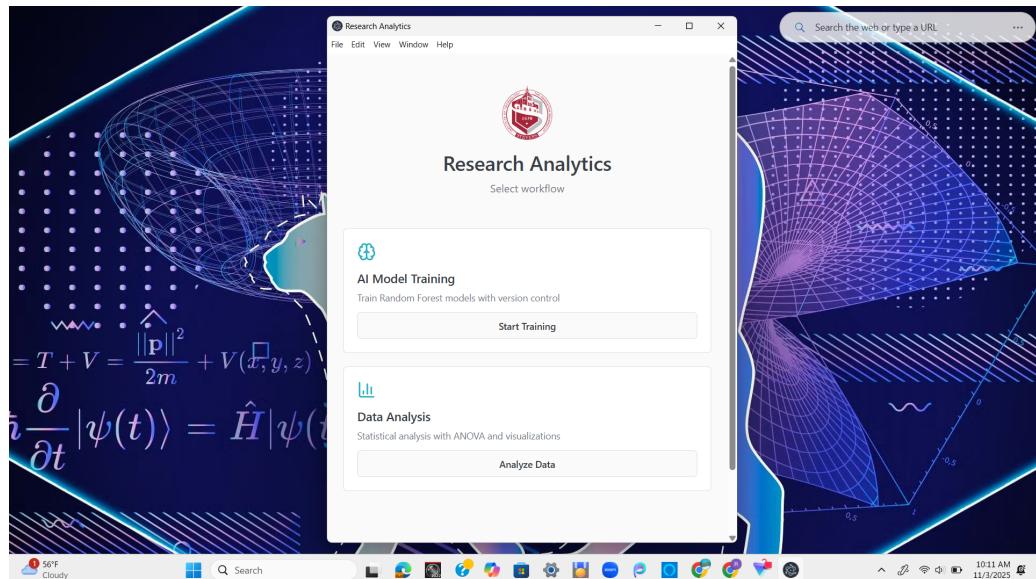
## <sup>22</sup> Statement of Need

<sup>23</sup> Biological and biomedical researchers routinely need to apply machine learning and statistics <sup>24</sup> to experimental data, but existing tools create significant barriers. Powerful frameworks <sup>25</sup> like scikit-learn ([Pedregosa et al., 2011](#)) and R ([R Core Team, 2023](#)) require programming <sup>26</sup> expertise that many experimental scientists lack. Tools operate in isolation. Researchers must <sup>27</sup> manually transfer data between separate programs for statistical testing, machine learning, and <sup>28</sup> visualization, reducing efficiency and introducing errors ([Baker, 2016](#)).

<sup>29</sup> Our software addresses analysis and computational limitations by providing both web-based and <sup>30</sup> desktop applications that combine Random Forest classification ([Breiman, 2001](#)) with standard <sup>31</sup> statistical tests (t-tests, ANOVA, correlation) in one interface. The dual deployment model <sup>32</sup> offers flexibility: researchers can use the browser version or download the standalone desktop <sup>33</sup> application for offline work and enhanced data privacy. Unlike Jupyter notebooks ([Kluyver et al., <sup>34</sup> 2016](#)), it requires no coding knowledge. The limitations seen through visual tools like Orange <sup>35</sup> ([Demšar et al., 2013](#)) are eliminated as the ASMLP-BR it includes comprehensive statistical <sup>36</sup> testing alongside machine learning. The platform enables complete workflows to upload data, <sup>37</sup> train models iteratively with version control, test hypotheses, and generate visualizations, all <sup>38</sup> without switching applications or writing code.

## <sup>39</sup> Key Features and Implementation

<sup>40</sup> The ASMLP-BR platform's modular interface organizes functionality into distinct tabs for data  
<sup>41</sup> upload, model training, prediction, result visualization, and statistical analysis (Figure 1). This  
<sup>42</sup> workflow-oriented design guides users through the complete analysis pipeline while maintaining  
<sup>43</sup> access to all features.



**Figure 1:** Interface dashboard showing the main analysis modules.

## <sup>44</sup> Architecture and Core Technologies

<sup>45</sup> The application is built with React 18.3 and TypeScript, leveraging Vite for optimized production  
<sup>46</sup> builds and Electron for desktop packaging. The implementation follows a modular component  
<sup>47</sup> architecture that separates concerns across data processing, model training, statistical analysis,  
<sup>48</sup> and visualization layers. Core dependencies include `ml-random-forest` (v2.1) for machine  
<sup>49</sup> learning algorithms, `papaparse` (v5.5) for robust CSV parsing, and `recharts` (v2.15) for  
<sup>50</sup> SVG-based interactive visualizations. All computation occurs client-side, eliminating server  
<sup>51</sup> dependencies and ensuring data privacy. The desktop application packages the same codebase  
<sup>52</sup> for Windows, macOS, and Linux platforms.

## <sup>53</sup> Data Upload and Preprocessing

<sup>54</sup> The platform supports CSV file upload through drag-and-drop or file browser interfaces. Upon  
<sup>55</sup> upload, the system performs automatic file structure detection and displays an interactive  
<sup>56</sup> preview table showing the first 100 rows. Summary statistics (mean, median, standard deviation,  
<sup>57</sup> quartiles, min/max) are computed for all numerical columns. Data validation identifies missing  
<sup>58</sup> values, offering users options for row deletion or mean/median imputation. Preprocessing  
<sup>59</sup> capabilities include z-score normalization, min-max scaling to [0,1], and automatic integer  
<sup>60</sup> encoding of categorical variables. Column type detection distinguishes between numerical,  
<sup>61</sup> categorical, and target variables, with manual override options.

## <sup>62</sup> Machine Learning Pipeline

<sup>63</sup> The platform implements Random Forest classification (Breiman, 2001), widely used for  
<sup>64</sup> chemical property prediction and QSAR modeling (Svetnik et al., 2003). Figure 2 illustrates  
<sup>65</sup> the complete implementation workflow from initial data upload through final model storage,

<sup>66</sup> showing how user interactions flow through data preprocessing, automated hyperparameter  
<sup>67</sup> optimization, model training, evaluation, and version management.

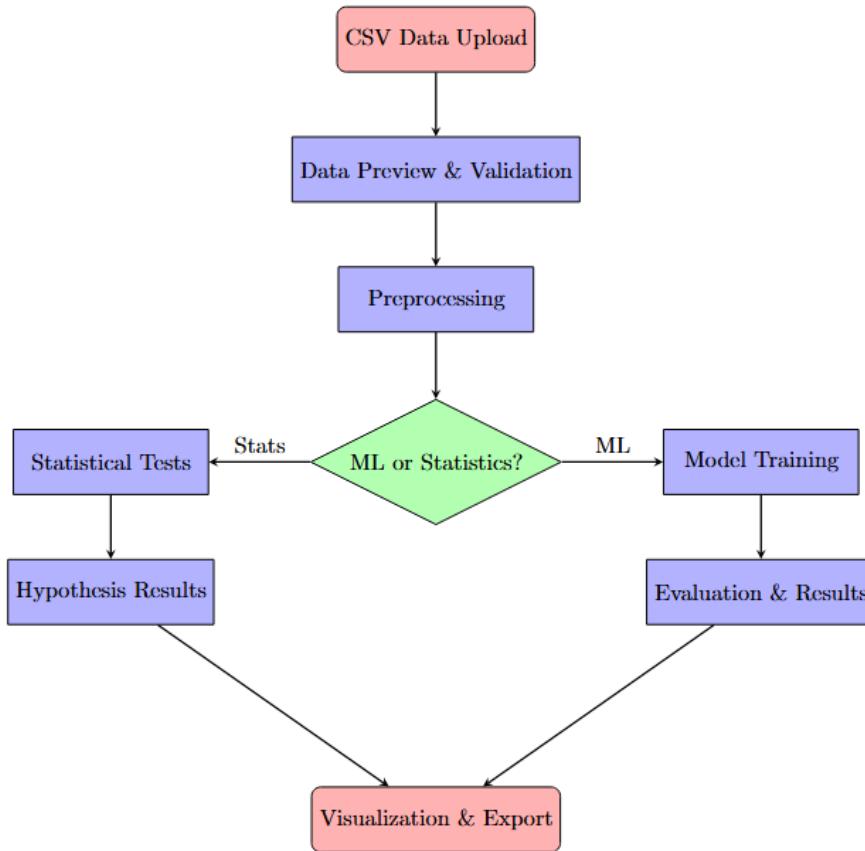


Figure 2: Implementation workflow from data upload through model storage.

<sup>68</sup> Recognizing that most researchers lack expertise in hyperparameter tuning, the system automatically  
<sup>69</sup> optimizes Random Forest parameters based on dataset characteristics. The optimization  
<sup>70</sup> algorithm adjusts the number of trees (range: 10-500), maximum tree depth, and minimum  
<sup>71</sup> samples per split according to dataset size and feature dimensionality, eliminating the need for  
<sup>72</sup> manual configuration. Training executes asynchronously with real-time progress indicators to  
<sup>73</sup> maintain interface responsiveness.

<sup>74</sup> The system performs stratified 80/20 train-test splitting to preserve class distribution, crucial for  
<sup>75</sup> imbalanced chemical datasets. Post-training, the interface displays comprehensive performance  
<sup>76</sup> metrics including accuracy, precision, recall, F1-score, and interactive confusion matrices.  
<sup>77</sup> Feature importance scores, computed via mean decrease in impurity, reveal which molecular  
<sup>78</sup> descriptors most influence classification, supporting interpretable model analysis.

<sup>79</sup> Trained models persist in browser local storage or local file system (desktop version) with  
<sup>80</sup> comprehensive version control. Researchers can save multiple model versions, each tagged  
<sup>81</sup> with training timestamp, dataset characteristics, and performance metrics. This versioning  
<sup>82</sup> system enables iterative model refinement, wherein users can load previous versions, add new  
<sup>83</sup> training data, and create improved versions while maintaining the training history. Models  
<sup>84</sup> export as JSON files for deployment, sharing, or backup purposes.

## 85 Statistical Analysis Tools

86 The platform provides both parametric and non-parametric statistical tests for hypothesis  
87 testing and exploratory analysis. For comparing group means, Welch's t-test ([Welch, 1947](#))  
88 handles unequal variances, while the Mann-Whitney U test offers a distribution-free alternative  
89 for non-normal data. One-way ANOVA enables multi-group comparisons. Correlation analysis  
90 includes Pearson's coefficient ([Pearson, 1895](#)) for linear relationships and Spearman's rank  
91 correlation for monotonic associations.

92 All statistical tests output comprehensive reports including p-values, effect sizes (Cohen's d, r),  
93 and 95% confidence intervals. The interface provides contextual guidance on assumption check-  
94 ing (normality, homoscedasticity) and appropriate test selection based on data characteristics.  
95 Visual diagnostics include Q-Q plots and residual plots for assumption validation.

## 96 Acknowledgements

97 The authors acknowledge the Department of Biomedical Engineering, Department of Biology,  
98 and Department of Computer Science at Stevens Institute of Technology for institutional  
99 support. This work was supported by computational resources provided by Stevens Institute of  
100 Technology. Last but not the least, special thanks to shadcn for the free to use opensource,  
101 userinterface templates.

## 102 References

- 103 Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S.,  
104 Irving, G., Isard, M., & others. (2016). TensorFlow: A system for large-scale machine  
105 learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and  
Implementation (OSDI)*, 265–283.
- 107 Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, 533, 452–454.  
108 <https://doi.org/10.1038/533452a>
- 109 Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal  
of Machine Learning Research*, 13, 281–305.
- 111 Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- 113 Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.  
114 <https://doi.org/10.1007/BF00994018>
- 115 Darnag, R., Minaoui, B., Glorenne, P. Y., Fakri, A., Zahrae, O., & Mourchid, M. (2010).  
116 QSAR studies of HEPT derivatives using support vector machines and neural networks.  
117 *QSAR & Combinatorial Science*, 29(5), 567–577. <https://doi.org/10.1002/qsr.200960055>
- 118 Demšar, J., Curk, T., Erjavec, A., Gorup, Č., Hočevar, T., Milutinovič, M., Možina, M.,  
119 Polajnar, M., Toplak, M., Starič, A., & others. (2013). Orange: Data mining toolbox in  
120 python. *Journal of Machine Learning Research*, 14, 2349–2353.
- 121 Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals  
of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- 123 Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning:  
Data mining, inference, and prediction* (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-84858-7>
- 126 Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K.,  
127 Hamrick, J., Grout, J., Corlay, S., & others. (2016). Jupyter notebooks—a publishing  
128 format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.),

- 129        *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90).  
130        IOS Press. <https://doi.org/10.3233/978-1-61499-649-1-87>
- 131        Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables  
132        is stochastically larger than the other. *Annals of Mathematical Statistics*, 18(1), 50–60.  
133        <https://doi.org/10.1214/aoms/1177730491>
- 134        Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press. ISBN: 978-  
135        0262018029
- 136        Pearson, K. (1895). Notes on regression and inheritance in the case of two parents. *Proceedings  
137        of the Royal Society of London*, 58, 240–242.
- 138        Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M.,  
139        Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D.,  
140        Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python.  
141        *Journal of Machine Learning Research*, 12, 2825–2830.
- 142        Perkel, J. M. (2021). Ten simple rules for writing and sharing computational analyses in jupyter  
143        notebooks. *PLOS Computational Biology*, 17(7), e1008993. <https://doi.org/10.1371/journal.pcbi.1008993>
- 145        R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation  
146        for Statistical Computing. <https://www.R-project.org/>
- 147        Spearman, C. (1904). The proof and measurement of association between two things. *American  
148        Journal of Psychology*, 15(1), 72–101. <https://doi.org/10.2307/1412159>
- 149        Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., & Feuston, B. P. (2003).  
150        Random forest: A classification and regression tool for compound classification and QSAR  
151        modeling. *Journal of Chemical Information and Computer Sciences*, 43(6), 1947–1958.  
152        <https://doi.org/10.1021/ci034160g>
- 153        Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine  
154        Learning Research*, 9, 2579–2605.
- 155        Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,  
156        Burovski, E., Peterson, P., Weckesser, W., Bright, J., & others. (2020). SciPy 1.0:  
157        Fundamental algorithms for scientific computing in python. *Nature Methods*, 17, 261–272.  
158        <https://doi.org/10.1038/s41592-019-0686-2>
- 159        Welch, B. L. (1947). The generalization of student's problem when several different population  
160        variances are involved. *Biometrika*, 34(1-2), 28–35. <https://doi.org/10.1093/biomet/34.1-2.28>