



Python vs Julia: A Comparative Study for Process Simulations

Fiammetta Caccavale, Carina L. Gargalo, Jesper Frandsen, Krist V. Gernaey, Ulrich Krühne

Process and Systems Engineering Centre (PROSYS), Department of Chemical and Biochemical Engineering, Technical University of Denmark, Søtofts Plads, Building 228A, 2800 Kgs. Lyngby, Denmark

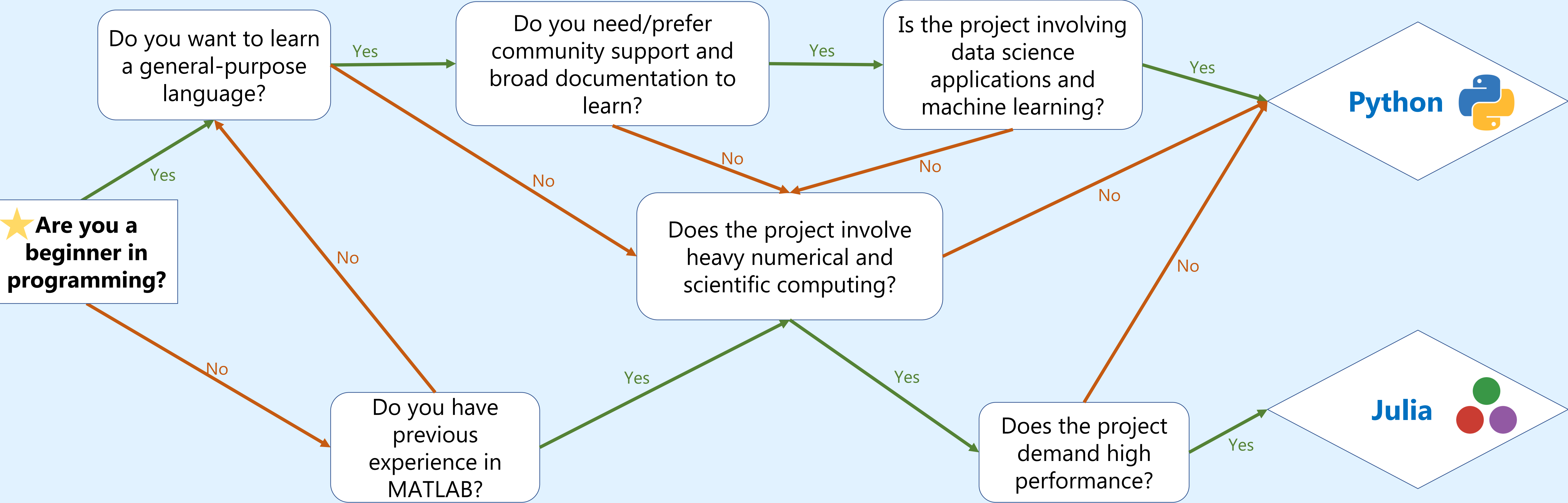


Introduction

In recent years, the chemical engineering field has been shifting from MATLAB to **Python for large-scale optimizations and data science applications**. However, a new programming language, **Julia**, is gaining terrain in the field and has the **potential to improve simulation speed and accuracy further**. This work explores the differences between the two languages and proposes a detailed overview of the best-suited applications for each language. A **GitHub repository** containing tutorials on how to get started with Julia and examples of comparable code in both Python and Julia is introduced. The integration of Process Analytical Technology (PAT) with advanced modelling and simulation is greatly enhanced by the performance capabilities of modern programming languages like Julia and Python, underscoring the critical role of language selection in optimizing analytics. This study will aid researchers and practitioners in making an **informed decision on which language fits a specific project best**.

Decision flowchart

In order to aid researchers in the choice between Python and Julia, we propose this decision flowchart that summarizes the main differences and strengths of the two languages, hopefully helping to guide the decision process.



Implementation

To compare the performance of Python and Julia, we implemented different models for fermentation in both languages.

- The study shows that the results are comparable, although the execution time in Julia is much faster.
- The models present a system of ordinary differential equations (ODE), but the divergence would be even more noticeable with systems of partial differential equations (PDE).
- A tutorial on how to get started with Julia is also provided.

Results

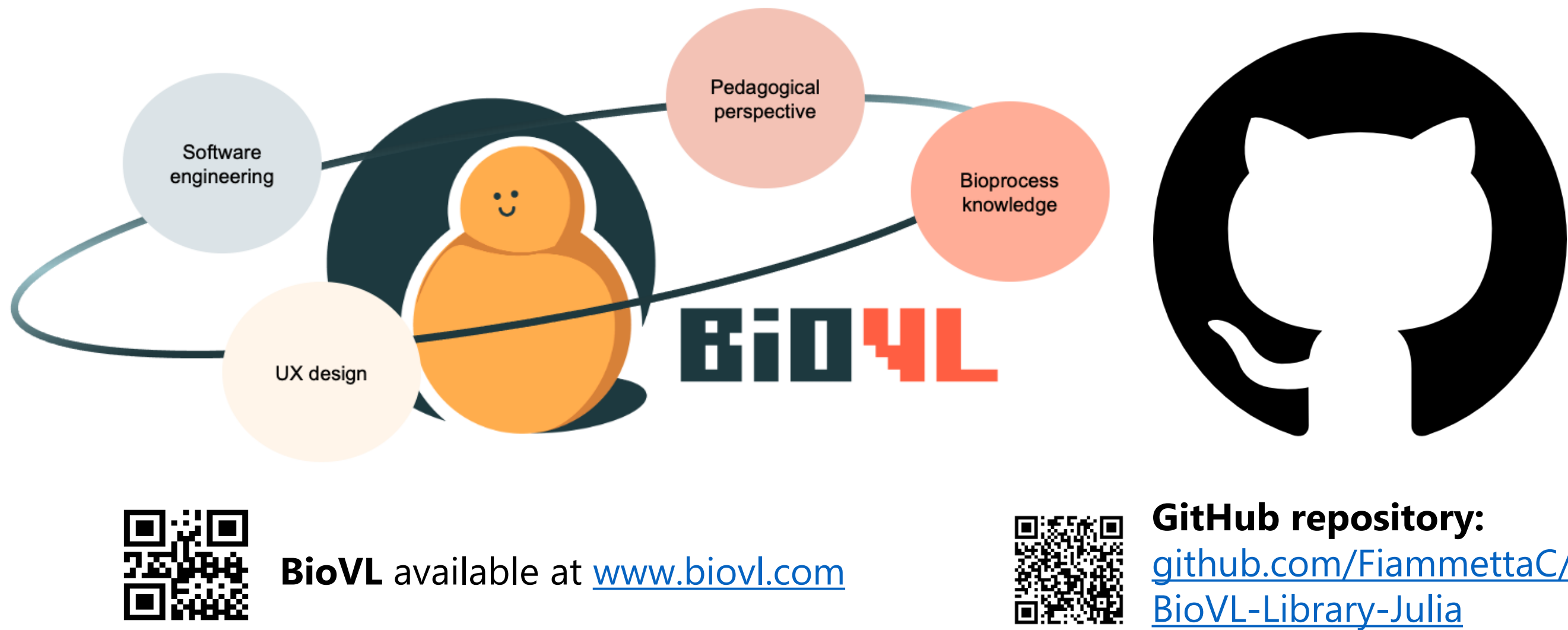
Python and Julia are both valid options in the field of chemical engineering. They do have, however, clear advantages:

- **Python advantages:** Easy-to-learn syntax, broad documentation, vast selection of machine learning models, large and active community, good libraries for visualizations, widespread and accepted.
- **Julia advantages:** fast and optimized for heavy numerical and scientific computing, similarities with MATLAB which can facilitate the transition, SciML library has very good ML models.

Model	Python		Julia	
	Runtime	memory	runtime	memory
Monod batch	0.0060 s	77226 MiB	0.0002 s	0.02 MiB
S. cerevisiae aerobic batch	0.0427 s	77897 MiB	0.0057 s	7.99 MiB
S. cerevisiae aerobic fedbatch	0.0864 s	77853 MiB	0.0063 s	10.32 MiB

Availability

BioVL is an educational software platform that aims at educating through active engagement. BioVL supports developing critical thinking by incorporating highly engaging toolboxes such as interactive chatbots and gamification.



Conclusions

We provide a decision flowchart to help researchers make an informed decision on which language is more suitable to given project requirements. We propose a comparison of models in Julia and Python for different types of fermentation at various conditions. We also introduce a tutorial on how to get started with Julia and a few tips and tricks which can speed up the process.

