

Die BioLabSim Programmier-Umgebung



Python basierte Entwicklung

- Basierend auf Python 3.9 (release '20)
- Objekt-Orientierte Programmierung
- Biologisch inspirierte Klassen verknüpft mit experiment-ähnlichen Funktionen.
- 1458 'Lines of Code'
- 824 'Lines of comments'
- Detaillierte Doku (readthedocs)

Integrated Libraries



- Biopython⁴: Kodierung von Sequenzier-Daten
- Cobrapy⁵: Simulation metabolischer Aktivität
- Exp2lpynb³: Vorhersage der Promoteraktivität

Code Development

- Git Versionskontrolle
- Open Access via RWTH GitLab
- Developer Manual
- Lokale Installation via Packet-Manager



Simulations Komponenten

Logarithmisches Wachstumsmodell
Das Wachstum wird mit der Gleichung von Verhulst berechnet:

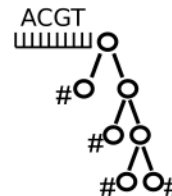
$$P(t) = \frac{K}{1 + \left(\frac{K-P_0}{P_0}\right)e^{-rt}}$$



mit 'K' als max Biomasse-Konzentration
'P₀' als initiale Biomasse und 'r' als temperatur-abhängige Wachstumsrate.

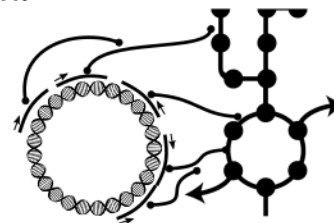
Promotor Aktivitäts Vorhersage

Ein Maschinen-Lernprogramm auf Basis eines Random-Forest sagt die Expressions-Aktivität für Sequenzen mit 40 Nukleotiden vorher. Das Programm wurde auf einer Promotor-Library von *E. coli* trainiert³.



Metabolische Fluss-Vorhersage

Ein metabolisches Modell von *E. coli* wird mit einem zufällig erzeugten Genom, inklusive Promotoren und kodierenden Sequenzen, verbunden. Änderungen der Promotor-Aktivität werden durch lineare Optimierung auf die Aktivität im Metabolismus übertragen.



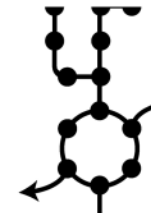
Anwendungsbeispiele

Rekombinante Expression



- Versuch #1: Messung des temperatur-spezifischen Zell-Wachstums.
- Analyse #1: Regression zur Raten-Bestimmung, max. Biomasse-Konzentration
- Versuch #2: Klonierung eines Promoters, dazu Promotor-Literaturstudie
- Analyse #2: Berechnung der Annealing Temperatur.
- Versuch #3: Integration von optimaler Temperatur, Biomasse und Promotor
- Analyse #3: Verhältnis von GC-Verhältnis auf Expressionsstärke

Metabolic Engineering (geplant)



- Versuch #1: Sequenzierung und Assembly eines Genoms
- Analyse #1: Fastq Analyse mit Greedy Algorithmus
- Versuch #2: Promoter Identifikation
- Analyse #2: Korrelation Promotor zu Sequenz
- Versuch #3: Promotor Mutationen zur Anpassung des Metabolismus
- Analyse #3: Veränderung der Flussraten durch gezielte Promotor-Anpassung

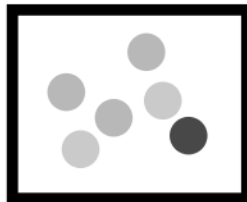
Das Lehrformat

- Teamarbeit am Computer
- Interdisziplinäre Gruppen
- Möglich in Seminarraum oder in Videokonferenz
- Einsetzbar als Übung bei MOOCs, z.B. EdX



- Cloud basierter Einsatz
- RWTHjupyter Software Service
- Weltweiter Zugriff mit RWTH Account
- Keine Installationen nötig
- Hardware unabhängig
- Sicher und zuverlässig

- Ergebnissicherung durch Interaktives "Audience Response System"
- Gemeinsame Grafik für die Vorhersagen aller Gruppen



iARS nova

Die Entwickler



Dr. Ulf Liebal
iAMB

ulf.liebal@rwth-aachen.de



Rafael Schimassek
iAMB

rafael.schimassek@rwth-aachen.de

iAMB, ABBT, RWTH Aachen:

Iris Broderius
Prof. Lars Blank



Zugang für RWTH GitLab Nutzer:
<https://git.rwth-aachen.de/ulf.liebal/biolabsim>



Exzellenzinitiative
The Fuel Science Center
PFSDS015: iStrainML

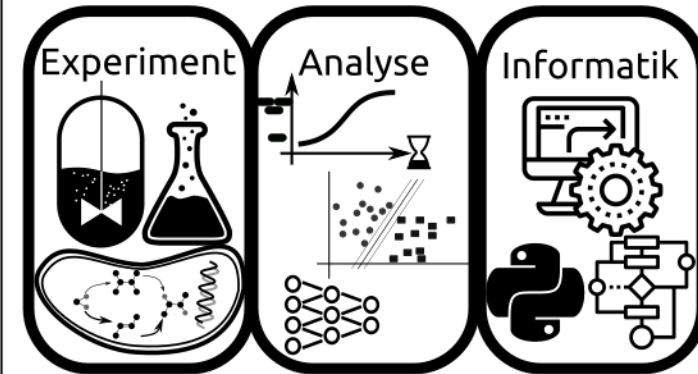


Literatur:

1. Alber *et al.*, **2019**, doi: 10.1038/s41746-019-0193-y
2. Brown & Wilson, **2018**, doi: 10.1371/journal.pcbi.1006023
3. Liebal *et al.*, **2020**, doi: 10.1101/2020.12.14.422740
4. Cock *et al.*, **2009**, doi: 10.1093/bioinformatics/btp163
5. Ebrahim *et al.*, **2013**, doi: 10.1186/1752-0509-7-74

BioLabSim

Interdisziplinäre Lehre von Informatik-Mikrobiologie



Der Fortschritt in der biologischen Forschung und der biotechnologischen Praxis beruht auf automatisierten Experimenten und computergestützten Datenanalysen¹.

Biowissenschaftliche Lehre muss Kompetenzen für Informatik, Datenanalyse und maschinellem Lernen in einem relevanten Kontext vermitteln².

BioLabSim ist ein Workflow, in dem die Voraussetzungen und Ergebnisse von mikrobiologischen Experimenten simuliert werden. Die Studierenden navigieren in einem modernen und weit verbreiteten Programmierumfeld und erlernen Datenanalysen und maschinelles Lernen an realistischen Beispielen.