



WEIHENSTEPHAN · TRIESDORF
University of Applied Sciences

Prof. Dr.-Ing. Kristina Eisen

Digitale Werkzeuge in den Ingenieurwissenschaften

Fakultät für Bioingenieurwissenschaften

Modul: Programmieren DaBiSim, MBT1/2, FK BI

Semester: Sommersemester 2025

Prüfung: Projektarbeit

- Bearbeitungszeitraum: 3 Wochen
- Beginn: 14. Mai 2025
- Abgabe: 13. Juni 2025
- Abzugebende Unterlagen:
 - Python-Programmcode as .py oder .ipynb-Datei
 - Kurzbericht (ca. 3 Seiten) inkl. aller verwendeter Quellen und Unterlagen
 - Hinweis: Unterstützung durch ChatBots ist ebenfalls als Quelle anzugeben! Zitieren Sie dazu Ihre Prompts und geben Sie das Ausführdatum an!
- Empfehlung: Dokumentation und Versionskontrolle via GitHub, GitLab
- Anmerkung:
 - Projektthemen haben Entwicklungscharakter und genügen dem geforderten Neuigkeitsanspruch
 - Finden Sie sich bei keinem Thema, stimmen Sie mit Prof. Eisen ein selbstgewähltes Thema ab!

- Projektthemen¹⁾ und Einteilung:

| Team | Matrikelnummer | Person | Titel |
|------|----------------|--------|-------|
| 1 | | | |
| 1 | | | |
| 2 | | | |
| 2 | | | |
| 3 | | | |
| 3 | | | |
| 4 | | | |
| 4 | | | |
| 5 | | | |
| 5 | | | |
| 6 | | | |
| 6 | | | |
| 7 | | | |
| 7 | | | |
| 8 | | | |
| 8 | | | |
| 9 | | | |
| 9 | | | |
| 10 | | | |
| 10 | | | |

Projektthemen Programmieren DaBiSim, MBT SoSe 2025

Projekt 1: Bildanalyse: Digitale Pathologie

Ziel: Arbeiten Sie sich in die Grundprinzipien der digitalen Pathologie ein. Fokussieren Sie sich dabei auf die methodische Mustererkennung in Mikroskopiebildern auf Basis verschiedener Algorithmen, z.B. zur Bildsegmentierung.

Hier sind Links zu Startliteratur, um sich mit dem Thema vertraut zu machen.

<https://www.nature.com/articles/s41379-021-00919-2>

<https://www.nature.com/articles/s41571-019-0252-y>

Aufgabenstellung:

Erstellen Sie ein Python-Programm, dass die Grundprinzipien der Bilderkennung zu medizinisch-diagnostische Zwecken anhand geeigneter Testdaten demonstriert.

Testdaten sind z.B. auf <https://www.kaggle.com/datasets?tags=13207-Computer+Vision> zu finden.

Stellen Sie die Ergebnisse interaktiv graphisch dar. Erstellen Sie dazu ein Dashboard, um in der Rolle des *User* verschiedene Szenarien testen zu können. Dabei beabsichtigen Sie die Genauigkeit und Robustheit der automatisierten Erkennung von Tumorgewebe in Mikroskopiebildern zu maximieren.

Projekt 2: Simulation der Enzymkinetik und Datenanalyse mit Python

Ziel:

Simulieren Sie eine Enzymkinetik mithilfe der Michaelis-Menten-Gleichung. Analysieren Sie die Daten und dokumentieren Sie die Ergebnisse. Wenden Sie grundlegende Python-Konzepte an. Setzen Sie interaktive Datenvisualisierungen ein, um die Reaktionskinetik darzustellen und die Verständlichkeit zu erhöhen.

Aufgaben:

1. Berechnen Sie eine Enzymkinetik mittels Michaelis-Menten-Gleichung.
2. Speichern Sie die Substratkonzentrationen und Reaktionsraten.
3. Visualisieren Sie die Auswirkungen der Substratkonzentration sowie weiterer Reaktionsbedingungen auf die Reaktionsgeschwindigkeit.
4. Simulieren Sie eine Reihe verschiedener Substratkonzentrationen in Szenarien und stellen Sie die Ergebnisse dar.
5. Setzen Sie einen objektorientierten Ansatz ein, um obige Aufgaben zu lösen.

Aufgabenbeschreibung:

Schreiben Sie ein Python-Programm, das die Reaktionsraten basierend auf variierenden Substratkonzentrationen mit der Michaelis-Menten-Gleichung berechnet. Sie sollten eine Reihe von Konzentrationen simulieren und die Ergebnisse speichern sowie visualisieren. Achten Sie darauf, wie sich Parameter wie V_{\max} und K_m auf das Enzymverhalten auswirken, und dokumentieren Sie Ihre Erkenntnisse. Ermöglichen Sie über Ihre Darstellung ebenfalls, wie sich unterschiedliche Reaktionsbedingungen darstellen lassen, um verschiedene Szenarien testen und Trends darstellen zu können.

Projekt 3: Analyse von DNA-Sequenzen

Ziel:

Analyse der Zusammensetzung von DNA-Sequenzen. Ziel ist die Untersuchung der Nukleotidzusammensetzung, die Extraktion von Mustern sowie die Manipulation von Sequenzen durch Implementierung von umgekehrten Ergänzungen.

Aufgaben:

1. Lesen Sie eine DNA-Sequenz ein, z.B. aus einer **FASTA**-Datei.
2. Berechnen Sie die Nukleotidhäufigkeiten.
3. Erzeugen Sie das umgekehrte Komplement einer gegebenen DNA-Sequenz
4. Visualisieren Sie Ihre Ergebnisse, z.B. mittels Balkendiagramm
5. Nutzen Sie einen objektorientierten Ansatz, um verschiedenen Berechnungsmethoden abzubilden.

Aufgabenbeschreibung:

Schreiben Sie ein Python-Programm, das eine DNA-Sequenz aus einer Datei liest, die Nukleotidvorkommen zählt und das umgekehrte Komplement der Sequenz erzeugt. Sie müssen die Sequenz mit Hilfe der Datenstrukturen von Python manipulieren und Funktionen schreiben, um diese Aufgaben zu automatisieren. Visualisieren Sie die Nukleotidhäufigkeiten in einem Balkendiagramm. Stellen Sie sicher, dass Ihr Programm mit verschiedenen DNA-Sequenzen umgehen kann, indem Sie mehrere FASTA-Dateien testen und vergleichen.

Projekt 4: Analyse von Proteinstrukturdaten

Ziel:

Analyse einer Proteinstruktur durch Extraktion von Atomkoordinaten aus einer PDB-Datei. Berechnung von Eigenschaften wie Molekulargewicht und Erstellung von 3D-Visualisierungen der Proteinstruktur.

Aufgaben:

1. Extrahieren Sie Atomkoordinaten und Reste aus einer PDB-Datei.
2. Berechnen Sie das Molekulargewicht des Proteins.
3. Erstellen Sie ein 3D-Punktdiagramm der atomaren Struktur des Proteins.
4. Nutzen Sie objektorientierte Programmierung zur Verwaltung mehrerer Proteindateien.

Aufgabenbeschreibung:

Schreiben Sie ein Python-Skript, das atomare Daten aus einer PDB-Datei extrahiert und das Molekulargewicht des Proteins berechnet. Visualisieren Sie die 3D-Struktur und gestalten Sie ein GUI. Organisieren Sie Ihren Code derart, dass flexible Dateneingabe ermöglicht wird, z.B. für mehrere Ketten in einem Protein.

Projekt 5: Analyse von Genexpressionsdaten

Ziel:

Analysieren Sie Genexpressionsdaten durch Berechnung der *Fold Changes* zwischen behandelten und Kontrollproben. Führen Sie logarithmische Transformationen ein und erstellen Sie ein Vulkandiagramm, um die Ergebnisse zu visualisieren.

Aufgaben:

1. Lesen Sie die Genexpressionsdaten ein und berechnen Sie die Fold Changes zwischen den behandelten und den Kontrollproben.
2. Wenden Sie logarithmische Transformationen auf die Daten an.
3. Visualisieren Sie die Ergebnisse (*volcano plot*).
4. Verwenden Sie objektorientierte Programmierung, um mehrere experimentelle Bedingungen zu handhaben.

Aufgabenbeschreibung:

Schreiben Sie ein Python-Skript, das die *Fold Changes* in der Genexpression zwischen zwei Bedingungen berechnet und die Ergebnisse mit Hilfe eines *volcano plots* visualisiert. Achten Sie darauf, dass Sie logarithmische Transformationen korrekt anwenden, um signifikante Unterschiede in den Expressionsniveaus hervorzuheben.

Überlegen Sie, wie die biologische Bedeutung der *Fold Changes* zu interpretieren ist und was einen signifikanten Unterschied ausmacht. Überlegen Sie, wie die Ergebnisse zu interpretieren sind.

Projekt 6: Explorative Datenanalyse

Ziel:

Explorative Datenanalyse realer biotechnologischer Daten. Dabei sollen fehlende Werten, Ausreißer, Datenverteilungen und Inkonsistenzen berücksichtigt werden, um die Datenqualität zu verbessern und eine anschließende Modellierung zu ermöglichen. Erste Trends und Muster sollen über bekannte Methoden identifiziert und dargestellt werden.

Aufgaben:

1. Laden und bereinigen Sie einen biotechnologischen Datensatz (z.B. Fermentationsdaten).
2. Zeigen Sie, wie Sie mit fehlenden Daten, Ausreißern und Inkonsistenzen umgehen.
3. Visualisieren und analysieren Sie die bereinigten Daten und vergleichen Sie diese mit den Rohdaten
4. Nutzen Sie objektorientierte Programmierung zur Automatisierung der Datenvorverarbeitung.
5. Stellen Sie alles in einem interaktiven Dashboard dar.

Aufgabenbeschreibung:

Entwickeln Sie Python-Code zur Bereinigung und Vorverarbeitung eines biotechnologischen Datensatzes, um sicherzustellen, dass dieser frei von fehlenden Werten, Ausreißern und Inkonsistenzen ist. Dokumentieren Sie die Auswirkungen Ihrer Bereinigungsmethoden auf die Qualität des Datensatzes und visualisieren Sie sowohl die Rohdaten als auch die bereinigten Daten.

Überlegen Sie, wie sich Ihre Bereinigungsmethoden auf nachgelagerte Analysen auswirken könnten, z. B. ob zu viele oder zu wenige Daten entfernt wurden. Basierend auf ersten Mustern, formulieren Sie einen Vorschlag welche Modellierungsmöglichkeiten Sie sehen oder Fragestellungen abgeleitet werden können.

Projekt 7: Enzyminhibition

Ziel:

Simulieren Sie die Inhibitionskinetik von Enzymen und analysieren Sie die Auswirkungen von kompetitiven und nicht-kompetitiver Inhibition. Nutzen Sie numerische Methoden zur Lösung von Gleichungen im Zusammenhang mit Enzym-Substrat-Wechselwirkungen.

Aufgaben:

1. Implementierung von kompetitiven und nicht-kompetitiven Hemmungsgleichungen.
2. Simulieren Sie die Reaktionsgeschwindigkeiten bei unterschiedlichen Inhibitorkonzentrationen.
3. Benutze `fsolve` von SciPy, um Gleichungen für Gleichgewichtssubstratkonzentrationen zu lösen.
4. Erstellen Sie vergleichende Diagramme, die die Wirkung von Inhibitoren auf die Enzymaktivität zeigen.

Aufgabenbeschreibung:

Entwickeln Sie ein Python-Skript, um die Hemmungskinetik von Enzymen zu simulieren und zu analysieren. Zeigen Sie wie Inhibitoren die Reaktionsgeschwindigkeit beeinflussen. Dokumentieren Sie die Unterschiede zwischen kompetitiver und nicht-kompetitiver Hemmung. Verwenden Sie die *numerischen Solver von SciPy*, um Gleichgewichtskonzentrationen zu finden und überprüfen Sie Ihre Ergebnisse mit Visualisierungen.

Stellen Sie sicher, dass Sie sowohl niedrige als auch hohe Inhibitorkonzentrationen und deren unterschiedliche Auswirkungen berücksichtigen.

Projekt 8: Modellierung von Fermentationsdaten

Ziel:

Modellieren Sie Fermentationsdaten und analysieren Sie die Auswirkungen von Variablen wie Temperatur, pH-Wert und Substratkonzentration auf die Produktbildung.

Aufgaben:

1. Simulieren Sie Fermentationsdaten mit Variablen wie Substratkonzentration, Temperatur und Produktbildung.
2. Speichern Sie die Daten und berechnen Sie Korrelationen zwischen den verschiedenen Variablen.
3. Visualisieren Sie die Korrelationen, z.B. mit einer Heatmap.
4. Verwenden Sie objektorientierte Programmierung, um verschiedene Fermentationsbedingungen betrachten und darstellen zu können.

Aufgabenbeschreibung:

Schreiben Sie ein Python-Skript, das Fermentationsdaten für verschiedene Variablen erzeugt und speichert. Analysieren Sie die Korrelation zwischen den einzelnen Variablen und visualisieren Sie die Beziehungen mithilfe von z.B. Heatmaps. Dokumentieren Sie, wie verschiedene Parameter während der Fermentation verlaufen und zusammenwirken. Erstellen Sie für alle Parameter ein interaktives Dashboard, welches das Testen verschiedener Szenarien ermöglichen soll.

Denken Sie sorgfältig über die Beziehungen zwischen den Variablen nach und darüber, wie sich Änderungen bei einer Variable auf andere auswirken können, insbesondere während des Fermentationsprozesses.

Projekt 9: Simulation einer Temperaturregelung von Bioreaktoren

Ziel:

Simulieren Sie die Temperaturregelung in einem Bioreaktor mit Hilfe von Rückkopplungsschleifen. Führen Sie Konzepte der Regelungstheorie ein und entwickeln Sie einen einfachen PID-Regler zur Temperaturregelung.

Aufgaben:

1. Schreiben Sie eine Python-Funktion zur Simulation von Temperaturänderungen in einem Bioreaktor.
2. Implementieren Sie einen PID-Regler, um eine Zieltemperatur zu halten.
3. Visualisieren Sie Temperaturänderungen über die Zeit und vergleichen Sie kontrollierte und unkontrollierte Systeme.
4. Erstellen Sie objektorientierte Klassen für den Bioreaktor und den PID-Regler.

Aufgabenbeschreibung:

Entwickeln Sie ein Python-Programm zur Simulation von Temperaturschwankungen in einem Bioreaktor. Implementieren Sie einen PID-Regler, um die Temperatur auf der Grundlage von Abweichungen von einem Zielwert anzupassen. Visualisieren Sie die Leistung des Systems und dokumentieren Sie, wie sich Änderungen der Parameter des Reglers auf die Stabilität des Systems auswirken.

Berücksichtigen Sie die Funktionsweise realer Regelsysteme und stellen Sie sicher, dass Ihr Modell deren Verhalten genau wiedergibt.

Projekt 10: Modellierung des Sauerstofftransfers bei der Fermentation

Ziel:

Modellierung der Sauerstofftransferraten in einem Bioreaktor, die für aerobe Fermentationsprozesse entscheidend sind. Einführung und Lösung von Differentialgleichungen im Zusammenhang mit der Sauerstoffaufnahme und -übertragung.

Aufgaben:

1. Schreiben Sie Python-Code zur Modellierung des Sauerstofftransfers von der Gas- zur Flüssigphase in einem Bioreaktor.
2. Implementieren Sie Differentialgleichungen, die die Sauerstoffaufnahme- und -übertragungsraten beschreiben.
3. Lösen Sie diese Gleichungen mit numerischen Methoden.
4. Visualisieren Sie die Sauerstoffkonzentration im Zeitverlauf und analysieren Sie deren Einfluss auf das Zellwachstum.

Aufgabenbeschreibung:

Schreiben Sie Python-Code, um den Sauerstofftransfer in einem Bioreaktor zu modellieren, indem Sie Differentialgleichungen verwenden, um die Gas-Flüssigkeits-Grenzfläche und die Sauerstoffaufnahme durch Zellen zu beschreiben. Lösen Sie die Gleichungen und stellen Sie dar, wie sich die Sauerstoffkonzentration im Laufe der Zeit verändert.

Dokumentieren Sie, wie die Sauerstoffverfügbarkeit das Zellwachstum und die Produktivität während der aeroben Fermentation beeinflusst.

Projekt 11: Modellierung der Glykolyse

Ziel:

Modellierung zur grundlegenden Verfolgung des Flusses von Metaboliten in einem Stoffwechselweg am Beispiel der Glykolyse

Aufgaben:

1. Schreiben Sie Python-Funktionen, um die Umwandlung von Substraten in Produkte bei der Glykolyse zu simulieren. Tipp: Starten Sie mit der stöchiometrischen Reaktionsgleichung.
2. Verfolgen Sie die Konzentration der Metaboliten über die Zeit.
3. Visualisieren Sie den Fluss der Metaboliten im Stoffwechselweg.
4. Verwenden Sie objektorientierte Programmierung.

Aufgabenbeschreibung:

Schreiben Sie Python-Code, um den Fluss von Metaboliten durch einen grundlegenden Stoffwechselweg wie die Glykolyse zu modellieren. Verfolgen Sie die Veränderungen in den Substrat- und Produktkonzentrationen im Laufe der Zeit und visualisieren Sie den Fluss durch den Stoffwechselweg. Dokumentieren Sie, wie sich Änderungen der Enzymaktivität auf den gesamten Stoffwechsel auswirken.

Berücksichtigen Sie die Auswirkungen von Stoffwechselengpässen und der Verfügbarkeit von Substraten auf den gesamten Stoffwechselweg.

Projekt 12: Adversarial image classification – Wie täusche ich eine KI?

Arbeiten Sie sich in die Grundprinzipien des *adversarial machine learnings* ein. Hier sind Quellen zur ersten Einarbeitung ins Thema:

https://en.wikipedia.org/wiki/Adversarial_machine_learning

<https://arxiv.org/abs/1412.1897>

Stellen Sie Funktionsprinzipien dar und erarbeiten Sie Chancen und Risiken. Erstellen Sie ein Python -Programm, das die Grundlagen und das Funktionsprinzip dieser (Test-)-

Verfahren zeigt. Berücksichtigen Sie dabei geeignete Testdaten sowie graphisch interaktive Ergebnisdarstellung!

Ihr Ziel ist es, die Genauigkeit und Robustheit des maschinellen Sehens für autonomes Fahren zu maximieren. Fassen Sie Ihre Ergebnisse, Chancen und Risiken sowie Erkenntnisse zusammen.

Hier finden Sie Daten und Datensätze, die öffentlich zugänglich sind und die Sie im Rahmen Ihrer Projekte verwenden können.

- NCBI GenBank: Sie können nach bestimmten Genen oder Organismen suchen und deren FASTA-Sequenzen herunterladen.
<https://www.ncbi.nlm.nih.gov/nucleotide/>
- NCBI RefSeq: Bietet kuratierte und nicht-redundante Sequenzen von Referenzgenomen. <https://www.ncbi.nlm.nih.gov/refseq/>
- Ensembl: Eine Datenbank von Genomen für eine Vielzahl von Arten, wo Sie FASTA-Dateien von genomischen, cDNA- oder Proteinsequenzen herunterladen können. <https://www.ensembl.org/info/data/ftp/index.html>
- UniProt: Bietet Proteinsequenzen und deren Annotationen im FASTA-Format.
<https://www.uniprot.org/downloads>
- www.kaggle.com
- <https://www.rcsb.org/>