

Die aktuelle Version dieses Dokuments befindet sich entweder am schwarzen Brett im Gebäude A12 vor dem Raum A12.2.09 oder im Internet auf der Seite des [Studiengangs Bioinformatik](#).

## Prof. Dr. Andreas Dominik

- **Offene Themen für Abschlussarbeiten: Deep Learning**

Wir befassen uns zunehmend mit dem Einsatz von Deep Learning in der Bioinformatik (große künstliche neuronale Netze) und entwickeln immer wieder neue Themen. Sprechen sie mich an, wenn Sie daran Interesse haben - wir finden dann sicher etwas für Sie. In den Projekten können verschiedene Programmiersprachen zum Einsatz kommen.

- **Projekt/Thesis: Krebsdiagnose mit Deep Learning**

Krebsdiagnose beruht heute hauptsächlich auf der manuellen Beurteilung (durch Ärzte) der Muster, die von den Zellen in Gewebeproben gebildet werden. Zur automatisierten Mustererkennung soll künstliches neuronales Netz (Deep Neural Network) entwickelt und auf die Diagnose des Adenokarzinoms der Prostata angewendet werden. Im Projekt können verschiedene Programmiersprachen zum Einsatz kommen, für die es Deep-Learning-Frameworks gibt (Python, Java oder c++; aber auch schönere, wie beispielsweise Lua oder Julia - wer Lust hat, mal *etwas Anderes* zu machen, kann sich hier austoben).

- **Projekt/Thesis: Datamining in Zeitreihen der Herzfrequenzvariabilität (HRV) mithilfe relativer RR-Intervalle**

HRV-Zeitreihen sollen verwendet werden, um schnell verlässliche Aussagen über den Zustand eines Patienten in der Klinik zu erhalten. Wir haben von Kooperationspartnern EKGs von Patienten und Probanden. Ihre Aufgabe wird sein, in diesen Daten nach Unterschieden zwischen gesunden und kranken Studienteilnehmern zu suchen.

*Technologien: Datamining, Machine Learning, Chaostheorie.*

- **Projekt/Thesis: Entwicklung eines Algorithmus zur Berechnung der fraktalen Dimension von Zeitreihen und mehrdimensionalen Punktmustern.**

Es soll ein Tool entwickelt und implementiert werden, mit dessen Hilfe die fraktalen Eigenschaften biologischer Muster untersucht werden können. Idealerweise entsteht ein Softwarepaket, das der Community zur Verfügung gestellt werden kann (entweder als R-Paket oder als Standaloneanwendung). Im Anwendungsteil der Arbeit kann der neue Algorithmus auf vorhandene Datensätze unserer Kooperationspartner angewendet werden (Diagnose von Herz-Kreislaufkrankheiten und Krebs).

*Technologien: Datamining, Machine Learning, Chaostheorie.*

- **Projekt/Thesis: Entwicklung und Validierung eines R-Pakets zur semi-überwachten Klassifizierung**

Ein in unserer Gruppe entwickelter und auf neuronalen Netzen basierender Klassifizierer soll als R-Paket umgesetzt und verfügbar gemacht werden. Vorhanden sind erste stand-alone-Implementierungen des Klassifizierers (c und Java) sowie eine Visualisierungssoftware für die neuronalen Netze (R). Weiterentwicklung des Algorithmus, sowie die Anwendung auf vorhandene Datensätze unserer Kooperationspartner runden das Projekt ab.

*Technologien: Datamining, künstl. neuronale Netze, Machine Learning, R.*

## Prof. Dr. Thomas Letschert

- **Projekt/Thesis: Vergleich der Ausdrucksmächtigkeit von Modelica und anderen im Bereich der Bioinformatik verwendeten Modellierungssprachen**

Theoretische Analyse der Sprache Modelica (objektorientierte deklarative Sprache, die beschreibende Gleichungen des Systems automatisch mit vorimplementiertem Solver lösen kann) im Vergleich zu state-of-the-art Modellierungssprachen wie Matlab; Literaturarbeit; Ziel: Erstellung einer Übersicht über die in der Bioinformatik und Systembiologie verwendeten Modellierungssprachen und ihrer Ausdrucksmächtigkeit

## Christopher Schölzel, M. Sc.

- **Projekt/Thesis: Implementierung eines Herzmodells in Matlab/Mathematica (Bachelor/Master)**

Schrittweises Lösen eines Modells mit 19 Gleichungen zur Untersuchung von Herzfrequenzvariabilität, Simulation und Analyse, Vergleich mit Messdaten und publizierten Ergebnissen, Implementierungen in Java und Modelica existieren bereits als Referenz.
- **Projekt/Thesis: Implementierung von biologischen Modellen mit Modelica (Bachelor/Master)**

Umsetzung eines existierenden Modells eines biologischen Systems (Herz, Blutkreislauf, Nervenzellen, Genexpression, ...) in der Sprache Modelica (objektorientierte deklarative Sprache, die beschreibende Gleichungen des Systems automatisch mit vorimplementiertem Solver lösen kann); Vergleich der Simulationsergebnisse mit dem Original
- **Thesis: Umwandlung von JSim-Modellen des Physiome-Projekts in die Sprache Modelica (Master)**

Das Physiome-Projekt (<http://physiome.org/>) sammelt verschiedenste Modelle zur menschlichen Physiologie in einer eigenen Beschreibungssprache namens JSim. Diese Sprache ist sehr ähnlich zu Modelica, lässt aber einige zentrale Eigenschaften, wie z.B. die Objektorientierung vermissen. Ihre Aufgabe wäre es herauszufinden, ob und wie JSim-Modelle in Modelica überführt werden können (anhand konkreter Beispiele oder eines vereinfachten Subsets der Sprache JSim).
- **Projekt/Thesis: Entwicklung eines Webtools für Modelica (Bachelor/Master)**

Verschiedene Arten von Tools sind möglich (Grafikeditoren, Codeeditoren, Simulationsumgebungen, Lernumgebungen, ...); Ziel ist es, das Tool im "German Network for Bioinformatics Infrastructure - de.NBI" anzubieten; Programmiersprache frei wählbar; entsprechende Desktop-Tools existieren in Java bzw. Scala
- **Projekt/Thesis: Portierung einer JavaFX-Anwendung ins Web (Bachelor/Master)**

JavaFX-Anwendungen basieren schon auf XML- und CSS-Dateien und sind daher Webanwendungen sehr nahe. Das Ausführen von Java(FX)-Anwendungen über Browserplugins wird allerdings (vermutlich zu Recht) von den meisten Browsern nicht mehr unterstützt. Ziel dieser Arbeit ist es, herauszufinden, wie man mit möglichst wenig Aufwand eine "echte" Webanwendung aus einer JavaFX-Anwendung machen kann und das erarbeitete Rezept an einer bestehenden Anwendung auszuprobieren. Idealerweise könnte so ein Tool für das "German Network for Bioinformatics Infrastructure - de.NBI" entstehen.
- **Projekt/Thesis: Entwicklung von Mini-Parsern für die statische Codeanalyse in der Lehre (Bachelor/Master)**

Mini-Parser nehmen sich nicht eine ganze Sprache zum Parsen vor, sondern nur die für ihre Aufgabe relevanten Teile. So könnte z.B. ein Parser, der nur Methodendefinitionen und Methodenaufrufe von Java parsen kann, trotzdem bereits erkennen ob eine Methode rekursiv definiert ist oder nicht. Ziel dieser Arbeit ist es, Fragestellungen der statischen Codeanalyse zu identifizieren, die für die Überprüfung studentischer Programmierabgaben interessant sind, und für diese Fragestellungen entsprechende Mini-Parser für die Sprache Java zu entwickeln.
- **Projekt/Thesis: Entwicklung einer interaktiven Simulationsumgebung für Modelica-Modelle (Bachelor/Master)**

Modelica ist eine objektorientierte deklarative Sprache, die beschreibende Gleichungen eines mathematischen Modells automatisch mit vorimplementiertem Solver lösen kann. Modelica-Modelle werden i.d.R. simuliert indem man zu Beginn der Simulation eine Parametereinstellung wählt und die Simulation dann für eine vordefinierte Anzahl von Zeitschritten rechnen lässt. Was aber, wenn man in die Simulation eingreifen möchte, um Auswirkungen von Änderungen der Parameter am "lebenden" Objekt zu testen? In dieser Arbeit soll mit Hilfe der Bibliothek JFMI eine Simulationsumgebung geschaffen werden, mit der dies möglich ist. Die Entwicklung erfolgt in Java (oder Scala), Modelica-Kenntnisse sind nicht erforderlich.

- **Projekt/Thesis: Big Data und Wearable Computing in der Medizin (Bachelor/Master)**

Projektphase: Recherche zu verschiedenen kommerziell erhältlichen tragbaren Systemen zur Bestimmung von Blutdruck bzw. Herzrate (Uhren, Armbänder, etc.); Thesis: Eines oder mehrere der ausgewählten Systeme werden mit Fachbereichsgeldern angeschafft, um eine Studie zur Qualität und Verwertbarkeit der Daten anzustellen, die von den Geräten aufgenommen werden.

**Anna Valeria Blesius, M. Sc.**

leider keine vorhanden