

Kompetenzorientiertes Studierendenfeedback



Den Lernfortschritt spürbar machen.

Amir Madany Mamlouk, Dominik Mairhöfer, Katrin Lämmermann



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
NEURO- UND BIOINFORMATIK

Lübeck, den 12. Oktober 2023



Kompetenzorientierung

Modul CS1400-KP04, CS1400

Einführung in die Bioinformatik (EinBioinfo)

Dauer:	Angebotsturnus:	Leistungspunkte:
1 Semester	Jedes Wintersemester	4

Lehrveranstaltungen:

- CS1400-V: Einführung in die Bioinformatik (Vorlesung, 2 SWS)
- CS1400-Ü: Einführung in die Bioinformatik (Übung, 1 SWS)

Workload:

- 45 Stunden Präsenzstudium
- 20 Stunden Prüfungsvorbereitung
- 55 Stunden Selbststudium

Lehrinhalte:

- Leben, Evolution & das Genom
- Sequence Assembly - Maschinelles Auslesen von genetischer Information
- DNA Sequenzmodelle & Hidden Markov Ketten
- Viterbi-Algorithmus
- Sequence Alignment & Dynamische Programmierung
- Unüberwachte Datenanalyse (k-means, PCA, ICA)
- DNA Microarrays & GeneChip-Technologien

Qualifikationsziele/Kompetenzen:

- Die Studierenden können die Grundkonzepte der Informationskodierung, -transkription und -translation in Lebewesen benennen.
- Sie können einen einfachen Greedy-Algorithmus zur näherungsweise Lösung des Shortest-Common-Superstring-Problems angeben.
- Sie können für eine gegebene Modellierungsaufgabe entscheiden, ob sie mittels einer Markov-Kette oder mittels eines Hidden-Markov-Modells (HMM) gelöst werden kann.
- Sie können an Beispielen erklären, wie mittels dynamischer Programmierung die exakte Lösung einer gegebenen Fragestellung ermittelt werden kann.
- Sie können die vorgestellten Algorithmen und Modelle (in Matlab) implementieren.
- Sie können grundlegende Methoden des unüberwachten Lernens anwenden und deren Ergebnisse interpretieren.
- Sie können erklären, wie Microarray- und DNA-Chip-Technologien funktionieren.

Qualifikationsziele/Kompetenzen:

- Die Studierenden können die Grundkonzepte der Informationskodierung, -transkription und -translation in Lebewesen benennen.
- Sie können einen einfachen Greedy-Algorithmus zur näherungsweise Lösung des Shortest-Common-Superstring-Problems angeben.
- Sie können für eine gegebene Modellierungsaufgabe entscheiden, ob sie mittels einer Markov-Kette oder mittels eines Hidden-Markov-Modells (HMM) gelöst werden kann.
- Sie können an Beispielen erklären, wie mittels dynamischer Programmierung die exakte Lösung einer gegebenen Fragestellung ermittelt werden kann.
- Sie können die vorgestellten Algorithmen und Modelle (in Matlab) implementieren.
- Sie können grundlegende Methoden des unüberwachten Lernens anwenden und deren Ergebnisse interpretieren.
- Sie können erklären, wie Microarray- und DNA-Chip-Technologien funktionieren.

Constructive Alignment



UNIVERSITÄT ZU LÜBECK
SEKTION MINT
INSTITUT FÜR NEURO- UND BIOINFORMATIK

LZK Einführung in die Bioinformatik CS1400
Wintersemester 2022

Datum / Beginn: 20. Februar 2023, 12:00 Uhr
Bearbeitungszeit: 60 Minuten (60 Punkte entsprechenden dabei 20 XP)
Umfang: 9 Seiten (Deckblatt, 4 Aufgaben)

Nachname: _____
Vorname: _____
Matrikelnr.: _____
Studiengang: _____

Hiermit bestätige ich den vollständigen Erhalt der Klausurunterlagen.

Datum, Ort _____ Unterschrift _____

Schreiben Sie bitte Ihren Namen auf *jeden* Zettel!
Schreiben und zeichnen Sie bitte *nicht* mit Bleistift!
Erlaubte Hilfsmittel: *nicht programmierbarer Taschenrechner und Lineal*

Aufgabe Nr.:	1	2	3	4	Summe
Punktzahl:	15	15	15	15	60
Davon erreicht:					

Qualifikationsziele/Kompetenzen:

- Die Studierenden können die Grundkonzepte der Informationskodierung, -transkription und -translation in Lebewesen benennen.
- Sie können einen einfachen Greedy-Algorithmus zur näherungsweise Lösung des Shortest-Common-Superstring-Problems angeben.
- Sie können für eine gegebene Modellierungsaufgabe entscheiden, ob sie mittels einer Markov-Kette oder mittels eines Hidden-Markov-Modells (HMM) gelöst werden kann.
- Sie können an Beispielen erklären, wie mittels dynamischer Programmierung die exakte Lösung einer gegebenen Fragestellung ermittelt werden kann.
- Sie können die vorgestellten Algorithmen und Modelle (in Matlab) implementieren.
- Sie können grundlegende Methoden des unüberwachten Lernens anwenden und deren Ergebnisse interpretieren.
- Sie können erklären, wie Microarray- und DNA-Chip-Technologien funktionieren.

Biggs, J. & Tang, C. (2011): Teaching for Quality Learning at University. 4th Edition. New York; Maidenhead: OpenUniversity Press.

CA in der Bioinformatik



UNIVERSITÄT ZU LÜBECK
SEKTION MINT
INSTITUT FÜR NEURO- UND BIOINFORMATIK

LZK Einführung in die Bioinformatik CS1400
Wintersemester 2022

Datum / Beginn: 20. Februar 2023, 12:00 Uhr
Bearbeitungszeit: 60 Minuten (60 Punkte; entsprechendes dabei 20 XP)
Umfang: 9 Seiten (Deckblatt, 4 Aufgaben)

Nachname: _____
Vorname: _____
Matrikelnr.: _____
Studiengang: _____

Hiermit bestätige ich den vollständigen Erhalt der Klausurunterlagen.

Datum, Ort: _____ Unterschrift: _____

Schreiben Sie bitte Ihren Namen auf *jeden* Zettel!
Schreiben und zeichnen Sie bitte *nicht* mit Bleistift!
Erlaubte Hilfsmittel: nicht programmierbarer Taschenrechner und Lineal

Aufgabe Nr.:	1	2	3	4	Summe
Punktzahl:	15	15	15	15	60
Davon erreicht:					

Klausuraufgaben

Aufgabe 1
Hidden Markov Modelle

Aufgabe 2
K-Means-Algorithmus

Aufgabe 3
Shortest Common Superstring

Aufgabe 4
Lokales Sequence Alignment (EBI)

Kompetenzen (Modulbeschreibung)

Die Studierenden können die Grundkonzepte der Informationskodierung, -transkription und -translation in Lebewesen benennen.

Sie können einen einfachen Greedy-Algorithmus zur näherungsweise Lösung des Shortest-Common-Superstring-Problems angeben.

Sie können für eine gegebene Modellierungsaufgabe entscheiden, ob sie mittels einer Markov-Kette oder mittels eines Hidden-Markov-Modells (HMM) gelöst werden kann.

Sie können an Beispielen erklären, wie mittels dynamischer Programmierung die exakte Lösung einer gegebenen Fragestellung ermittelt werden kann.

Sie können die vorgestellten Algorithmen und Modelle implementieren (in Java)

Sie können grundlegende Methoden des unüberwachten Lernens anwenden und deren Ergebnisse interpretieren.

Sie können erklären, wie Microarray- und DNA-Chip-Technologien funktionieren.

Klausur als Maß für Kompetenzen?



Modul CS1400-KP04, CS1400

Einführung in die Bioinformatik (EinBioinfo)

Dauer:	Angebotsturnus:	Leistungspunkte:
1 Semester	Jedes Wintersemester	4

Lehrveranstaltungen:	Workload:
- CS1400-V: Einführung in die Bioinformatik (Vorlesung, 2 SWS)	- 45 Stunden Präsenzstudium
- CS1400-Ü: Einführung in die Bioinformatik (Übung, 1 SWS)	- 20 Stunden Prüfungsvorbereitung
	- 55 Stunden Selbststudium

Lehrinhalte:

- Leben, Evolution & das Genom
- Sequence Assembly - Maschinelles Auslesen von genetischer Information
- DNA Sequenzmodelle & Hidden Markov Ketten
- Viterbi-Algorithmus
- Sequence Alignment & Dynamische Programmierung
- Unüberwachte Datenanalyse (k-means, PCA, ICA)
- DNA Microarrays & GeneChip-Technologien

Qualifikationsziele/Kompetenzen:

- Die Studierenden können die Grundkonzepte der Informationskodierung, -transkription und -translation in Lebewesen benennen.
- Sie können einen einfachen Greedy-Algorithmus zur näherungsweise Lösung des Shortest-Common-Superstring-Problems angeben.
- Sie können für eine gegebene Modellierungsaufgabe entscheiden, ob sie mittels einer Markov-Kette oder mittels eines Hidden-Markov-Modells (HMM) gelöst werden kann.
- Sie können an Beispielen erklären, wie mittels dynamischer Programmierung die exakte Lösung einer gegebenen Fragestellung ermittelt werden kann.
- Sie können die vorgestellten Algorithmen und Modelle (in Matlab) implementieren.
- Sie können grundlegende Methoden des unüberwachten Lernens anwenden und deren Ergebnisse interpretieren.
- Sie können erklären, wie Microarray- und DNA-Chip-Technologien funktionieren.

Modul CS1400-KP04, CS1400

Einführung in die Bioinformatik (EinBioinfo)

Dauer:	Angebotsturnus:	Leistungspunkte:
1 Semester	Jedes Wintersemester	4

Lehrveranstaltungen:	Workload:
- CS1400-V: Einführung in die Bioinformatik (Vorlesung, 2 SWS)	- 45 Stunden Präsenzstudium
- CS1400-Ü: Einführung in die Bioinformatik (Übung, 1 SWS)	- 20 Stunden Prüfungsvorbereitung
	- 55 Stunden Selbststudium

Was ist eigentlich mit den verbleibenden **98 Stunden Präsenz- und Selbststudium?**

Epic Win



“An **epic win** is an outcome so **extraordinarily positive**, you didn't even know it was possible until you achieved it.”





Vorlesungen sind Anti-Spiele!

(take-home-message I)

Zwei Hauptprobleme:

Mangelnde Eingebundenheit (akademische Integration!)

Fehlendes (direktes) Feedback (Kompetenzempfinden)

Gameful Learning



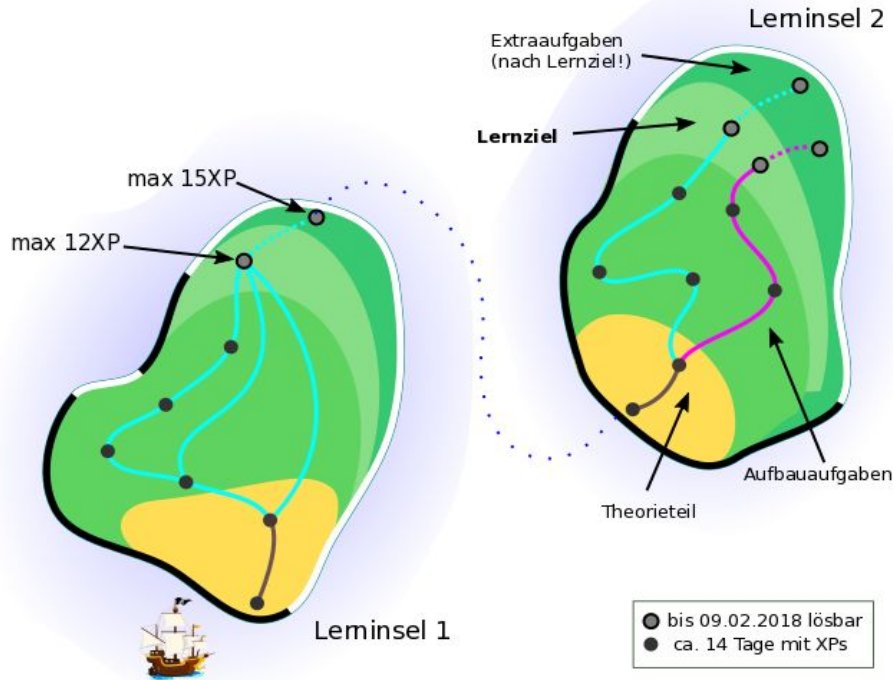
- from zero to hero -



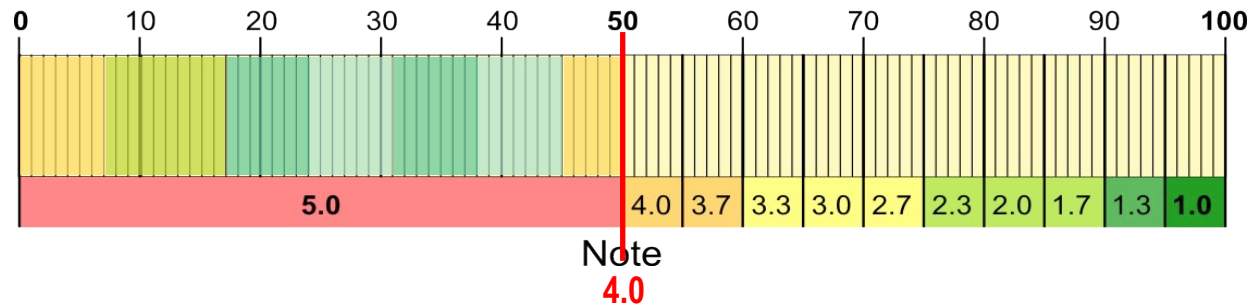
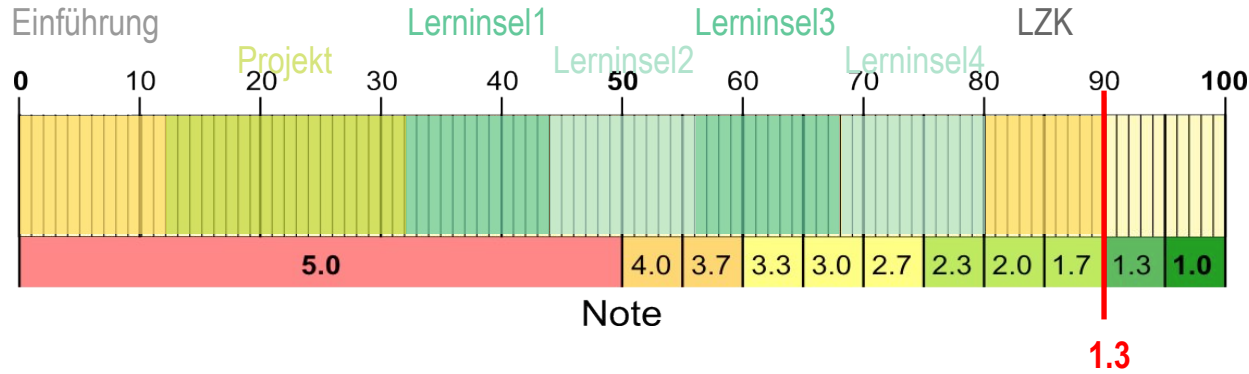
Madany Mamlouk, A. (2023): Gameful Learning als Ausgangspunkt für eine positiv bestärkende Lernkultur in den MINT-Fächern. In: Neues Handbuch Hochschullehre, Band 109, Signatur C 2.44

Lerninseln

Lerninseln (schematisch)



XP-Bewertung



Kompetenzorientierung auf Lerninseln



Kompetenzen laut Modulbeschreibung

Die Studierenden können die Grundkonzepte der Informationskodierung, -transkription und -translation in Lebewesen benennen.

Sie können einen einfachen Greedy-Algorithmus zur näherungsweise Lösung des Shortest-Common-Superstring-Problems angeben.

Sie können für eine gegebene Modellierungsaufgabe entscheiden, ob sie mittels einer Markov-Kette oder mittels eines Hidden-Markov-Modells (HMM) gelöst werden kann.

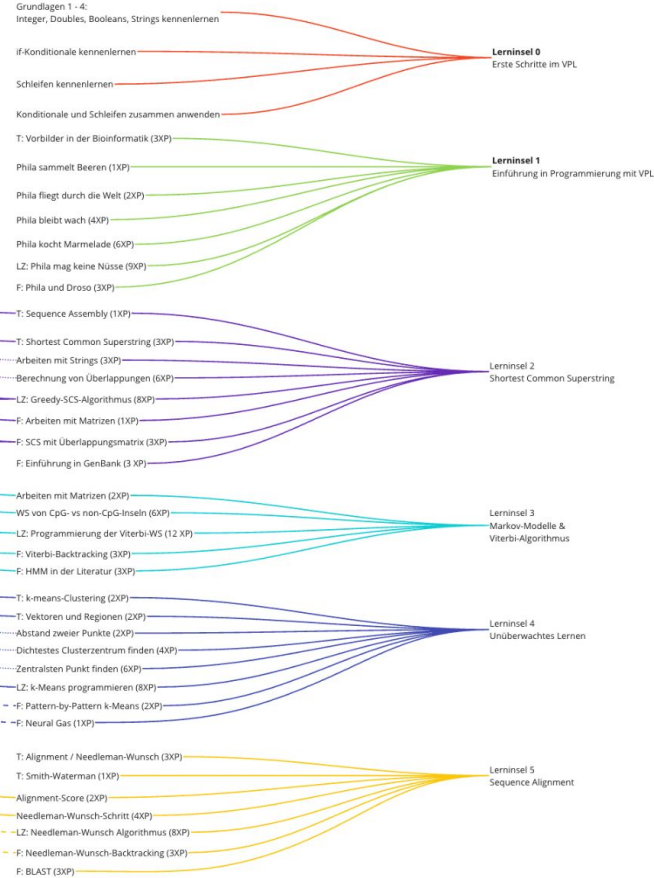
Sie können an Beispielen erklären, wie mittels dynamischer Programmierung die exakte Lösung einer gegebenen Fragestellung ermittelt werden kann.

Sie können die vorgestellten Algorithmen und Modelle implementieren (in Java)

Sie können grundlegende Methoden des unüberwachten Lernens anwenden und deren Ergebnisse interpretieren.

Sie können erklären, wie Microarray- und DNA-Chip-Technologien funktionieren.

Lerninseln



Lerninsel SCS



Kompetenzen laut Modulbeschreibung

Die Studierenden können die Grundkonzepte der Informationskodierung, -transkription und -translation in Lebewesen benennen.

Sie können einen einfachen Greedy-Algorithmus zur näherungsweisen Lösung des Shortest-Common-Superstring-Problems angeben.

Sie können für eine gegebene Modellierungsaufgabe entscheiden, ob sie mittels einer Markov-Kette oder mittels eines Hidden-Markov-Modells (HMM) gelöst werden kann.

Sie können an Beispielen erklären, wie mittels dynamischer Programmierung die exakte Lösung einer gegebenen Fragestellung ermittelt werden kann.

Sie können die vorgestellten Algorithmen und Modelle implementieren (in Java)

Sie können grundlegende Methoden des unüberwachten Lernens anwenden und deren Ergebnisse interpretieren.

Sie können erklären, wie Microarray- und DNA-Chip-Technologien funktionieren.

Lerninsel 2 - SCS

- T: Sequence Assembly (1XP)
- T: Shortest Common Superstring (3XP)
- Arbeiten mit Strings (3XP)
- Berechnung von Überlappungen (6XP)
- LZ: Greedy-SCS-Algorithmus (8XP)
- F: Arbeiten mit Matrizen (1XP)
- F: SCS mit Überlappungsmatrix (3XP)
- F: Einführung in GenBank (3 XP)

Studierendenfeedback mit Gradbook XP

Dieses Projekt wird von der Stiftung Innovation in der Hochschullehre im Rahmen des Vorhabens **HySkiLabs - Gesund(heit) lehren und lernen in hybriden Skills-Labs** gefördert.



Stiftung
Innovation in der
Hochschullehre

Bisher: Wilde Punktelisten



Lerninsel 2				
L2: Theorie				
QUIZ	Theorie 1: Sequence Assembly (1 XP)	25.00 %	1.00	0-1 100.00 %
QUIZ	Theorie 2: Shortest Common Superstring (3 XP)	75.00 %	1.07	0-3 35.67 %
AGGREGATION	L2: Theorie total Include empty grades.	33.33 %	2.07	0-4 51.75 %
L2: Grundlagen				
VIRTUAL PROGRAMMING LAB	Grundlagen 1: Arbeiten mit Strings (3 XP)	0.00 %	1.50	0-3 50.00 %
VIRTUAL PROGRAMMING LAB	Grundlagen 2: Berechnung von Überlappungen (6 XP)	0.00 %	6.00	0-6 100.00 %
VIRTUAL PROGRAMMING LAB	Lernziel Greedy-SCS Algorithmus (8 XP)	100.00 %	8.00	0-8 100.00 %
AGGREGATION	L2: Grundlagen total Highest grade. Include empty grades.	66.67 %	8.00	0-8 100.00 %
L2: Fortgeschritten				
VIRTUAL PROGRAMMING LAB	Fortgeschritten 1: Arbeiten mit Matrizen (1 XP)	-	1.00	0-1 100.00 %
VIRTUAL PROGRAMMING LAB	Fortgeschritten 2: SCS mit Überlappungsmatrix (3 XP)	-	1.88	0-3 62.67 %
QUIZ	Fortgeschritten A: Einführung in GenBank (3 XP)	-	× 0.00	0-3 0.00 %
CALCULATED GRADE	L2: Fortgeschritten total Highest grade. Include empty grades.	25.00 % (Extra credit)	1.88	0-3 62.67 %
AGGREGATION	Lerninsel 2 total	-	11.95	0-12 99.58 %

Grade item	Grade	Range	Percentage
Einführung in die Bioinformatik (VUE) – CS1400			
CALCULATED GRADE	Lerninsel 0	-	0-1 -
CALCULATED GRADE	Lerninsel 1	12.00	0-15 80.00 %
CALCULATED GRADE	Lerninsel 2	12.00	0-15 80.00 %
CALCULATED GRADE	Lerninsel 3	12.00	0-15 80.00 %
CALCULATED GRADE	Lerninsel 4	6.00	0-15 40.00 %
CALCULATED GRADE	Lerninsel 5	11.00	0-15 73.33 %
CALCULATED GRADE	Projekt	14.50	0-20 72.50 %
ASSIGNMENT	Weihnachtsquest	-	0-3 -
MANUAL ITEM	Extrapunkte	-	0-10 -
CALCULATED GRADE	Lerninsel-Block (insgesamt)	67.50	0-80 84.38 %
CALCULATED GRADE	Klausurpunkte	-	0-60 -
CALCULATED GRADE	Klausur-XP's	-	0-20 -
CALCULATED GRADE	Kurspunkte (Note)	67.50 (3,0)	0-100 67.50 %
Lerninseln			

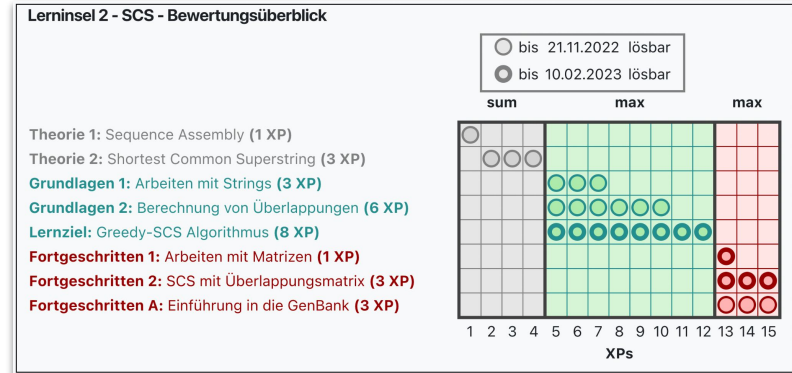
Wilde Punkteliste SCS



Punkteüberblick Lerninsel 2

Lerninsel 2					
L2: Theorie					
QUIZ	Theorie 1: Sequence Assembly (1 XP)	25.00 %	1.00	0-1	100.00 %
QUIZ	Theorie 2: Shortest Common Superstring (3 XP)	75.00 %	1.07	0-3	35.67 %
AGGREGATION	L2: Theorie total Include empty grades.	33.33 %	2.07	0-4	51.75 %
L2: Grundlagen					
VIRTUAL PROGRAMMING LAB	Grundlagen 1: Arbeiten mit Strings (3 XP)	0.00 %	1.50	0-3	50.00 %
VIRTUAL PROGRAMMING LAB	Grundlagen 2: Berechnung von Überlappungen (6 XP)	0.00 %	6.00	0-6	100.00 %
VIRTUAL PROGRAMMING LAB	Lernziel: Greedy-SCS Algorithmus (8 XP)	100.00 %	8.00	0-8	100.00 %
AGGREGATION	L2: Grundlagen total Highest grade. Include empty grades.	66.67 %	8.00	0-8	100.00 %
L2: Fortgeschritten					
VIRTUAL PROGRAMMING LAB	Fortgeschritten 1: Arbeiten mit Matrizen (1 XP)	-	1.00	0-1	100.00 %
VIRTUAL PROGRAMMING LAB	Fortgeschritten 2: SCS mit Überlappungsmatrix (3 XP)	-	1.88	0-3	62.67 %
QUIZ	Fortgeschritten A: Einführung in GenBank (3 XP)	-	0.00	0-3	0.00 %
CALCULATED GRADE	L2: Fortgeschritten total Highest grade. Include empty grades.	25.00 % (Extra credit)	1.88	0-3	62.67 %
AGGREGATION	Lerninsel 2 total	-	11.95	0-12	99.58 %

Fristen und Verrechnung aus dem Kurs (Lerninsel 2)



Gradbook XP - Radarplots



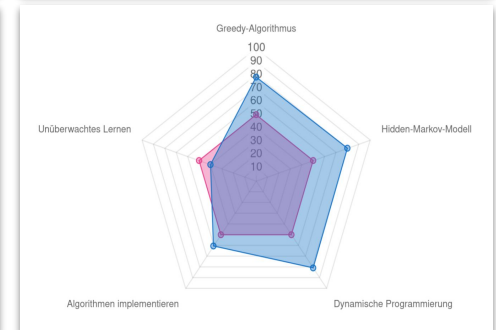
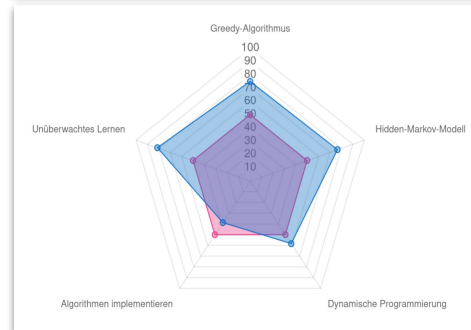
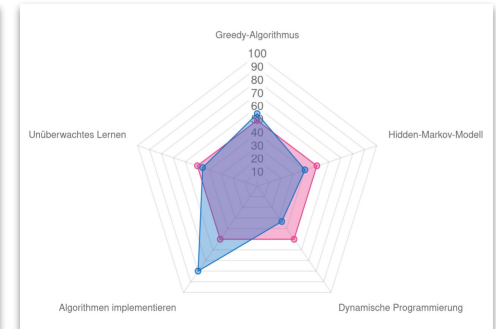
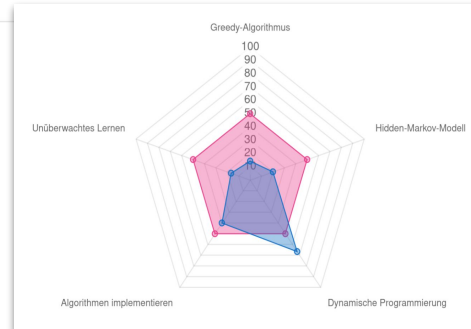
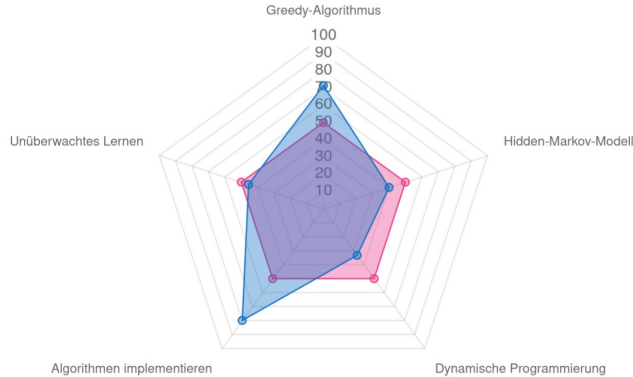
Test Home Dashboard My courses

AI
RS

Course Participants Grades **Competencies** Gradebook XP

Kompetenzen

■ You ■ Success



Gradbook XP - Backstage



Test Home Dashboard My courses Site administration

🔔 🗨️ AU ▾ Edit mode



Gradebook XP

Oberfläche zum Anlegen von neuen Kompetenzen (in moodle)

Go back

List of competencies

Greedy-Algorithmus

ID: 5 Parent ID: 0

Name: Greedy-Algorithmus Parent Name:

Description: Sie können einen einfachen Greedy-Algorithmus zur näherungsweise Lösung des Shortest-Common-Superstring-Problems angeben.

Edit competency

Delete competency

Hidden-Markov-Modell

ID: 6 Parent ID: 0

Name: Hidden-Markov-Modell Parent Name:

Description: Sie können für eine gegebene Modellierungsaufgabe entscheiden, ob sie mittels einer Markov-Kette oder mittels eines Hidden-Markov-Modells (HMM) gelöst werden kann.

Edit competency

Delete competency

Add competency



Gradbook XP - Wissenskanten



Course Settings Participants **Grades** Reports More ▾

Greedy-Algorithmus

Zuweisen von (gewichteten) Lerneinheiten zu Kompetenzen

Parent

Competency name

Competency description

Assignments

Projektabgabe
Theorie 2: Vektoren und Regionen (2 XP)
Weihnachtsquest
Theorie: Vorbilder in der Bioinformatik (3 XP)
Fortgeschritten A: Einführung in GenBank (3 XP)
Fortgeschritten A: HMM in der Literatur (3 XP)
Theorie 1: k-Means-Clustering (2 XP)
Fortgeschritten A: BLAST (3 XP)
Theorie 1: Alignment / Needleman-Wunsch (3 XP)
Theorie 2: Smith-Waterman (1 XP)
Grundlagen 1: Integer
Grundlagen 2: Gebrochene Zahlen
Grundlagen 3: Booleans
Grundlagen 4: Strings
Grundlagen 5: if-Konditionale
Grundlagen 6: Schleifen
Grundlagen 7: Konditionale und Schleifen
Fortgeschritten: Phila und Drosio (3 XP)



Theorie 1: Sequence Assembly (1 XP) | 1
Theorie 2: Shortest Common Superstring (3 XP) | 1
Grundlagen 1: Arbeiten mit Strings (3 XP) | 1
Grundlagen 2: Berechnung von Überlappung
Lernziel: Greedy-SCS Algorithmus (8 XP) | 1
Fortgeschritten 1: Arbeiten mit Matrizen (1 XP) | 1
Fortgeschritten 2: SCS mit Überlappungsmat

Weight

Save changes Cancel

Okay, what's next?



- Integration & Evaluation in laufender Vorlesung
- Erweiterung von komplexen Verrechnungen von XPs
- Anbindung an OER- oder andere Wissensquellen
- Recommendation-System für externe Lernquellen
- RollOut als Moodle-Plugin uniweit oder sogar noch weiter
- Wissenslandschaften teilen / (weiter-)entwickeln / beforschen
- Fachdidaktische Beforschung der Kompetenzübergänge
- KI-Agenten, die die Lern-/Lehrlandschaft nutzen

Gamers



Phase I

"I will succeed... ..or die trying!"



Phase II

"I didn't even know this is possible!"



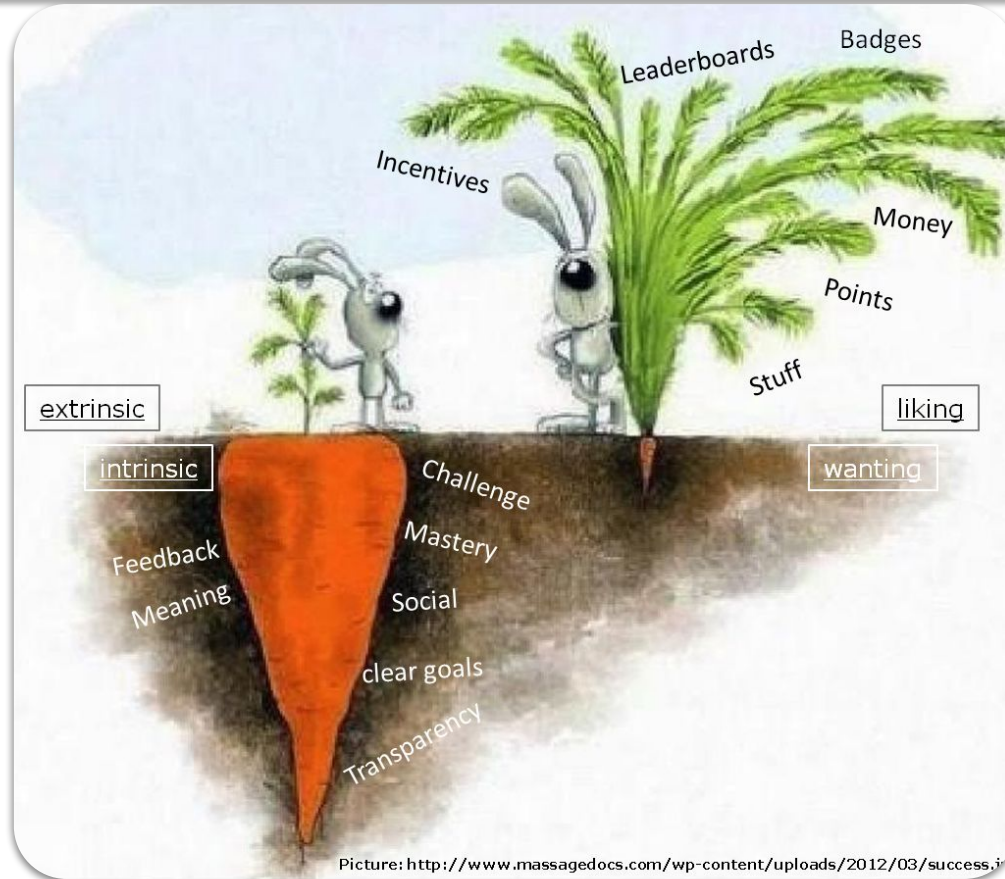
Phase III

"I already have plans with my new achievements!"

Gameful Learning vs Gamification



Gameful Learning



Gamification

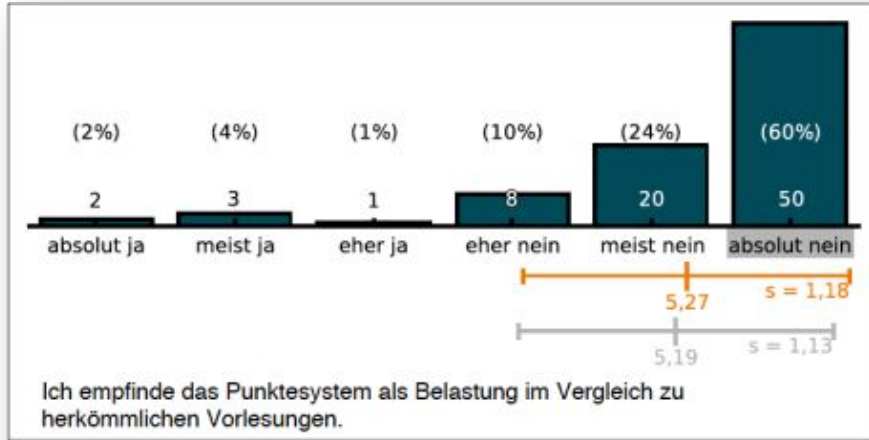
Kompetenzorientierung im Wandel



Beispiel: Shortest Common Superstring	KS1 (Wissen)	KS2 (Verstehen)	KS3 (Anwenden)	max XP
Hamilton-Graph zeichnen, Fusion des Greedy-Algorithmus an dem Graph durchführen.	Klausur (80%) Theorie (4XP)			4XP
Konkatenation von Strings, Anhängen von Zeichen, Methoden .equals() und .charAt() kennenlernen und üben	Basis I (4XP)			
Obiges Wissen kombinieren, um zwei Strings zu vergleichen und die größte Überlappung zu berechnen.		Klausur (20%) Basis II (6XP)		
Programmierung eines Greedy-Algorithmus, der eine Näherung an den SCS berechnet.			Übung (alt) Lernziel (8XP)	8XP
Programmierung des Greedy mit einer LookUp-Table (schnellere Version)			(Sternchen) Expert (3XP)	3XP
TOTAL				15XP

Kompetenzstufen nach: Bloom, B.: Taxonomie von Lernzielen im kognitiven Bereich, Weinheim und Basel, 1976

Was sagen die Studierenden?



- Statt für andere Veranstaltungen zu lernen, habe ich lieber freiwillig Bioinfo gemacht, obwohl ich bereits eine Punktzahl erreicht hatte, mit der ich zufrieden bin.
- Das ist für mich der größte Erfolg: **Ich habe so richtig Lust bekommen Neues zu Lernen!**
- Das XP-System ist gut, weil der **Druck** aus der Veranstaltung genommen wird.
- Bei mir hat sich absolut der 'Computerspiel Effekt' eingestellt: Punkte sammeln, **Spaß** sich Problemen zu stellen und am Ende **selbstständig** zu einer Lösung zu gelangen!
- Bei jeder Veranstaltung XPs zu sammeln würde mir einen **Fokus auf persönliche Schwerpunkte** ermöglichen.

VPL - Virtual Programming Lab



- Aufgaben können sogar am Handy gelöst werden.
- Feedback sofort
- Punkte werden automatisch gutgeschrieben
- Umfangreicher Plagiatscheck

Beschreibung Abgabeliste Ähnlichkeit Testaktivität

Abgabe Bearbeiten Abgabesicht Bewertung Vorige Abgabeliste

Datei > Bearbeiten > Optionen > Vollbildmodus | Speichern Ausführen Debuggen Evaluieren Konsole | Über

```
Aufgabe.java
1- /*Du hast als Eingabe die Strings eingabe1 und eingabe2 und sollst die Überlappung
2- *Das Ergebnis ist der Wert, der am Ende in der Variable maxOverlap steht.*/
3-
4- int maxOverlap = 0;
5-
6- /*Dabei musst du darauf achten, dass deine Schleife nicht über die Länge des
7- *Dies schaffst du, indem du die Länge beider Strings mit dem Befehl min(Länge
8- *und in einem Integer speicherst.*/
9- int lEingabe1=eingabe1.length();
10- int lEingabe2=eingabe2.length();
11- int maxVgl=min(lEingabe1,lEingabe2);
12-
13- for (int i=0;i<maxVgl;i++)
14- {
15-     System.out.println("line\n");
16-     if (eingabe1.substring(lEingabe1-(1+i),lEingabe1).equals(eingabe2.substring(
17- }
```

Bewertungsvorschlag: 0 / 3

Kommentare

Ergebnisse
Das Programm bestand 0/8 Tests (0/5 offene und 0/3 versteckte).
Die bestandenen Tests liefern 0/3 Punkte.

Offene Tests
1. Fehler: Eingabe

ACGT
TGCA

lieferte

line
line
line

Ausführung

VPL 3.1.5

◀ Grundlagen 1: Arbeiten mit Strings Direkt zu: ▶ Lernziel: Annäherung des Shortest Common Superstring ▶

VPL - Screenshot



- Testszenarien zeigen anschaulich, was (nicht) geklappt hat.

Konsole: Verbunden (Läuft: 33 seg)

Maximum Overlap

Eingabe-Strings	
ACGT	TGCA
Dein Overlap ist: 3	

Deine Lösung:

```
ACGT
TGCA
```

Musterlösung:

```
ACGT
  TGCA
```

Dein Ergebnis ist falsch

Lösung

◀ 1 von 5 ▶

Wie VL entstehen sollten...

