UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
NEURO- UND BIOINFORMATIK

Praktikum

Warum der genetische Code universell ist

Seves Keser
Matr.-Nr.: 628521, Medizinische Informatik

Betreuer:
PD Dr. rer. nat. Amir Madany Mamlouk

Lübeck, den 13. Juli 2019

Inhaltsverzeichnis

1 Einleitung

Der genetische Code, also die Zuordnung der 20 verschiedenen Aminosäuren zu jeweils 3 aufeinander folgenden Nukleotiden ist universell auf diesem Planeten. Mithilfe dieser Zuordnung (Abbildung 1) wird festgelegt wie die Nukleotidsequenz aus dem Genom eines Organismus in Polypeptidketten (Proteine) abgelesen und übersetzt wird. Es wird vermutet, dass der Code universell verwendet wird, da er die verschiedenen Eigenschaften der Aminosäuren im Falle einer Mutation bestmöglich konserviert, also dass die Wahrscheinlichkeit groß ist, dass z.B Aminosäuren mit ähnlichen Polaritäten miteinander ausgetauscht werden.
Ein viel verfolgter Ansatz ist es, den natürlichen genetischen Codes mit zufälligen genetischen Codes, bei denen die Codesonne blockweise permutiert wurde, zu vergleichen. Bisher wurde dort jedoch stets nur ein Vergleichskriterium verwendet. In dieser Arbeit soll eine Kombination von Kriterien gesucht werden, die die Wahrscheinlichkeit „bessere“, also gegen Mutationen robustere genetische Codes in den Zufallscodes zu finden minimiert. Der Greedy-Algorithmus soll zudem angepasst werden sodass der berechnete Score sein Minimum nur erreicht wenn alle Teilscores des untersuchten Codes besser sind als die des natürlichen Codes.
Als Kernziel soll die Hypothese folgende Hypothese geprüft werden:
„Der natürliche genetische Code ist durch die Evolution ausgewählt worden, da er die verschiedenen Charakteristika der Aminosäuren im falle einer Sequenzmutation optimal konserviert.“
Ein Hinweis dafür dass diese Hypothese richtig sein könnte wäre es, wenn unter Berücksichtigung verschiedener Charakteristika der Aminosäuren die relative Anzahl der Zufallscodes mit einer besseren Konservierung sinkt.

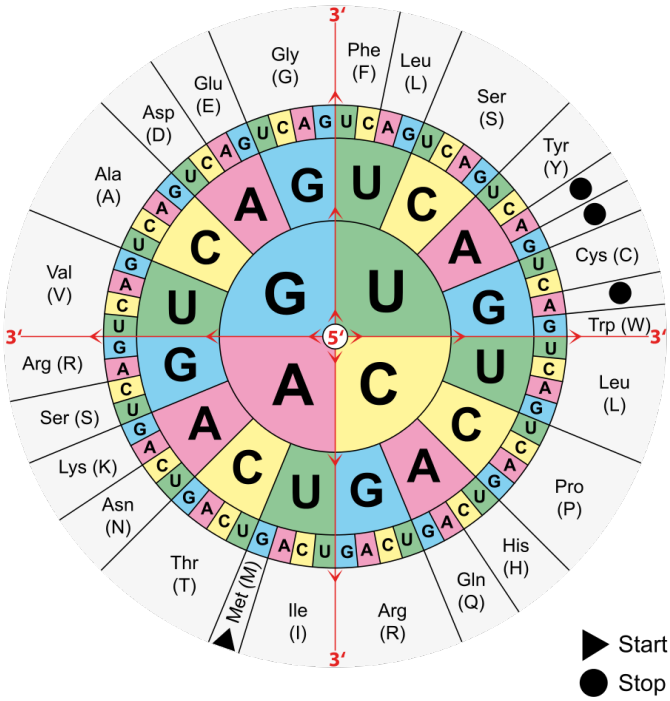

Abbildung 1: Die Zuordnung der Aminosäuren zu den 64 Codons der RNA [22]

2 Bisherige Arbeiten

David Haig, und Laurence D. Hurst veröffentlichten 1991 den Artikel „A Quantitative Measure of Error Minimization in the Genetic Code“[9]. In diesem wurde der natürliche genetische Code mit 10.000 Zufallscodes verglichen. Dabei verwenden sie verschiedenen Kriterien und stellen fest, dass besonders die Hydropobizität und die Polarität der Aminosäuren vom natürlichen genetischen Code besonders gut konserviert wird.
Stephan J. Freeland und Laurence D. Hurst erweiterten diese Berechnungen [10] und zeigten, dass es unter einer Million zufälliger genetischer Codes nur einen gibt der die Polarität der Aminosäuren besser konserviert. Die Arbeit von Freeland und Hurst betrachtete jedoch nur Punktmutationen. Eine Erweiterung auf Shiftmutationen führte dann R.Geyer in ihrer Arbeit durch[7]. Sie zeigte, dass der natürliche genetische Code auch für Shiftmutationen im Vergleich besser konserviert als ein Großteil der Zufallscodes. In meiner Bachelor- und Masterarbeit [5][6] erweiterte ich die Berechnungen um Gewichtungen aus realen Nukleotidsequenzen. Diese Gewichtungen beinhalten Informationen darüber, welche Nukleotide und Tripletts wahrscheinlicher in einer Sequenz vorzufinden sind und bewerten aus häufigeren Konstellationen entstehende Mutationen dann mit einem höheren Fehlerwert. Ebenfalls stellte ich in dieser Arbeit erstmals einen Greedy-Algorithmus vor, der gezielt dem „besten“ Code sucht.

3 Daten

3.1 Eigenschaften der Aminosäuren

In dieser Arbeit werden die bereits in meiner Masterarbeit verwendeten Charakteristika der Aminosäuren verwendet. Aus dem „CRC Handbook of Chemistry“[18] wurden

Tabelle 1: Polarität, Hydrophobitizät und Molekularvolumen der essentiellen Aminosäuren. (entspricht Tabelle 1 in [9])

Aminosäure	Polarität	Hydrophobizität	Molekularvolumen
Ala	7,0	1,8	31
Arg	9,1	-4,5	124
Asn	10,0	-3,5	54
Asp	13,0	-3,5	56
Cys	4,8	2,5	55
Glu	12,5	-3,5	83
Gln	8,6	-3,5	85
Gly	7,9	-0,4	3
His	8,4	-3,2	96
Ile	4,9	4,5	111
Leu	4,9	3,8	111
Lys	10,1	-3,9	119
Met	5,3	1,9	105
Phe	5,0	2,8	132
Pro	6,6	-1,6	32,5
Ser	7,5	-0,8	32
Thr	6,6	-0,7	61
Trp	5,2	-0,9	170
Tyr	5,4	-1,3	136
Val	5,6	4,2	84

folgende Charakteristika verwendet:

- M_r: Molekulargewicht
- pK_a: Negativer Logarithmus der Säuredissoziationskonstanten der COOH-Gruppen
- pK_b: Negativer Logarithmus der Säuredissoziationskonstanten der NH_2-Gruppen
- pI: pH-Wert am isoelektrischen Punkt

Die Werte dieser Eigenschaften sind Tabelle 2 zu entnehmen.

3.2 Genomsequenzen

Die Daten der Genomsequenzen wurden aus der Genbank entnommen. Dabei wurde im Rahmen dieser Arbeit das menschliche Chromosom 1 sowie eine Datei mit allen codierenden Sequenzen des Menschen (CCDS) verwendet. Die Sequenz des menschlichen

Tabelle 2: Molekulare Eigenschaften der essentiellen Aminosäuren

Aminosäure	M_r	pK_a	pK_b	pI
Ala	89,09	2,33	9,71	6,00
Arg	174,20	2,03	9,00	10,76
Asn	132,12	2,16	8,73	5,41
Asp	133,10	1,95	9,66	2,77
Cys	121,16	1,91	10,28	5,07
Glu	147,13	2,16	9,58	3,22
Gln	146,15	2,18	9,00	5,65
Gly	75,07	2,34	9,58	5,97
His	155,16	1,70	9,09	7,59
Ile	131,17	2,26	9,60	6,02
Leu	131,17	2,32	9,58	5,98
Lys	146,19	2,15	9,16	9,74
Met	149,21	2,16	9,08	5,74
Phe	165,19	2,18	9,09	5,48
Pro	115,13	1,95	10,47	6,30
Ser	105,09	2,13	9,05	5,68
Thr	119,12	2,20	8,96	5,60
Trp	204,23	2,38	9,34	5,89
Tyr	181,19	2,24	9,04	5,66
Val	117,15	2,27	9,52	5,96

Chromosoms wurde wie bereits in [6] beschrieben vorverarbeitet. Auch die Gewichtungsoptionen wurden von dort übernommen.

4 Methoden

4.1 Vergleichsparameter

In dieser Arbeit wird als Vergleichsparameter der GMS-Score verwendet. Dieser wird wie in [5] definiert berechnet und gibt eine Maßzahl für die Konservierungsfähigkeit eines genetischen Codes an. Kleinere Werte bedeuten, dass ein Code besser konserviert, also robuster gegen Mutationen ist, größere Werte dementsprechend dass es ein eher instabiler Code ist. Für die Berechnungen wird folgende Notation verwendet:

- $P(c_i)$ ist die Eigenschaft der Aminosäure, die durch das Codon c_i codiert wird.
- $P(M^j(c_i))$ ist die Eigenschaft der Aminosäure, die durch die j-Mutation des Codons c_i codiert wird.
- m_i ist die Anzahl der möglichen Mutationen, welche das Codon c_i nicht zu einem Stoppcodon machen. Für Punktmutationen gibt es an der ersten Position 174 und an den beiden anderen 176 mögliche Mutationen. Für Shiftmutationen können je Richtung 232 verschiedene Mutationen vorkommen.
- W sind die Gewichtungen zur Mutation

Die Standardabweichung bei Mutationen wird wie folgt berechnet:

$$D_x = \sum_{i=1}^{61} \sum_{j=1}^{m_i} (P(c_i) - P(M^j(c_i))^2 W) \tag{1}$$

Die Scores MS1, MS2 und MS3 repräsentieren den quadrierten Mittelwert der Abweichung durch eine Mutation, MS0 den Mittelwert über alle drei Codonpositionen.

$$MS1 = \frac{D_1}{m_1} \ , \ MS2 = \frac{D_2}{m_2} \ , \ MS3 = \frac{D_3}{m_3} \ , \ MS0 = \frac{D_1 + D_2 + D_3}{m_1 + m_2 + m_3} \tag{2}$$

R. Geyer führte die Scores rMS, lMS und fMS ein, welche die Abweichung der Polaritäten der Aminosäuren nach einem Verschieben des Leserasters bewerten.

$$rMS = \frac{D_r}{m_r} \ , \ lMS = \frac{D_l}{m_l} \ , \ fMS = \frac{D_r + D_l}{m_r + m_l} \tag{3}$$

Um einen Score zu haben, der sowohl über die Punkt- als auch über die Shiftmutationen summiert, wird der GMS verwendet:

$$GMS = \frac{D_1 + D_2 + D_3 + D_r + D_l}{m_1 + m_2 + m_3 + m_r + m_l} \tag{4}$$

4.2 Parameter für den Greedy-Algorithmus

Ein Greedy-Algorithmus wird verwendet um über mehrere Iterationen evolutionsähnlich die Ergebnisse zu mutieren, die besten zu selektieren um diese dann in der nächsten Iteration zu verwenden. Um den Algorithmus zu definieren, wird eine Maßzahl die minimiert werden soll, ein Verfahren zur Mutation sowie eine Definition der Selektion benötigt.

4.2.1 Vergleichsparameter

Der in [5] vorgestellte Greedy-Algorithmus minimiert den GMS-Score im Bezug auf die Polarität der Aminosäuren. Der GMS-Score ist definiert als ein Mittelwert über alle möglichen Mutationen. Dabei wird kein Wert darauf gelegt, dass alle der darin enthaltenen Werte niedriger sind als die entsprechenden Werte den natürlichen genetischen Codes sondern lediglich der Mittelwert minimiert. Dies führte zu dem Effekt, dass bei allen maximalkonservierenden Codes die durch den Algorithmus gefunden wurden der MS2-Wert schlechter war als der des natürlichen Codes. Um nach Codes zu suchen, die den natürlichen genetischen Code in allen Eigenschaften schlagen muss der Vergleichsparameter angepasst werden. Folgende Notationen werden verwendet:

- x ist der Vergleichsparameter der mit dem Algorithmus minimiert werden soll
- s_i sind die Scores der jeweiligen Eigenschaften des zu untersuchenden Codes
- c_i sind die Scores der jeweiligen Eigenschaften des natürlichen genetischen Codes
- $max(x, y)$ gibt den größeren der beiden Parameter zurück

Der Score x wird wie folgt berechnet:
Wenn $\exists i : c_i >= s_i$:

$$x = \sum_{i=0}^{n} max(0, s_i - c_i) \tag{5}$$

Wenn $\forall i : c_i > s_i$:

$$x = \sum_{i=0}^{n} c_i - s_i \tag{6}$$

Mit diesen Vergleichsparametern ist der Score x immer positiv solange mindestens ein Parameter des zu untersuchenden Codes größer ist als der Parameter des natürlichen Codes und immer negativ wenn alle Parameter des zu untersuchenden Codes kleiner sind als die des natürlichen Codes.

4.2.2 Mutationsverfahren

Jeder Code in der Suchmenge wird in jeder Iteration wie folgt mutiert und als neuer Code der Suchmenge hinzugefügt:

- Tausch eines Aminosäurepaars, jede mögliche Option wird verwendet (190 neue Codes)
- Tausch von 5 Aminosäurepaaren, nur eine Kombination je Code in der Suchmenge

Insgesamt werden so aus jedem Code 191 neue Codes erzeugt. Dabei entstehen auch doppelte Codes, dies werden nach der Permutation entfernt sodass jeder Code nur einmal in der Suchmenge vorkommt.

4.2.3 Selektionskriterium

Es werden in jeder Iteration die 1000 Codes mit dem niedrigsten Score x als Suchmenge in die nächste Iteration übernommen.

4.3 Technische Umsetzung

Grundlage der in dieser Arbeit durchgeführten Berechnungen ist das Java-Framework, welches bereits in meiner Bachelorarbeit und Masterarbeit zum Einsatz kam [5][6][21][24]. Dieses Framework wurde in einem neuen Git-Branch erweitert sodass es anstellen verschiedener Scores für die gleiche Eigenschaft der Aminosäuren nun den GMS für verschiedene Aminosäure-Eigenschaften berechnen und vergleichen kann.
Um RAM-Speicherplatz zu sparen wurde zudem die Garbage Collection optimiert sodass nach der Erstellung der Sequenzstatistiken die Daten der Sequenz aus dem Speicher entfernt werden. Die Berechnungen wurden auf Desktop-Prozessoren (Intel Core i7-4770 bzw Core i7-4790K) auf 3 Threads durchgeführt. Mehr Threads hätten die Berechnungsgeschwindigkeit erhöhen können, dies wäre jedoch zu Lasten anderer auf den Maschinen laufenden Anwendungen gegangen.
Der gesamte Code dieser Arbeit ist öffentlich auf GitHub verfügbar. Zu finden ist er unter `https://github.com/Phreag/DNA_Distribution_Analysis/tree/Intern_Multiparam`. Das Repository enthält zudem eine PDF-Version dieser Arbeit.
Zur besseren Reproduzierbarkeit sind in Tabelle 3 die für diese Arbeit verwendeten Methodenaufrufe dokumentiert. Alle dort genannten Methoden befinden sich in der Klasse `MainClass.java` und sind statisch sodass sie direkt aus der main-Methode aufgerufen werden können.

Um längere Sequenzen wie das menschliche Chromosom 1 verarbeiten zu können, ist es nötig den Java Heapspace zu vergrößern. Für tiefe Rekursionsaufrufe ist außerdem eine Erhöhung des Stacks nötig.
In dieser Arbeit wurden Java dafür die Programmparameter `Xmx4G -Xss8m -XX:+UseG1GC -XX:MaxHeapFreeRatio=20 -XX:MinHeapFreeRatio=10` mitgegeben. Die im Vergleich zu [6] ergänzten Parameter legen Fest dass der G1-Garbage Collector verwendet wurd und dass der Heap stets so groß ist, dass 10-20% davon frei sind. So wird nicht benötigter Speicher direkt an das System zurückgegeben.

Tabelle 3: Verwendete Codeaufrufe zur Berechnung der Daten

Methodenaufruf	Daten/Verwendung
`millionMultiParam()`	Tabelle 4
`billionMultiParamTA_TT_Chr1()`	Suche des Codes für Tabelle 6
`runCodeFinderMultiCharacterstics()`	Tabelle 8
`runCodeFinderMultiScore()`	Tabelle 7

5 Ergebnisse

5.1 Vergleich mit Zufallscodes

In bisherigen Arbeiten wurden die Optimierungseigenschaften des genetischen Codes immer auf eine Eigenschaft der Aminosäuren hin untersucht. In dieser Arbeit sollen alle eingangs genannten Eigenschaften gleichzeitig untersucht werden und nach Codes gesucht werden die alle diese Parameter besser konservieren als der natürliche genetische Code.
Erste Probe-Berechnungen zeigten, dass bereits ohne die Anwendung von Gewichtungen das untersuchen von einer Million Zufallscodes nicht ausreichen würde. Daher wurde ein neues Codeset generiert welches 100 Millionen Zufallscodes enthält. In diesem Set wurde anschließend nach Codes gesucht die alle dieser Eigenschaften besser konservieren. Als Gewichtungsoptionen wurden die Nukleotid-a-priori (NA), die Triplett-a-priori (TA) und die Triplett-Übergangswahrscheinlichkeiten (TT) verwendet. Die Wahrscheinlichkeiten dass ein zufälliger Code alle eingangs vorgestellten Charakteristika besser konserviert als der natürliche genetische Code ist Tabelle 4 zu entnehmen.
Die Gewichtungskombination [TA]+[TT] hatte bereits in vorherigen Arbeiten die Wahrscheinlichkeit für konservierendere Zufallscodes stark gesenkt, jedoch nicht soweit, dass keine Codes mehr gefunden werden konnten. Zudem wurde bisher zumeist nur in einer Million Zufallscodes gesucht.

Tabelle 4: Absolute Anzahl der Codes aus den 100 Millionen Zufallscodes, die für alle Charakteristika einen geringeren GMS-Score besitzen als der natürliche Code.

Gewichtung	Gewichtungsquelle	Anzahl konservierenderer Codes
keine	—	38
NA	Chromosom 1	31
TA	Chromosom 1	1
TT	Chromosom 1	3
NA+TA	Chromosom 1	1
NA+TT	Chromosom 1	5
TA+TT	Chromosom 1	**0**
NA+TA+TT	Chromosom 1	1
NA	CCDS	30
TA	CCDS	6
TT	CCDS	7
NA+TA	CCDS	6
NA+TT	CCDS	4
TA+TT	CCDS	3
NA+TA+TT	CCDS	3

Um Daten zu erhalten wurde die Berechnung nur für die Gewichtungskombination [TA]+[TT] nochmals mit einer Milliarde Zufallscodes durchgeführt. Bei dieser Berechnung wurde dann ein Code gefunden der alle für Charakteristika einen geringeren GMS-Score als der natürliche Code besitzt. Die Tabellen 5 und 6 zeigt die Codontabelle für

den natürlichen Code und den gefundenen Code. Die beiden Codes unterscheiden sich in 19 von 20 codierten Aminosäuren. Nur die Aminosäure Ser wird bei beiden Codes mit den gleichen Tripletts codiert.

Tabelle 5: Codierungstabelle des natürlichen genetischen Codes

1. Base	**2. Base**				**3. Base**
	T	**C**	**A**	**G**	
T	Phe	Ser	Tyr	Cys	**T**
					C
	Leu		*STOP*	*STOP*	**A**
				Trp	**G**
C	Leu	Pro	His	Arg	**T**
					C
			Gln		**A**
					G
A	Ile	Thr	Asn	Ser	**T**
					C
			Lys	Arg	**A**
	Met				**G**
G	Val	Ala	Asp	Gly	**T**
					C
			Glu		**A**
					G

Tabelle 6: Codierungstabelle des gefundenen besseren Codes für die Gewichtungsoption [TA]+[TT]

1. Base	2. Base				3. Base
	T	**C**	**A**	**G**	
T	Leu	Ser	Phe	Pro	**T**
					C
	Val		*STOP*	*STOP*	**A**
				Asp	**G**
C	Val	Tyr	Ile	Gln	**T**
					C
			Trp		**A**
					G
A	Met	Cys	His	Ser	**T**
					C
	Gly		Arg	Gln	**A**
					G
G	Ala	Thr	Lys	Glu	**T**
					C
			Asn		**A**
					G

5.2 Ergebnisse des Greedy-Algorithmus

In meine Bachelorarbeit [5] hatten alle durch den Greedy-Algorithmus gefundenen besseren Codes gemeinsam, dass der MS2-Wert stets schlechter war als der den natürlichen genetischen Codes. Da jedoch die anderen Werte deutlich geringer waren, war auch der mimimierte GMS kleiner. Bei der Suche in Zufallscodes wurde dort nirgends ein Code gefunden der für alle Fehlerwerte (Ms1, MS2, MS3, rMS, lMS) geringere Werte aufweist. Die Anfangs beschriebene Anpassung soll genau dies beheben, denn so erreichen nur Codes die in allen Scores besser sind einen negativen Wert. Der Algorithmus wurde für diese Arbeit in zwei Konfigurationen angewendet:

- Variante 1: Suche nach dem Code mit dem kleinsten Score über alle einzelnen Mutationsarten (Ms1, MS2, MS3, rMS, lMS), Werte bezogen auf die Polarität der Aminosäuren.
- Variante 2: Suche nach dem Code der für alle Eigenschaften (Polarität, Hydrophobizität, Molekularvolumen etc) einen kleineren GMS besitzt.

Der Greedy-Algorithmus fand für beide Varianten optimale Codes, der Verlauf über die Iterationen war erwartbar, die Ergebnisse scheinen daher valide zu sein. Abbildung 2 zeigt den Verlauf des Minimum, Maximum und des Mittelwertes der errechneten Scores aller Codes in der Suchmenge über die ersten 40 Iterationen des Algorithmus. Alle Berechnungen wurden mit den Gewichtungen [TA]+[TT] durchgeführt. Die Codes die in den jeweiligen Durchläufen als die besten gefunden wurden, sind in Tabelle 7 und 8 dargestellt.

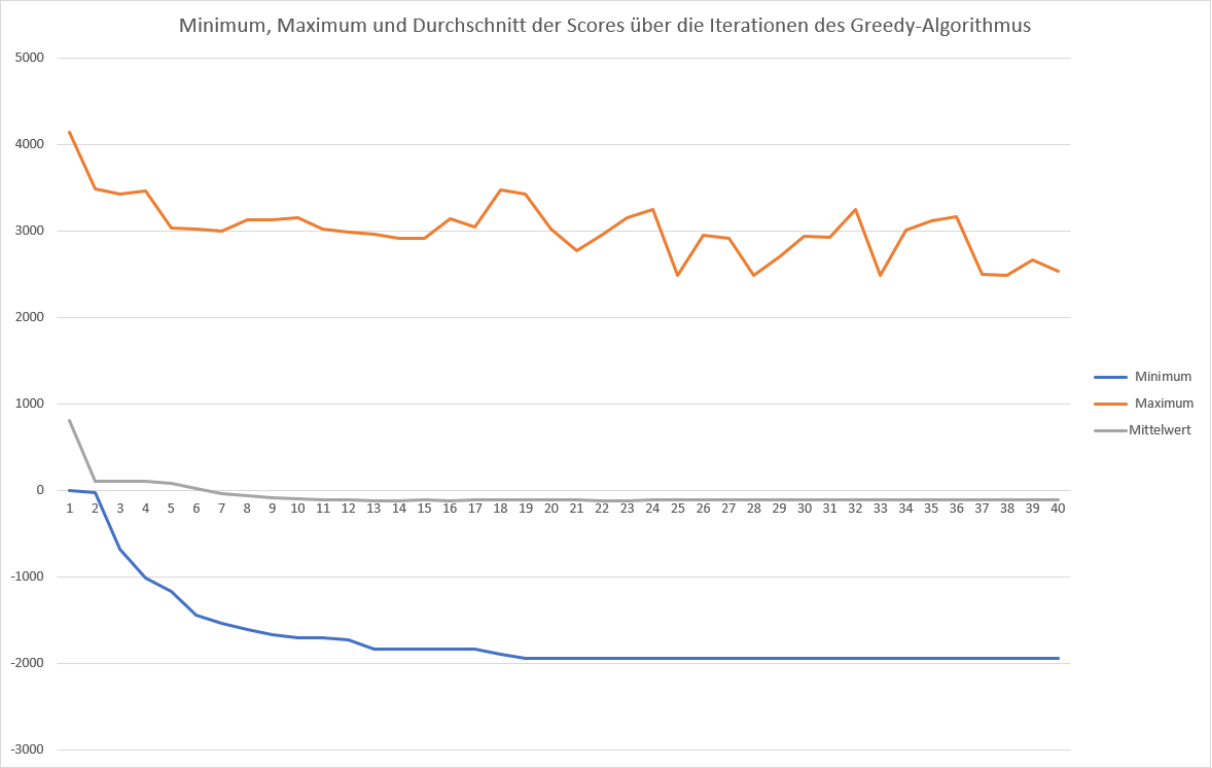

Abbildung 2: Verlauf der Werte des minimalen, maximalen und mittleren Scores über 40 Iterationen des Greedy-Algorithmus bei der Suche nach dem Code der die verschiedenen Eigenschaften der Aminosäuren maximal konserviert. Startmenge ist das Standard-Codeset von R. Geyer [7]

Tabelle 7: Codierungstabelle der bei Variante 1 gefundenen optimalen Codes. Die Aminosäuren Thr/Pro und Ile/Leu haben die gleichen Polaritäten und sind daher austauschbar.

1. Base	**2. Base**				**3. Base**
	T	**C**	**A**	**G**	
T	Arg	Gly	Thr/Pro	Glu	**T**
					C
	Gln		*STOP*	*STOP*	**A**
				Asp	**G**
C	Gln	Ala	Phe	Ser	**T**
					C
			Trp		**A**
					G
A	Thr/Pro	Val	Cys	Gly	**T**
					C
	Tyr		Ile/Leu	Ser	**A**
					G
G	Asn	His	Ile/Leu	Lys	**T**
					C
			Met		**A**
					G

Tabelle 8: Codierungstabelle des bei Variante 2 gefundenen optimalen Codes. Bemerkenswert ist, dass 5 der 20 Aminosäuren bei diesem Code durch die gleichen Tripletts codiert werden wie im natürlichen genetischen Code.

1. Base	**2. Base**				**3. Base**
	T	**C**	**A**	**G**	
T	Ile	Met	Cys	Phe	**T**
					C
	Leu		*STOP*	*STOP*	**A**
				Trp	**G**
C	Leu	Val	Ala	Lys	**T**
					C
			Gln		**A**
					G
A	Thr	Pro	Ser	Met	**T**
					C
	Gly		Asn	Lys	**A**
					G
G	Tyr	His	Asp	Arg	**T**
					C
			Glu		**A**
					G

6 Fazit

Die Anfangs aufgestellte Hypothese konnte in dieser Arbeit bestärkt werden.
Eine Berücksichtigung der verschiedenen Charakteristika der Aminosäuren senkte die Wahrscheinlichkeit zufällig einen besser konservierenderen Code zu finden in der verwendeten Suchmenge auf bis zu 1:1.100.000.000.
Die Anpassung des Greedy-Algorithmus zeigte, dass es auch Codes gibt die alle Eigenschaften der Aminosäuren besser konservieren als der natürliche genetische Code. Auch gibt es Codes die für jede Mutationsart bessere Werte erreichen.
Besonders interessant ist, dass der beste durch den Algorithmus (Variante 2) gefundene Code in 5 der 20 Aminosäuren mit dem natürlichen genetischen Code übereinstimmt. Dies kann auch als Hinweis gesehen werden, dass die zu belegende Hypothese korrekt ist. Zudem könnte es ein Hinweis darauf sein, dass - unter der Annahme dass die Hypothese korrekt ist - in diesem Algorithmus bereits fast alle selektionsrelevanten Kritieren berücksichtigt werden. Insgesamt konnte in dieser Arbeit gezeigt werden, dass der genetische Code besonders gut darin ist, alle Eigenschaften der Aminosäuren optimal zu konservieren.

Literatur

[1] F.H.C. Crick, *The origin of the genetic code*, Journal of Molecular Biology,Volume 38, Issue 3,1968, Pages 367-379,

[2] J. D. Watson, F. H. C. Crick, *A Structure for Deoxyribose Nucleic Acid* Nature, 1953

[3] E. Kreyszig, *Advanced Engineering Mathematics (Fourth ed.)*, p. 880, eq. 5. ISBN 0-471-02140-7, 1979

[4] B. Klaucke, *Der Effekt von nicht zufälligen Verteilungen in codierenden Daten auf die Mutationsstabilität des genetischen Codes*, Universität zu Lübeck, 2017. Bachelorarbeit

[5] Keser, S. *The DNA is More than One in a Million.* Universität zu Lübeck, 2016. Bachelorarbeit.

[6] Martenstein, S. (geb. Keser) *Titel* Universität zu Lübeck, 2019, Masterarbeit

[7] Geyer, R. *Frameshift Mutations of the Genetic Code and Their Impact on the Polarity Conservation of Amino Acids.* Universität zu Lübeck, 2014. Bachelorarbeit.

[8] Woese, C. R., Dugre, D. H., Saxinger, W. C., and Dugre, S. A. .*The Molecular Basis for the Genetic Code.* Proc Natl Acad Sci USA 55(4) (1996), 966–974.

[9] Haig, D. und Hurst, L. D. *A Quantitative Measure of Error Minimization in the Genetic Code.* Journal of Molecular Evolution 33 (1991), 412–417.

[10] Freeland, S. J. und Hurst, L. D. *The Genetic Code Is One in a Million.* Journal of Molecular Evolution 47 (1998), 238–248.

[11] *Xorshift RNGs*, George Marsaglia, The Florida State University, 2003

[12] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P Flannery, *Numerical Recipes, The Art of Scientific Computing, Third Edition*,Cambridge, Page 346f., 2007

[13] J. Kyte, RF. Doolittle, *A simple method for displaying the hydropathic character of a protein.*, J Mol Biol 157:105-132m, 1982

[14] L. Sagan, *On the origin of mitosing cells*, Journal of theoretical biology, 1967 Mar;14(3):255-74.

[15] International Human Genome Sequencing Consortium, *Finishing the euchromatic sequence of the human genome*, Nature 431, 931–945, 2004

[16] R. Grantham *Amino acid difference formula to help explain protein evolution.* Science 185:862-864, 1974

[17] Louise T. Chow, Richard E. Gelinas, Thomas R. Broker, Richard J. Roberts, *An amazing sequence arrangement at the 5' ends of adenovirus 2 messenger RNA*, Cell. 12 (1): 1–8., 1977

[18] D.R. Lide, *Handbook of Chemistry and Physics*, 85th Edition, CRC Press, 2004, Section 7, Page 1.

[19] W W. De Jong, L. Ryden, *Causes of more frequent deletions than insertions in mutations and protein evolution.* Nature. 1981 Mar 12;290(5802):157-9.

[20] M W. Nachman, S L. Crowell, *Estimate of the Mutation Rate per Nucleotide in Humans* GENETICS September 1, 2000 vol. 156 no. 1 297-304

[21] *BioJava 4.1.0 Open Source Bibliothek*, Website: http://biojava.org Quellcode: https://github.com/biojava/biojava/

[22] *Wikipedia: Code-Sonne*, gemeinfreies Bild, Quelle: https://de.wikipedia.org/wiki/Code-Sonne#/media/File:Aminoacids_table.svg

[23] *NCBI GenBank*, Website: https://www.ncbi.nlm.nih.gov/genbank/

[24] *Apache commons-io 1.3.2 Open Source Bibliothek*, Website: https://commons.apache.org/proper/commons-io/

[25] NCBI *The Consensus CDS (CCDS) project*, Jan. 2018. https://www.ncbi.nlm.nih.gov/projects/CCDS/CcdsBrowse.cgi

[26] NCBI, *The Genetic Codes*, Webseite: https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi?chapter=tgencodes#SG2

[27] UIPAC, *DNA Sequence Format*, Webseite: https://iupac.org/, Codes bezogen über: https://www.genomatix.de/online_help/help/sequence_formats.html#IUPAC