Technische Hochschule Bingen
Fachbereich 2 – Technik, Informatik und Wirtschaft
Angewandte Bioinformatik (B. Sc.)

prot-fin, ein toller Titel

Bachelorarbeit
abgegeben am: 26.08.2024
von: Franz-Eric Sill

Dozent: Prof. Dr. Asis Hallab

Zusammenfassung

...

Abstract

...

Literatur

[Kid+85] Akinori Kidera u. a. “Statistical analysis of the physical properties of the 20 naturally occurring amino acids”. In: *Journal of Protein Chemistry* 4.1 (Feb. 1985), S. 23–55. ISSN: 1573-4943. DOI: 10.1007/BF01025492. URL: https://doi.org/10.1007/BF01025492.

[LOH+14] MARC LOHSE u. a. “Mercator: a fast and simple web server for genome scale functional annotation of plant sequence data”. In: *Plant, Cell & Environment* 37.5 (2014), S. 1250–1258. DOI: https://doi.org/10.1111/pce.12231. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/pce.12231. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/pce.12231.

[Wan03] Avery Wang. “An Industrial Strength Audio Search Algorithm.” In: Jan. 2003.

Abbildungsverzeichnis

Tabellenverzeichnis

Inhaltsverzeichnis

1 Einleitung

. . . Wissenschaftlicher Kontext, zufällige Ähnlichkeit in Alignments . . .

Diese Bachelorarbeit, die auf dem Projekt prot-fin aufbaut, stellt sich diesem Problem zufälliger Ähnlichkeit und beschäftigt sich daher mit der Frage, ob es möglich ist, funktionsähnliche Proteine über ihre physikalischen Eigenschaften zu identifizieren, anstelle der lediglichen Buchstaben ihrer Aminosäuren, und ob das die Problematik umgeht.

Eine Grundlage hierfür bildet die Arbeit von Akinori Kidera *et. al.*, welcher in seiner Forschungsgruppe mittels statistischer Faktorenanalyse 188 physikalische Eigenschaften der 20 natürlich vorkommenden Aminosäuren auf lediglich 10 sogenannte Kidera-Faktoren reduziert hat, die zusammen all diese Eigenschaften am besten erklären [vgl. Kid+85]. So ist beispielsweise die Hydrophobizität ein ebensolcher Faktor, da diese mit vielen anderen Eigenschaften stark in Korrelation steht.

Der Einfluss eines jeden Faktors in einer Aminosäure lässt sich numerisch darstellen, sodass eine Aminosäuresequenz in 10 Vektoren übersetzt werden kann, welche nun ein statistisch auswertbares Abbild der physikalischen Struktur des Proteins erzeugen.

Der Algorithmus für die Analyse dieser Struktur ist von SHAZAM inspiriert, einer Anwendung, die Musiktitel anhand kürzester Tonaufnahmen identifiziert, selbst wenn diese Störgeräusche aufweisen. Basis hierfür stellt die Short Time Fourier Transformation (STFT) dar, welche in dem musikalischen Spektrum intervall-/fensterweise periodisch auftretende Signale analysiert, wodurch auch die Störgeräusche eine geringe Relevanz haben.

In Abbildung 1 wird dieser Sachverhalt für ein Fenster der STFT dargestellt. Es werden die Signalstärken für alle möglichen Frequenzen ermittelt, wobei das Reziproke einer Frequenz hier entspricht, jedes wievielte Element betrachtet wird. Bei Frequenz 0.5 wäre es also jedes $\frac{1}{0.5} = 2$te Element, hier offenbar sehr schwach ausgeprägt. Diese Signalstärke oder Amplitude der Frequenz wird über Summen der Originaldaten ermittelt. Frequenz 0 ist lediglich die Summe aller Eingabewerte, also hier allen Kidera-Fakor-Werten, die den numerischen Vektor der Eingabe-Aminosäurekette darstellten.

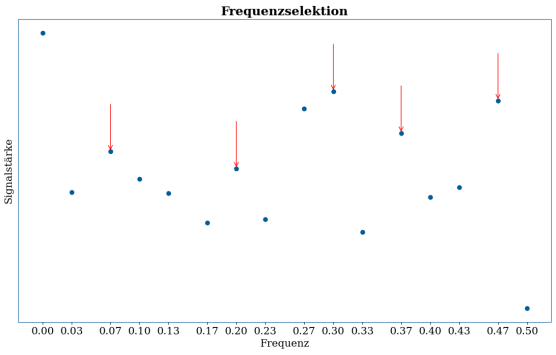

Abbildung 1: Spektralanalyse eines Fensters der STFT mit Markierung lokaler Maxima

Damit die Musikerkennung funktioniert, wird nun vorher eine Datenbank erstellt, welche die Periodizitäten der Eingabesongs mittels Hashing effizient auffindbar abspeichert, sodass der Abgleich mit einer Tonaufnahme sehr schnell und korrekt abläuft [vgl. Wan03]. Zudem werden hierfür aus den STFT-Ergebnissen auch nicht alle Frequenzen verwendet, sondern nur möglichst signifi-

kante davon. Bisher wird das mittels der lokalen Maxima der Amplituden erreicht.

Für die Anwendung auf Proteine werden statt des musikalischen Spektrums die numerischen Vektoren der Aminosäuresequenzen verwendet. Nun gibt es in prot-fin zwei verschiedene Anwendungsansätze:

1. **Single-Protein:** Als Eingabe erfolgt eine einzelne Aminosäuresequenz, für die das best passende Protein gesucht wird. Je mehr Übereinstimmung herrscht, desto funktionsähnlicher sollte es sein.
2. **Family-Matching:** Als Eingabe erfolgt eine Proteinfamilie. Die Periodizitäten, in denen sich alle Mitglieder dieser Familie ähneln, die also spezifisch für die Familie sind, werden verwendet, um Proteine zu finden, die auch in die Familie passen.

Um den Algorithmus für beide Ansätze auf die vergleichsweise kurzen Sequenzen von wenigen 100 Elementen abzustimmen (ein solcher Vektor für eine Sekunde Musik hätte etwa 40.000 Elemente), wurde in vorangegangenen Experimenten versucht, die Fenstergröße und die Überlappung zwischen diesen Fenstern bei der STFT zu optimieren, beziehungsweise auch die Anzahl gewählter Frequenzen, deren Amplituden auf Periodizität hindeuten. Hierbei zeigte sich allerdings eine schlechte Performanz hinsichtlich Speicher- und Laufzeitkomplexität. Die Wahl der Frequenzen der STFT Fenster und deren Abspeicherung müssen folglich noch verbessert werden.

Hierzu werden in dieser Arbeit mehrere Experimente angegangen.

1. **UniRef90-Sampling:** Je weniger Frequenzen in einem Fenster ausgewählt werden, desto weniger muss gespeichert werden. In diesem Experiment werden solche Fenster aus der UniRef90 Datenbank gesampelt. Sie enthält etwa 180 Mio. Aminosäuresequenzen, sodass aus jeder ein zufälliges Fenster gewählt wird. Die je Frequenz seltensten Amplituden sollen nun als Schwellwert das Wahlkriterium für eine Frequenz sein, was zu möglicherweise weniger selektierten Frequenzen mit dennoch guter Signifikanz führt.
2. **Filter Hashes:** Ein weiterer Ansatz die Datenbankgröße zu verringern ist das Entfernen von Hashes, die sehr häufig auftreten, also folglich wenig Informationsgehalt für die Identifikation eines Proteins haben. In diesem Experiment wird daher geprüft, wie streng das erfolgen darf, um nicht zu viel Daten zu verlieren, sodass das Matching dadurch beeinträchtigt würde.
3. **Target-Zone:** Die effiziente Abspeicherung mittels Hashing basiert darauf, die Frequenzen eines Fensters mit den Frequenzen der Nachfolgefenster zu kombinieren. Die Target-Zone bezeichnet dabei die Anzahl betrachteter Nachfolger und ist aktuell unbeschränkt. Folglich wird hier eine Größe ermittelt, die einen guten Kompromiss zwischen der Anzahl an Kombinationen und der Genauigkeit des Matchings bildet.
4. **Selection-Method:** Das letzte durchgeführte Experiment dieser Arbeit dient der Ermittlung einer Methode für die Frequenzselektion, die vom UniRef90-Sampling profitiert, aber die Auswahl zusätzlich reduziert. Dafür wird vor Betrachtung der Schwellwerte eine Vorselektion durchgeführt, einmal anhand der lokalen Maxima

der Amplituden, dann anhand der stärksten Abweichungen der Amplituden von ihren Schwellwerten und natürlich ohne Vorselektion als Nullprobe, die nur das Sampling einbezieht. Dabei wird die Datenbankerstellung abgebrochen, sobald deren Größe die Eingabe um ein 6-faches übersteigt.

Als Trainingsdaten wird eine Referenz-Datenbank von Pflanzen-Proteinen verwendet, die in Gruppen derselben Funktion eingeordnet sind. Diese Zuordnung wurde manuell von Experten durchgeführt in sogenannte MapMan-Bins, was heißt:
Derselbe Bin $\rightarrow$ dieselbe Funktion [LOH+14].

2 Material und Methoden

2.1 Grundalgorithmus

Algorithmus 1 Vorbereitung

Vorbedingung $sequence \in \{\text{A-Z}, \text{'}\Psi\text{'}, \text{'}\Omega\text{'}, \text{'}\Phi\text{'}, \text{'}\zeta\text{'}, \text{'}\Pi\text{'}, \text{'+'}, \text{'-'}\}^*$
Vorbedingung $0 \leq kf \leq 10$
Nachbedingung $|aa\_vector| = |sequence|$
Nachbedingung $v \geq 0 \ \forall \ v \in aa\_vector$

```
aa_vector ← array()
for each aa in sequence do
    kf_value ← get_kf_value(aa, kf)
    min_kf_value ← get_min_kf_value()
    aa_vector.append(kf_value + abs(min_kf_value))
end for
```

Voraussetzung für den Algorithmus ist ein numerischer Vektor, so wie es das Spektrum einer Tonspur bei SHAZAM darstellt. Um dies im Kontext von Proteinen zu erreichen, wird in prot-fin auf sogenannte Kidera-Faktoren zurückgegriffen. Diese Faktoren stammen aus einem Forschungsprojekt von Akinori Kidera, welches 1985 publiziert wurde. Inhalt des Projekts war die statistische Faktorenanalyse von 188 physikalischen Eigenschaften der 20 natürlichen Aminosäuren zur Ermittlung von 10 dieser Eigenschaften, durch die die anderen aufgrund hoher Korrelation erklärt werden können [vgl. Kid+85]. In Tabelle 1 sind diese dargestellt.

Tabelle 1: Kidera-Faktoren

Beschreibung	**A**	**C**	**D**	**E**	**F**	**G**	**...**
Helix/bend preference	-1.56	0.12	0.58	-1.45	-0.21	1.46	...
Side-chain size	-1.67	-0.89	-0.22	0.19	0.98	-1.96	...
Extended structure preference	-0.97	0.45	-1.58	-1.61	-0.36	-0.23	...
Hydrophobicity	-0.27	-1.05	0.81	1.17	-1.43	-0.16	...
Double-bend preference	-0.93	-0.71	-0.92	-1.31	0.22	0.1	...
Partial specific volume	-0.78	2.41	0.15	0.4	-0.81	-0.11	...
Flat extended preference	-0.2	1.52	-1.52	0.04	0.67	1.32	...
Occurrence in alpha region	-0.08	-0.69	0.47	0.38	1.1	2.36	...
pK-C	0.21	1.13	0.76	-0.35	1.71	-1.66	...
Surrounding hydrophobicity	-0.48	1.1	0.7	-0.12	-0.44	0.46	...

Folglich kann eine Aminosäuresequenz pro Faktor in einen numerischen Vektor übersetzt werden, wobei ein höherer absoluter Wert für mehr Relevanz des Faktors steht. Hätte man also die Beispielsequenz `EVKEFDGQGCFC`, wäre die Übersetzung für die Hydrophobizität die folgende:

E	V	K	E	F	D	G	Q	G	C	F	C
1.17	-0.4	1.7	1.17	-1.43	0.81	-0.16	1.1	-0.16	-1.05	-1.43	-1.05

Da für die Fourier Transformation negative Werte problematisch sind, wird der Vektor anschließend dahingehend normalisiert, dass das absolute Minimum aller Werte aus Tabelle 1 aufaddiert wird. Das absolute Minimum ist 2.33, weshalb der normalisierte Ergebnisvektor der Beispielsequenz nun so aussieht:

$$\begin{aligned} raw\_vec &= [1.17\ \text{-}0.4\ \ 1.7\ \ 1.17\ \text{-}1.43\ \ 0.81\ \text{-}0.16\ \ 1.1\ \text{-}0.16\ \text{-}1.05\ \text{-}1.43\ \text{-}1.05] \\ normalized\_vec &= raw\_vec + 2.33 \\ normalized\_vec &= [3.5\ \ 1.93\ \ 4.03\ \ 3.5\ \ 0.9\ \ 3.14\ \ 2.17\ \ 3.43\ \ 2.17\ \ 1.28\ \ 0.9\ \ 1.28] \end{aligned} \tag{1}$$

Algorithmus 2 Sammeln von Strukturdaten

Vorbedingung $v \geq 0\ \forall\ v \in aa\_vector$
Nachbedingung *constellation_map* is an Array of Arrays of Floats
  $constellation\_map \leftarrow$ `array`()
  $stft\_result \leftarrow$ `stft_transform`(aa_vector)
  for each *window* **in** $stft\_result$ **do**
    $selected\_frequencies \leftarrow$ `select_maxima`(*window*)
    $constellation\_map$.`append`($selected\_frequencies$)
  end for

Der erhaltene Vektor aus Algorithmus 1 wird nun strukturell analysiert. Dieses Vorgehen basiert auf der Short-Time-Fourier-Transformation (STFT), welche den Vektor fensterweise auf periodische Signale untersucht, wie z.B. dem wiederholten Auftreten von hydrophoben Aminosäuren im gleichen Abstand oder in der Musik ein Refrain oder dem Rhythmus. Für jedes Fenster werden die Frequenzen der auffälligsten Signale ausgewählt, wie in Abbildung 1 dargestellt mittels der lokalen Maxima, sodass über alle Intervalle eine sogenannte Constellation-Map entsteht. Diese Map wird dabei als Array repräsentiert, wobei jedes Element hierbei eine Liste darstellt, die ihrem Index entsprechend die gewählten Frequenzen des jeweiligen Fensters beinhaltet. In Abbildung 2 ist im linken Teil die visuelle Darstellung einer Constellation-Map als Scatter-Plot abgebildet.

Algorithmus 3 Hashing

Vorbedingung *constellation_map* is an Array of Arrays of Floats
Vorbedingung *protein_id* is a String
Vorbedingung $0 \leq kf \leq 10$
Nachbedingung *hashes* is a HashMap of $Int \rightarrow Int, String$

```
hashes ← hashmap()
window_idx ← 0
repeat
    selected_frequencies ← constellation_map.pop(0)
    for each frequency in selected_frequencies do
        successor_idx ← 0
        for each succ_frequencies in constellation_map do
            for each succ_frequency in succ_frequencies do
                hash ← create_hash(frequency, succ_frequency, successor_idx, kf)
                hashes[hash] ← (window_idx, protein_id)
            end for
        end for
        successor_idx ← successor_idx + 1
    end for
    window_idx ← window_idx + 1
until constellation_map is empty
```

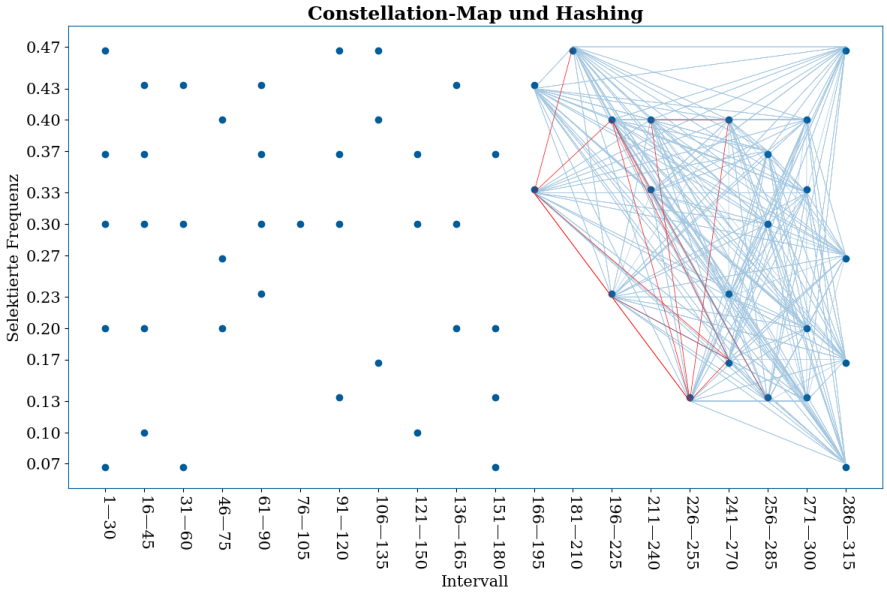

Abbildung 2: Constellation-Map und Hashing: Die Punkte bilden die Constellation-Map. Die zur Übersicht nur rechts eingezeichneten Kanten repräsentieren die Hash-Bildung, wobei rote Kanten ignorierte Hashes darstellen, weil sie mehrfach auftauchen.

Die erhaltene Map wird nun elementweise gehashed, um einen effizienten Vergleich mit anderen Maps zu ermöglichen. Um das zu erzielen wird jede ausgewählte Frequenz eines Fensters mit jeder weiteren Frequenz der Nachfolgefenster gepaart. Bildlich gesprochen werden also Kanten gebildet, wodurch die Map zu einem Graphen wird. Jede dieser Kanten bildet nun einen Hash, also einer Kombination aus den beiden Frequenzen/-Kantenenden und der Kantenlänge (hier `successor_idx`). In einer Hashmap wird sich folgend für den Hash die Position der Kante in der Constellation-Map gemerkt, sowie die ID des Proteins, für die diese Map erstellt wurde. Sollte ein Hash mehrfach vorkommen, verbleibt lediglich seine letzte Position. Dieses Verfahren wird im rechten Teil von Abbildung 2 repräsentativ dargestellt, wobei rote Kanten die ignorierten Kanten abbilden.

Single-Protein-Matching: Nachdem eine Datenbank mit den Hashes verschiedener Trainings-Proteine (TP) trainiert wurde, wird sie verwendet, um Proteine zu erkennen. Dazu werden für die jeweilige Eingabesequenz von Aminosäuren die Hashes gebildet und deren Positionen gespeichert. Um nun die Ähnlichkeit der Constellation-Map der Eingabe mit denen der TP zu bestimmen, werden pro Eingabe-Hash die Differenzen zwischen dessen Position mit den Positionen der trainierten Hashes gebildet und global pro Protein gezählt. Diese Differenzen repräsentieren den Abstand der Kante in der Eingabe-Map zur Kante der jeweiligen TP-Map, also wie weit die Eingabe-Map verschoben wäre, sollte es sich bei dem TP um das Original handeln. Auf diese Weise sammeln sich pro TP mehrere solcher potentiellen Abstände, wobei nun der Abstand, der am häufigsten aufgetreten ist, offensichtlich die meiste Übereinstimmung in den Kanten zeigt. Diese Tatsache qualifiziert diese Maximalanzahl als geeigneten Score (S1) für ein Match.

Da es große Proteine mit sehr langen Aminosäuresequenzen kürzere Sequenzen kleinerer funktionsungleicher Proteine enthalten können, reicht der ermittelte Score alleine nicht aus, da in diesem Fall sehr viele Kanten der Eingabe-Map übereinstimmen würden, sodass trotz Mis-Match der nahezu maximale Score erreicht werden würde.

Um das zu umgehen, wird der Jaccard-Similarity-Index (JSI) verwendet, einem Maß, das die Übereinstimmung zweier Mengen A und B wie folgt bewertet:

$$JSI(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Dieser Index nimmt einen Wert von 0 an, wenn beide Mengen disjunkt sind, und nähert sich der 1 je größer die Schnittmenge ist. Im Fall des Vergleichs zweier Constellation-Maps, also zwei Hash-Mengen, wird hier bewertet, wie viele Kanten sich die beiden Maps positionsunabhängig teilen. Durch diese Unabhängigkeit reicht der JSI alleine nicht als Score aus, sodass nur in Kombination/Multiplikation mit dem S1 ein robuster Score entsteht, da beide zusammen ihre Schwächen aufheben.

Family-Matching: Wenn prot-fin wie erwartet funktioniert, sollten beim Single-Protein-Matching die Matches mit den besten Scores funktionsähnliche Proteine sein. Ein weiterer Ansatz, solche Verwandten zu ermitteln, ist das Matching mit Proteinfamilien. Hierbei wird als Eingabe eine Tabelle akzeptiert, die lediglich eine Zuordnung von Protein-ID

und Familie enthält. Um nun Matches zu finden, werden von allen Hashes einer Familie nur die behalten, die in allen Mitgliedern vorkommen. Anschließend wird die Datenbank nach Proteinen durchsucht, welche ebenfalls diese Hashes enthalten, wobei als Score diesmal nur die Anzahl infrage kommt, wie viele der Hashes enthalten sind. Die Idee hinter dieser Methode ist, dass Proteine derselben Proteinfamilie, also mit ähnlichen Funktionen, möglicherweise familienspezifische Kanten in der Constellation-Map haben.

2.2 Experiment 1: UniRef90 Sampling

Ein wichtiger Bestandteil des Algorithmus ist die Selektion signifikanter Frequenzen zur Erstellung der Constellation-Map. Es wäre möglich, einfach alle Frequenzen auszuwählen und die Signalstärke in den Hash einfließen zu lassen. Problem hierbei ist aber, dass diese Vorgehensweise zu wesentlich mehr Hashes und einer folglich sehr großen Datenbank führt, was wiederum das Scoring/Matching verlangsamt. Ein Anspruch an prot-fin ist, dass die Datenbankgröße die Eingabegröße nicht wesentlich übersteigt, wobei es sich bei der Eingabe um eine einfache FASTA-Datei handelt.

Diesem Problem soll durch ein Sampling-Experiment abgeholfen werden. Darin werden aus etwa 180 Millionen Sequenzen je ein zufälliges Intervall für die STFT ausgewählt, transformiert und die Signalstärken je Frequenz gemerkt. Um nun daraus eine Selektionsmethode abzuleiten, werden die Grenzquantile einer jeden Frequenz ermittelt, um signifikant seltene Signalstärken zu ermitteln. Folglich ist es möglich, für die Constellation-Map nur diejenigen Frequenzen zu behalten, welche in den Randzonen der Signalstärken liegen, sodass nicht nur Signale infrage kommen, die für eine besonders starke Ausprägung eines Kidera-Faktors sprechen, sondern auch für den Fall der umgekehrten Ausprägung, wie z.B. Hydrophilie statt Hydrophobie.

Der Algorithmus wird daher insofern angepasst, dass bei der Frequenz-Selektion von den Maxima der Signalstärken nur die behalten werden, die die Grenzwerte über-/unterschreiten. Zudem wird beim Hashing je Frequenz noch die Information hinzugefügt, ob sie besonders stark oder schwach ist.

3 Ergebnisse

...

4 Diskussion

...

Eigenständigkeitserklärung

Ich bestätige, dass die eingereichte Arbeit eine Originalarbeit ist und von mir ohne weitere Hilfe verfasst wurde. Die Arbeit wurde nicht geprüft, noch wurde sie widerrechtlich veröffentlicht. Die eingereichte elektronische Version ist die einzige eingereichte Version.

Unterschrift	Ort und Datum

Erklärung zu Eigentum und Urheberrecht

Ich erkläre hiermit mein Einverständnis, dass die Technische Hochschule Bingen diese Arbeit Studierenden und interessierten Dritten zur Einsichtnahme zur Verfügung stellen und unter Nennung meines Namens (Franz-Eric Sill) veröffentlichen darf.

Unterschrift	Ort und Datum