Algorithmen der Sequenzanalyse: Motivanalysen, Sequenzalignment

AlgSeq – 18/11/2024

Prof. M. Sammeth

Motivanalysen

Beispiel, gegeben:

Dna= ttaccttaac
gatgtctgtc
acggcgttag
ccctaacgag
cgtcagaggt

PROFIL=

A:	4/5	0	0	1/5
C:	0	3/5	1/5	0
G:	1/5	1/5	4/5	0
T:	0	1/5	0	4/5

Profil-wahrscheinlichte 4-mere:

MOTIVE(*Profil, Dna*) = tt**acct**taac
g**atgt**ctgtc
acg**gcgt**tag
ccct**aacga**g
cgtcag**aggt**

PROFIL(*Motive*)=

A:	4/5	0	0	1/5
C:	0	3/5	1/5	0
G:	1/5	1/5	4/5	0
T:	0	1/5	0	4/5

usw.

PROFIL(MOTIVE(*Profil, Dna*)) → MOTIVE(PROFIL(MOTIVE(*Profil, Dna*))) →

PROFIL(MOTIVE(PROFIL(MOTIVE(*Profil, Dna*))))) → ...

Randomisierte Motiv-Suche

Algorithm: RANDOMISIERTEMOTIVSUCHE(Dna, k,t)

wähle zufällige k-mere $Motive = (Motiv_1, \ldots, Motiv_t)$ von je einer Sequenz in Dna

$BesteMotive \leftarrow Motive$

while für immer **do**

  $Profile \leftarrow$ PROFILE($Motive$)

  $Motive \leftarrow$ MOTIVE($Profile, Dna$)

  if SCORE($Motive$) $<$ SCORE($BesteMotive$) **then**

    $BesteMotive \leftarrow Motive$

  ’**else**

    return $BesteMotive$

Beispiel, ein implantiertes
(4,1)-Motiv **ACGT**:

Dna=

```
ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT
```

Motive=

```
taac
GTct
ccgG
acta
AGGT
```

PROFIL(*Motive*)=

```
A: 0.4 0.2 0.2 0.2
C: 0.2 0.4 0.2 0.2
G: 0.2 0.2 0.4 0.2
T: 0.2 0.2 0.2 0.4
```

Wahrscheinlichkeiten *Pr*(*kmer*, PROFIL(*Motive*)):

ttAC	tACC	ACCT	CCTt	CTta	Ttaa	taac
.0016	.0016	.0128	.0064	.0016	.0016	.0016
gATG	ATGT	TGTc	GTct	Tctg	ctgt	tgtc
.0016	.0128	.0016	.0032	.0032	.0032	.0016
ccgG	cgGC	gGCG	GCGT	CGTt	GTta	Ttag
.0064	.0032	.0016	.0128	.0032	.0016	.0016
cact	acta	ctaA	taAC	aACG	ACGA	CGAg
.0032	.0064	.0016	.0016	.0032	.0128	.0016
cgtc	gtca	tcag	cagA	agAG	gAGG	AGGT
.0016	.0016	.0016	.0032	.0032	.0032	.0128

MOTIVE(PROFIL(*Motive*))

```
ACCT
ATGT
GCGT
ACGA
AGGT
```

Evaluierung der Randomisierten Motiv-Suche

implantiertes (4,1)-Motiv

```
ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT
```

eher unwahrscheilich

alle Motive entsprechen den tatsächlich implementierten

```
ACCT
ATGT
GCGT
ACGA
AGGT
```

```
A: 0.8 0.0 0.0 0.2
C: 0.0 0.6 0.2 0.0
G: 0.2 0.2 0.8 0.0
T: 0.0 0.2 0.0 0.8
```

nicht so unwahrscheilich

ein Motiv entspricht dem tatsächlich implementierten

```
taac
GTct
ccgG
acta
AGGT
```

```
A: 0.4 0.2 0.2 0.2
C: 0.2 0.4 0.2 0.2
G: 0.2 0.2 0.4 0.2
T: 0.2 0.2 0.2 0.4
```

randomisierte Sequenzen (ohne implantiertes Motiv)

randomisierte k-mere

Matrize uninformativ (random)

```
A: 0.25 0.25 0.25 0.25
C: 0.25 0.25 0.25 0.25
G: 0.25 0.25 0.25 0.25
T: 0.25 0.25 0.25 0.25
```

Gibbs Sampling

- beginnt auch mit zufällig gewählten *k*-meren,

ist aber vorsichtiger als Randomisierte Motiv-Suche:

→ in jeder "Runde" wird lediglich ein *k*-mer betrachtet und entschieden, ob es behalten oder ersetzt werden soll.

```
ttaccttaac
gatatctgtc
acggcgttcg
ccctaaagag
cgtcagaggt
```

→

```
ttaccttaac
gatatctgtc
acggcgttcg
ccctaaagag
cgtcagaggt
```

```
ttaccttaac
gatatctgtc
acggcgttcg
ccctaaagag
cgtcagaggt
```

→

```
ttaccttaac
gatatctgtc
acggcgttcg
ccctaaagag
cgtcagaggt
```

RANDOMISIERTEMOTIVSUCHE

(kann alle *k*-mere in einer Runde austauschen)

GIBBSSAMPLER

(tauscht höchstens ein *k*-mer pro Runde)

Algorithm: GibbsSampler(Dna, k,t,N)

wähle zufällige k-mere $Motive = (Motiv_1, \ldots, Motiv_t)$ von je einer Sequenz in Dna

$BesteMotive \leftarrow Motive$

for $j \leftarrow 1$ bis N **do**

  $i \leftarrow$ Random(t)

  $Profile \leftarrow$ Profil-Matrix berechnet aus allen k-meren in $Motive$ außer $Motiv_i$

  $Motiv_i \leftarrow$ Profil-zufällig gewähltes k-mer in der i-ten Sequenz von Dna

  if Score($Motive$) $<$ Score($BesteMotive$) **then**

    $BesteMotive \leftarrow Motive$

implantiertes (4,1)-Motiv

```
ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT
```

zufällig gewählte Motive

```
ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT
```

zufällig 3. Sequenz gewählt

```
ttACCTtaac
gATGTctgtc
----------
cactaACGAg
cgtcagAGGT
```

Motive

```
taac
GTct
acta
AGGT
```

ZAEHLEN(*Motive*)=

A:	2	1	1	1
C:	0	1	1	1
G:	1	1	1	0
T:	1	1	1	2

PROFIL(*Motive*)=

A:	2/4	1/4	1/4	1/4
C:	0	1/4	1/4	1/4
G:	1/4	1/4	1/4	0
T:	1/4	1/4	1/4	2/4

3. Sequenz= ccgGCGTtag

ccgG	cgGC	gGCG	GCGT	CGTt	GTta	Ttag
0	0	0	1/128	0	1/256	**0**

→ kein (faires) gewichtetes Subsampling möglich wo Pr()= 0 !

Motive **taac**
GTct
acta
AGGT

PSEUDOCOUNT(*Motive*)=

A:	3	2	2	2
C:	1	2	2	2
G:	2	2	2	1
T:	2	2	2	3

PROFIL(*Motive*)=

A:	3/4	2/8	2/8	2/8
C:	1/8	2/8	2/8	2/8
G:	2/8	2/8	2/8	1/8
T:	2/8	2/8	2/8	3/8

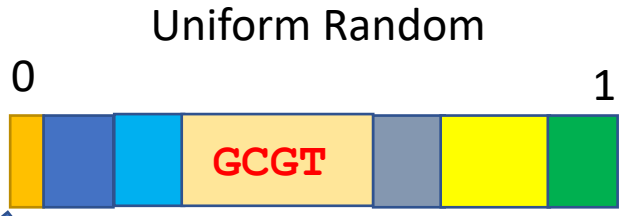
3. Sequenz= `ccgGCGTtag`

ccgG	cgGC	gGCG	GCGT	CGTt	GTta	Ttag
$4/8^4$	$8/8^4$	$8/8^4$	$\mathbf{24/8^4}$	$12/8^4$	$16/8^4$	$8/8^4$

gewichtetes Subsampling:

$$Random\left(\frac{4/8^4}{80/8^4}, \frac{8/8^4}{80/8^4}, \frac{8/8^4}{80/8^4}, \frac{24/8^4}{80/8^4}, \frac{12/8^4}{80/8^4}, \frac{16/8^4}{80/8^4}, \frac{8/8^4}{80/8^4}\right)$$

$$= Random\left(\frac{4}{80}, \frac{8}{80}, \frac{8}{80}, \frac{24}{80}, \frac{12}{80}, \frac{16}{80}, \frac{8}{80}\right).$$



Annahme: **GCGT** wurde gewichtet zufällig in der 3. Sequenz gewählt

Motive

taac	ttACCT**taac**
GTct	gAT**GTct**gtc
GCGT	ccg**GCGT**tag
acta	c**acta**ACGAg
AGGT	cgtcag**AGGT**

Neues Motiv in 3. Sequenz

```
----------
gAT GTct gtc
ccg GCGT tag
c acta ACGAg
cgtcag AGGT
```

1. Sequenz zufällig gewählt

GTct
GCGT
acta
AGGT

Motive

A:	3/8	1/8	1/8	2/8
C:	1/8	3/8	2/8	1/8
G:	3/8	2/8	3/8	1/8
T:	1/8	2/8	2/8	4/8

PSEUDOCOUNTS, PROFIL(*Motive*)

1. Sequenz= ttACCTtaac

ttAC	tACC	ACCT	CCTt	CTta	Ttaa	taac
$2/8^4$	$2/8^4$	$\mathbf{72/8^4}$	$24/8^4$	$8/8^4$	$4/8^4$	$1/8^4$

Annahme: **ACCT** wurde gewichtet zufällig in der 1. Sequenz gewählt

Annahme: **ACCT** wurde gewichtet zufällig in der 1. Sequenz gewählt

Motive

ACCT
GTct
GCGT
acta
AGGT

ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT

Neues Motiv in 1. Sequenz

ttACCTtaac
gATGTctgtc
ccgGCGTtag
----
cgtcagAGGT

4. Sequenz zufällig gewählt

ACCT
GTct
GCGT
AGGT

Motive

A:	3/8	1/8	1/8	1/8
C:	1/8	3/8	3/8	1/8
G:	3/8	2/8	3/8	1/8
T:	1/8	2/8	1/8	5/8

PSEUDOCOUNTS, PROFIL(*Motive*)

4. Sequenz= cactaACGAg

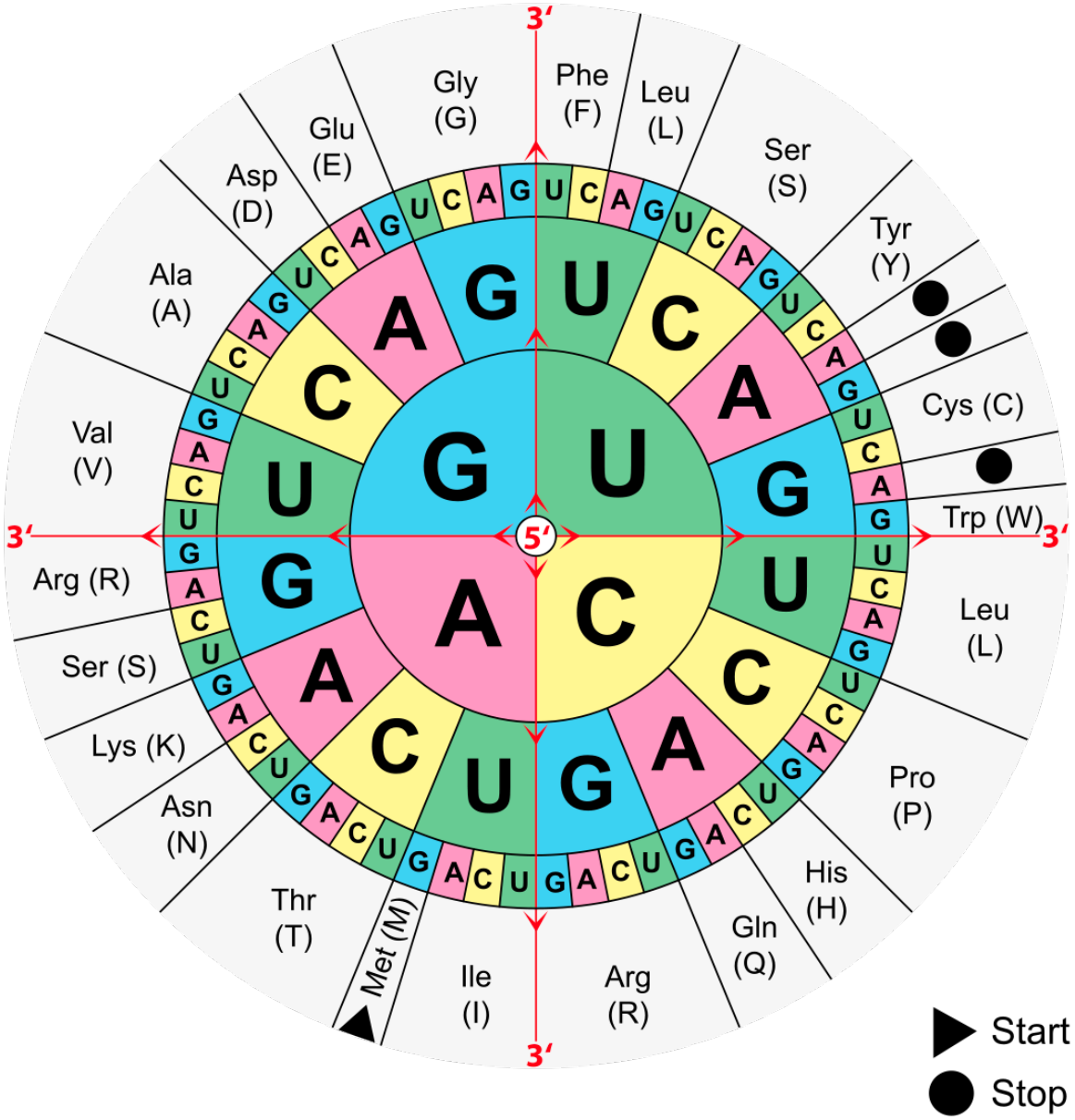
cact	acta	ctaA	taAC	aACG	ACGA	CGAg
$15/8^4$	$\mathbf{9/8^4}$	$2/8^4$	$1/8^4$	$9/8^4$	$\mathbf{27/8^4}$	$2/8^4$

Annahme: **ACGA** wurde gewichtet zufällig in der 4. Sequenz gewählt

ttACCTtaac
gATGTctgtc
ccgGCGTtag
cactaACGAg
cgtcagAGGT

Sequenz-Alignments

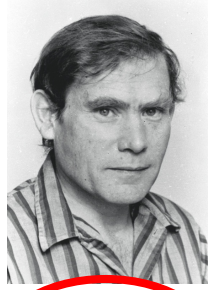
Knacken des Genetischen Codes



1961 – Entdeckung der Triplet-Natur von Codons

Francis Crick

Leslie Orgel



Sidney Brenner

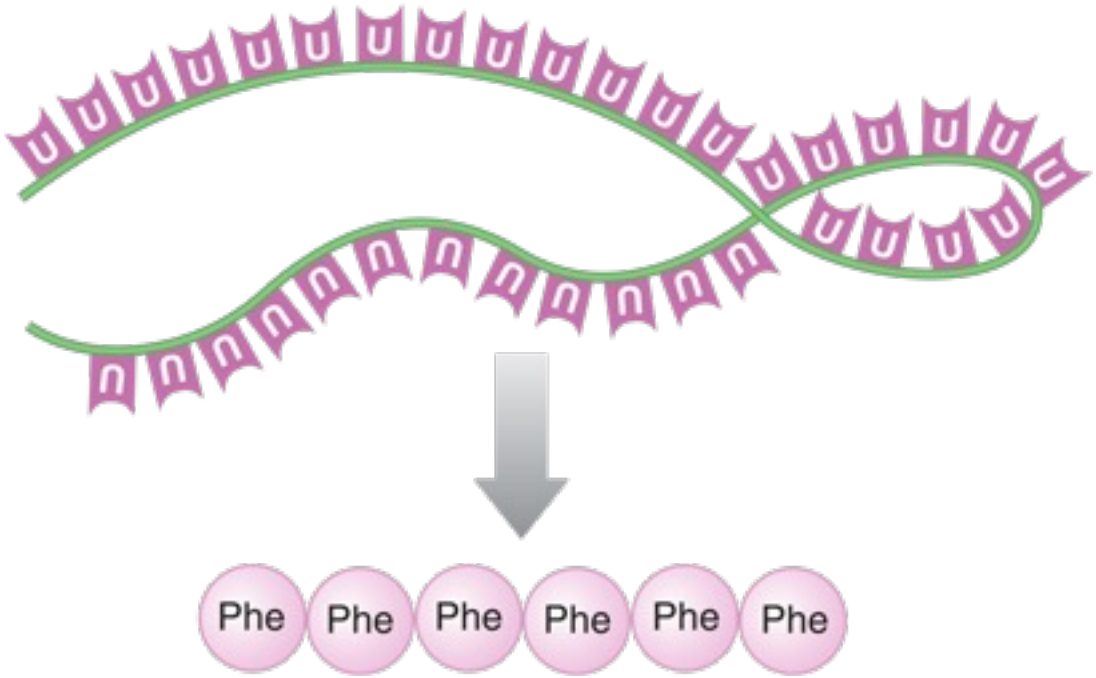
Alexander Rich James Watson

		MUTATION	PHENOTYPE
Wild-type sequence	A B C A B C A B C A B C A B C A B C A B C A B C	NONE	rII+
FC0 mutant	A B C A A B C A B C A B C A B C A B C A B C A B	+	rII−
Supression of FC0	A B C A A B A B C A B C A B C A B C A B C A B C	+ −	rII+
Two base additions	A B C A A B C A B C B A B C A B C A B C A B C A	+ +	rII−
Three base additions	A B C A A B C A B C B A B C C A B C A B C A B C	+ + +	rII+

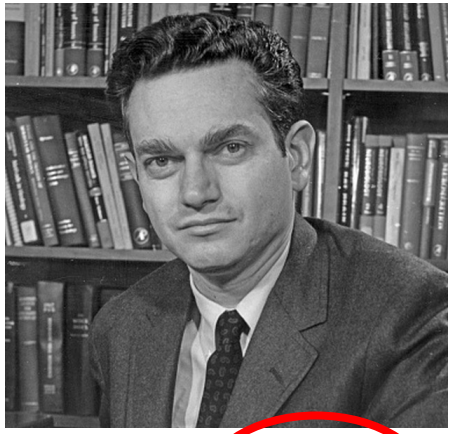
+ Base addition
− Base deletion

Experimente mit Phagen, Insertion von 3 Mutationen restauriert den Wildtyp!

1961 – erstes Codon UUU



in vitro Translationssystem für poly-*U* RNAs



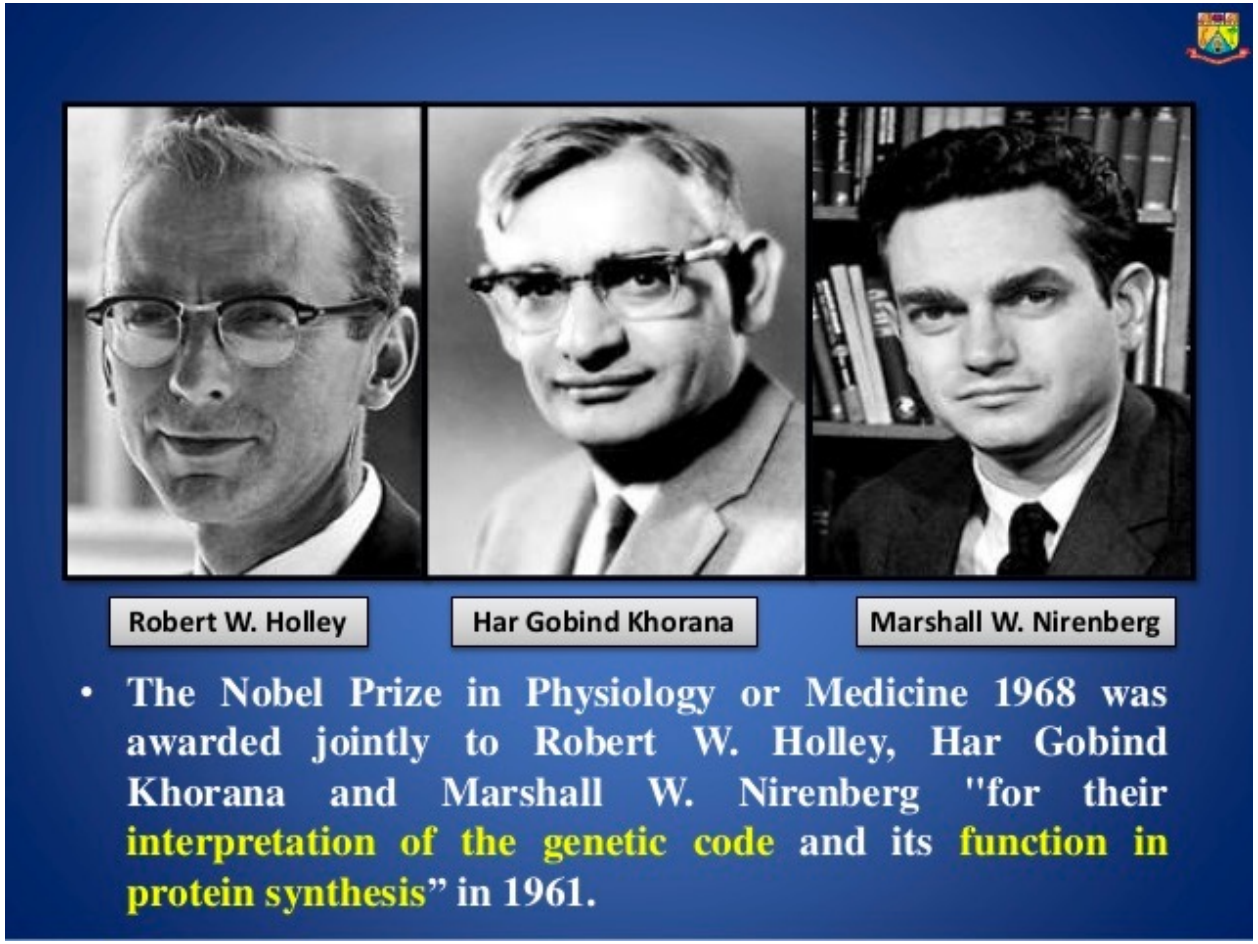
Marshall Nirenberg

1960s – weitere Experimente mit Homopolymeren

Har Gobind

z.B. UCUCUCUCUCUC…

➔LeuSerLeuSer

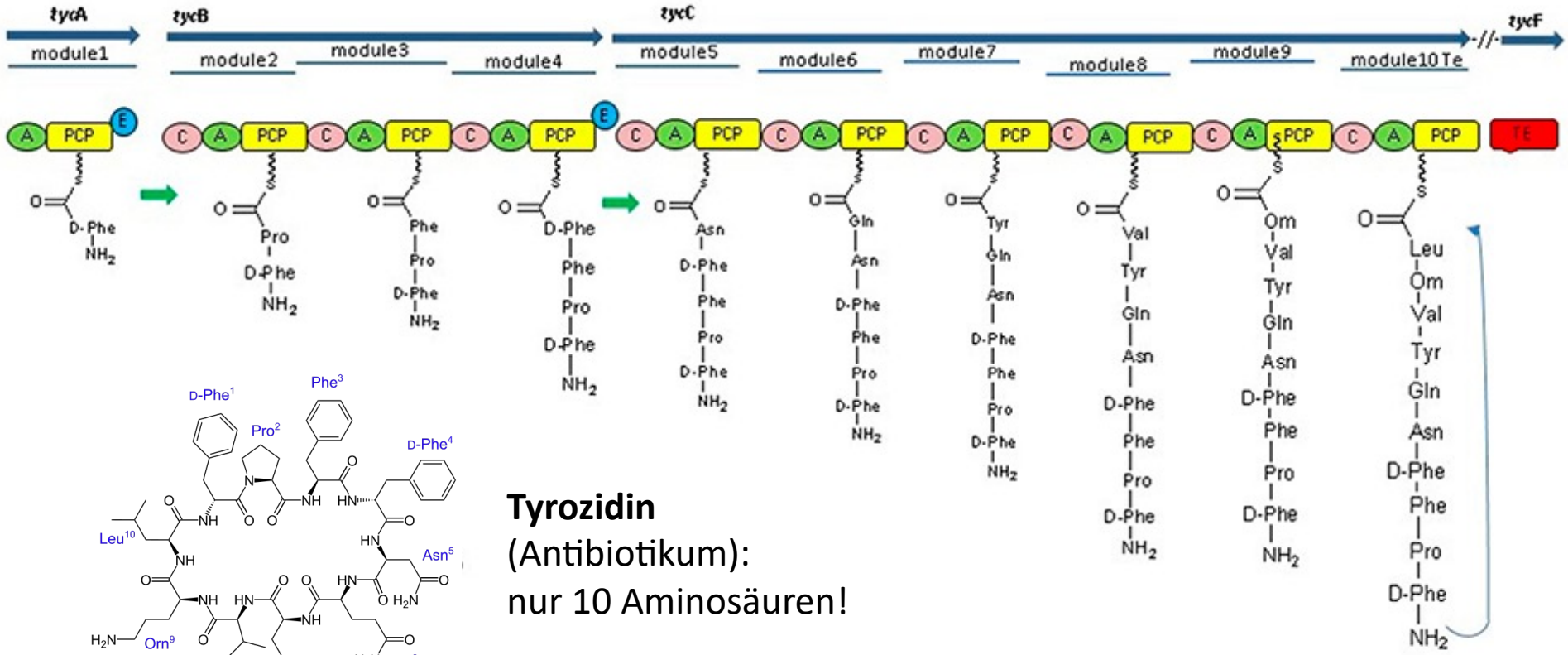


Der *Nicht*-Ribosomale Code

Bakterien und Pilze kodieren Antibiotika und andere *nicht-ribosomale Peptide* (***NRP***)
Enzym: ***NRP Synthetase*** (unabhängig von Riobosomen und genet. Code!)

DNA → RNA → **NRP Synthetase** → NRP

NRP Synthetasen bestehen aus Modulen, sog. Adenylierungs-Domänen (A-Domänen), wovon jede nur eine Aminsäure synthetisiert.
z.B. **NRP** Synthetase für **Tyrozidin**: 10 *A-Domänen* mit je 500 ASren → 5.000 AS Protein!



Tyrozidin
(Antibiotikum):
nur 10 Aminosäuren!

Knacken des Nicht-Ribosomalen Codes

BIOENGINEERING

Working Outside the Protein-Synthesis Rules

Proteins built in the ribosome are subject to certain restrictions, so researchers are harnessing a nonribosomal system that might one day make new drugs

Mohamed Marahiel

Sequenzen von drei *A-Domänen* bekannt

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTEFINHYGPTEATIGA
Orn:
AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYIYEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSAPTMISSLEILFAAGDRLSSQDAILARRAVGSGVYNAYGPTENTVLS

(jeweils nur ca. 110 der ca. 500 Aminosäuren einer A-Domäne gezeigt)

→ was sind die Gemeinsamkeiten und die (spezifischen) Unterschiede dieser 3 A-Domänen?

Beobachtungen zum Sequenzvergleich

1. Versuch: Direkter Vergleich der iten Aminosäure → 3 konservierte Spalten

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTEFINHYGPTEATIGA
Orn:
AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYIYEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSAPTMISSLEILFAAGDRLSSQDAILARRAVGSGVYNAYGPTENTVLS

2. Versuch: Sequenzähnlichkeiten "mit dem Auge" finden

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTEFINHYGPTEATIGA
Orn:
AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYIYEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSAPTMISSLEILFAAGDRLSSQDAILARRAVGSGVYNAYGPTENTVLS

→ zweite Sequenz "nach rechts" verschieben, durch Einfügen eines Platzhalters "–" am Beginn

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTEFINHYGPTEATIGA
Orn:
-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYIYEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSAPTMISSLEILFAAGDRLSSQDAILARRAVGSGVYNAYGPTENTVLS

→ 11 konservierte Spalten

Verschieben durch Einfügen von Platzhaltern (Alignieren)

3. weitere fünf (4+1) Platzhalter einfügen

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTEFINHYGPTEATIGA
Orn:
-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYI-YEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSA----PTMISSLEILFAAGDRLSSQDAILARRAVGSGVYNAYGPTENTVLS

→ 14 konservierte Spalten

4. und noch weitere drei Platzhalter einfügen

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTE-FINHYGPTEATIGA
Orn:
-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYI-YEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSA----PTMISSLEILFAAGDRLSSQDAILARRAVGSGV-Y-NAYGPTENTVLS

→ 19 konservierte Spalten (inkl. 2 Duplets)

Alignments zeigen konservierte/variable Bereiche

Das Alignment zeigt den konservierten Kern (Core) der A-Domänen:

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTE-FINHYGPTEATIGA
Orn:
-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYI-YEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSA----PTMISSLEILFAAGDRLSSQDAILARRAVGSGV-Y-NAYGPTENTVLS

→ dadurch konnte Marahiel die mit der kodierten Aminsäure variierende *Signatur* finden:

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTE-FINHYGPTEATIGA
Orn:
-AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYI-YEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSA----PTMISSLEILFAAGDRLSSQDAILARRAVGSGV-Y-NAYGPTENTVLS

Signaturen:

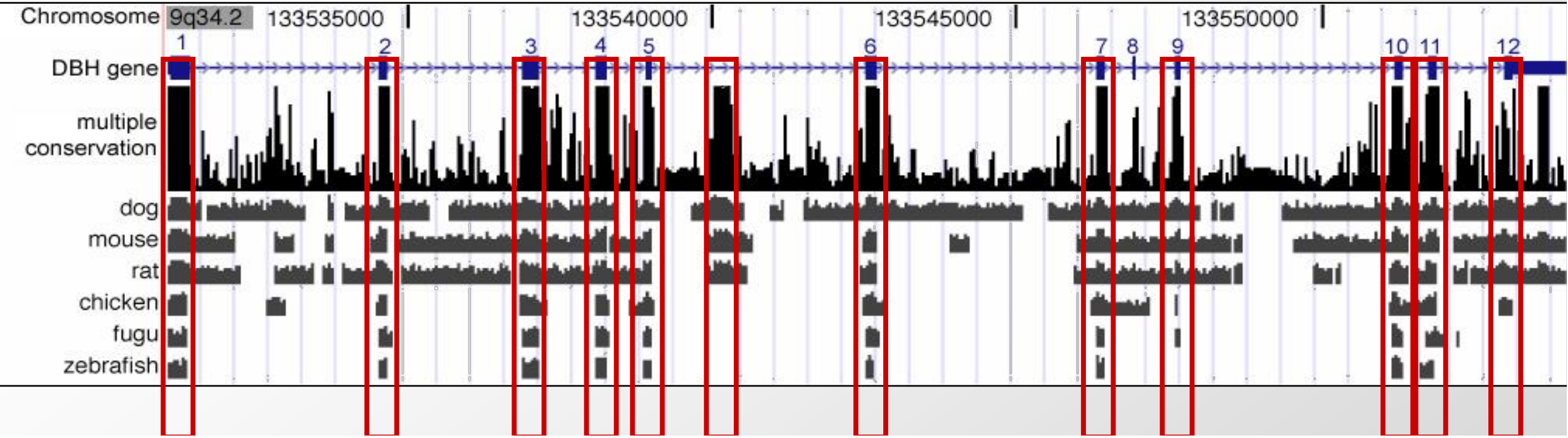
LTKVGHIG	→ Asp
VGEIGSID	→ Orn
AWMFAAVL	→ Val

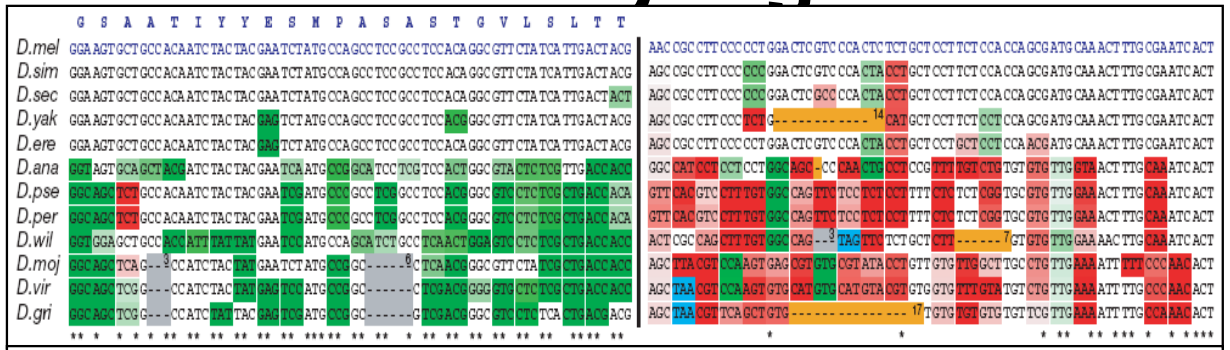
→ das wäre im ursprünglichen Alignment unmöglich gewesen:

Asp:
YAFDLGYTCMFPVLLGGGELHIVQKETYTAPDEIAHYIKEHGITYIKLTPSLFHTIVNTASFAFDANFESLRLIVLGGEKIIPIDVIAFRKMYGHTEFINHYGPTEATIGA
Orn:
AFDVSAGDFARALLTGGQLIVCPNEVKMDPASLYAIIKKYDITIFEATPALVIPLMEYIYEQKLDISQLQILIVGSDSCSMEDFKTLVSRFGSTIRIVNSYGVTEACIDS
Val:
IAFDASSWEIYAPLLNGGTVVCIDYYTTIDIKALEAVFKQHHIRGAMLPPALLKQCLVSAPTMISSLEILFAAGDRLSSQDAILARRAVGSGVYNAYGPTENTVLS

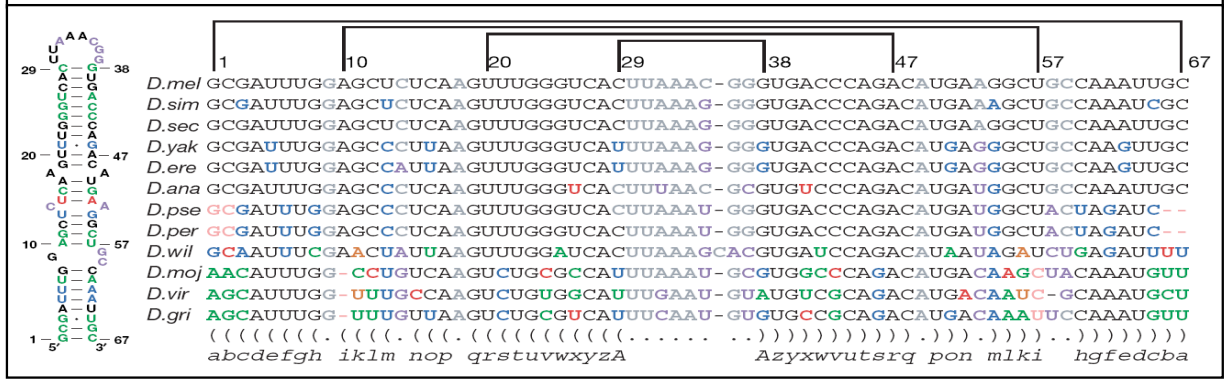
Weitere Anwendungen von Alignments:

Konservierung (phylogenet. Kontext)

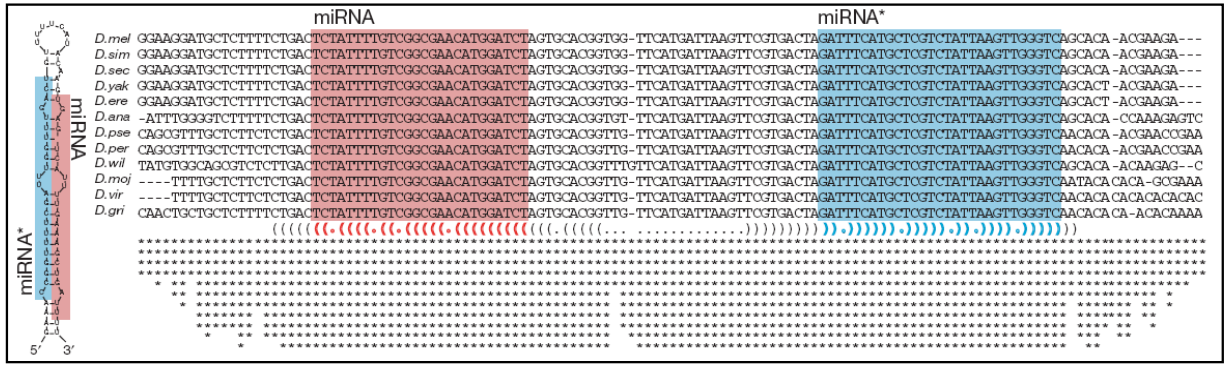




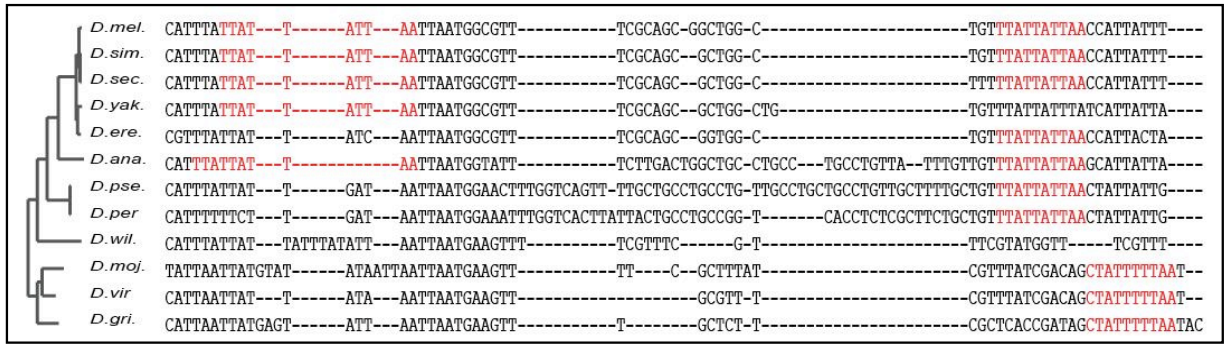
(Protein-)kodierende Gene



RNA Strukturen



miRNAs



Regulatorische Motive

Motivation von Alignments

Vereinfachung: paarweiser Vergleich

v= ATGCATGC
w= TGCATGCA

Beobachtungen:
- nicht alle äquivalenten Sequenzen (*v,w*) sind gleich lang
- auch bei gleichlangen Sequenzen $|v| = |w|$ entspricht das *i*-te Symbol des einen String *v* manchmal einer (völlig) anderen Position im anderen String *w*.

Alignment:
Einfügen von Platzhalter-Symbolen (”gaps”) um eine Matrix zu erhalten, in der gleich(-wertig)e Symbole in der gleichen Spalte $i < |v| + |w|$ stehen.
→ (Hamming-)Distanz minimiert.

z.B.

ATGCATGC → ATGCATGC-
TGCATGCA → -TGCATGCA

HammingDistanz = 8 → HammingDistanz = 2

ATGCTTA → ATGC-TTA-
TGCATTAA → -TGCATTAA

HammingDistanz = 5 → HammingDistanz = 3

Die Alignment Matrix zeigt In/Dels und Mis/Matches

Das Alignment zweier Sequenzen *v* und *w* ist eine ***Matrix***, so dass die *1. Zeile* die Symbole von *v* und die *2. Zeile* die Symbole von *w* (in Reihenfolge, von li nach re) beinhaltet.

v = A T G T T A T A

w = A T C G T C C

v	A	T	–	G	T	T	A	T	A
w	A	T	C	G	T	–	C	–	C

- Spalten mit gleichen Buchstaben heissen Matches und repräsentieren konservierte Nukleot.
- Spalten mit unterschiedlichen Buchstaben heissen Mismatches (= Substitutionen)
- Spalten mit einem Platzhalter "–" werden Indels genannt:
  - "–" in der 1. Zeile ist eine Insertion, repräsentiert Einfügen von *v* nach *w*
  - "–" in der 2. Zeile ist eine Deletion, repräsentiert Entfernen von *v* nach *w*
  - es gibt keine Spalte mit zwei Platzhaltern "–"!

➔ das Alignment zwischen *v* und *w* von oben zeigt: vier Matches, zwei Mismatches, eine Insertion und zwei Deletionen.

Alignment als ein Spiel

Zeichen in Strings (verbleibend)	Alignment (wachsend)	P.
A T G T T A T A A T C G T C C		
T G T T A T A T C G T C	A A	+1
G T T A T A C G T C C	A T A T	+1
G T T A T A G T C C	A T – A T C	
T T A T A T C C	A T – G A T C G	+1
T A T A C C	A T – G T A T C G T	+1
A T A C C	A T – G T T A T C G T –	
T A C	A T – G T T A A T C G T – C	
A C	A T – G T T A T A T C G T – C –	
	A T – G T T A T A A T C G T – C – C	

Regeln:

- in jeder Runde nehmen wir das *erste* (linke) Symbol *von einer oder beiden* Strings und fügen es hinten (rechts) ans Alignment
- wenn wir *zwei gleiche Symbole* ans Alignment fügen, bekommen wir +1 Punkt (Greedy!)
- wenn wir nur ein Sybol ans Alignment fügen, wird ein *Platzhalter* "–" am anderen String angefügt

➔ Das Alignieren von ATGTTATA und ATCGTCC gibt +4 Punkte insgesamt!

Und wie gewinnt man das Spiel?

- Nicht-Ribosomaler Code: "gutes" Alignment zeigt möglichst viele Spalten mit gleichen Symbolen (Matches);
- die Matches im Alignment definieren eine Sequenz von Symbolen, die so in der gleichen Reihenfolge in beiden Strings *v* und *w* vorkommen: das ist die ***"gemeinsame Subsequenz"***;

v = A T G T T A T A

w = A T C G T C C

```
v  A  T  -  G  T  T  A  T  A
w  A  T  C  G  T  -  C  -  C
```

- ein "gutes" Alignment findet eine lange bzw. die ***längste gemeinsame Subsequenz*** der algnierten Strings.

Längste Gemeinsame Subsequenz Problem: *Finde eine längste gemeinsame Subsequenz von zwei Strings.*

Input: Zwei Strings, *v* und *w*.

Output: Eine gemeinsame Subsequenz von *v* und *w* mit maximaler Länge.

Wie viele Alignments gibt es?

v= A T G T T A T A
w= A T C G T C C

Abschätzung nach Alignment-Spiel:

- höchstens 3 Möglichkeiten in jeder Runde
- höhstens $|v| + |w|$ Runden

$\in \boldsymbol{O}(3^{|v| + |w|})$ oder $\boldsymbol{O}(3^{2L})$ mit $L = |v| + |w|$

tatsächlich:

$$\frac{(|v|+|w|)!}{|v|!|w|!} \quad \text{bzw.} \quad \frac{2L!}{L!^2} \quad \text{mit } L = |v| + |w|$$

$$\sim \frac{2^{2L}}{\sqrt{\pi L}}$$

Stirlingformel:

mit $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

exponentiell viele, mögliche Alignments!
Brute Force nur mit (sehr) kurzen Sequenzen möglich

Übungen

1. Implementieren Sie den Algorithmus RANDOMISIERTEMOTIVSUCHE und wenden Sie diesen dann auf das Subtile Motiv Problem an. Wie groß ist das beste, und wie groß der Median von SCORE(Moitve) nach 20, 200 bzw. nach 2000 Durchläufen?

   Eingabe: Algo04_subtiles_motiv.txt, *k*= 15

2. Vergleichen Ihre die Ergebnisse aus (1) mit entsprechenden Werten nach 20, 200 und 200 Läufen von GIBBSSAMPLER, mit je N= 200 Iterationen.

   Eingabe: Algo04_subtiles_motiv.txt, *k*= 15

3. Implementieren Sie einen Greedy-Algorithmus nad dem im Seminar besprochenen Alignment-“Spiel”, mit 1 Punkt für Matches und 0 Punkten für Mismatches bzw. Gaps.

   Eingabe: `ATGCATGC , TGCATGCA`

4. Vergkeichen Sie die Ergebnisse aus (3) zu denen einer exhaustiven Suche nach dem besten Alignment zweier kurzer Sequenzen.

   Eingabe: `ATGCATGC, TGCATGCA`
   `Ausgabe: ATGC-TTA-, -TGCATTAA`