

① $2891 = 101101001001$
 111101010111

 $1 \quad 1111 \Rightarrow \text{Hamming Distanz} = 5$

② Der Genotyp hat die Länge $n=46$, also 45 mögliche Crossover spaltungen. Angenommen ein Crossover findet zu 100% statt, ergeben sich folgende Stellen inmitten der funktional zusammengehörenden Gruppe:

1010010101000101101101010100101011001011110111
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
 32

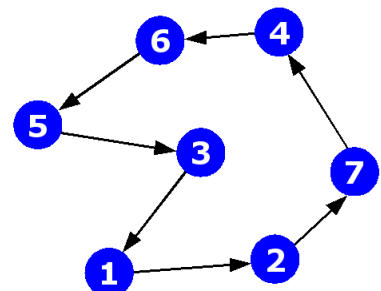
Das bedeutet, wenn an 32 der 45 möglichen crossover stellen gespalten wird, ist die Gruppe aufgebrochen.

$\frac{32}{45} = 0,711 = 71,11\%$

③ Pfadcodierung $P = (1274653)$

Kantencodierung $K = (2716354)$

Originalcodierung $O = (1152321)$



Start	Werte	Ans/codierung
1	1→2→3→4→5→6→7	1
2	2→3→4→5→6→7	1
7	3→4→5→6→7	5
4	3→4→5→6	2
6	3→5→6	3
5	3→5	2
3	3	1

④ 1. Absolute Gitterkoordinaten

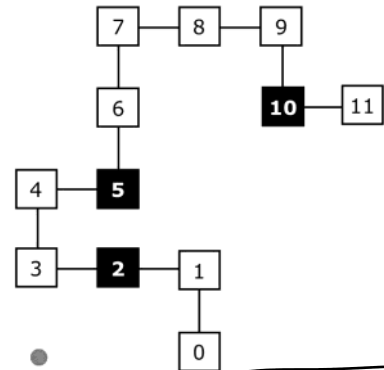
$P_0 (0/0)$	$P_3 (-2/1)$	$P_6 (-1/3)$	$P_9 (1/4)$
$P_1 (0/1)$	$P_4 (-2/2)$	$P_7 (-1/4)$	$P_{10} (1/3)$
$P_2 (-1/1)$	$P_5 (-1/2)$	$P_8 (0/4)$	$P_{11} (2/3)$

2. Relative Gitterkoordinaten

$P_0 (0/0)$	$P_{2 \rightarrow 3} (L)$	$P_{5 \rightarrow 6} (O)$	$P_{7 \rightarrow 9} (R)$
$P_{0 \rightarrow 1} (O)$	$P_{3 \rightarrow 4} (O)$	$P_{6 \rightarrow 7} (O)$	$P_{9 \rightarrow 10} (U)$
$P_{1 \rightarrow 2} (L)$	$P_{4 \rightarrow 5} (R)$	$P_{7 \rightarrow 8} (R)$	$P_{10 \rightarrow 11} (R)$

3. Relative Konformationskoordinaten

$P_0 (0/0)$	$P_{2 \rightarrow 3} (G)$	$P_{5 \rightarrow 6} (LG)$	$P_{7 \rightarrow 9} (G)$
$P_{0 \rightarrow 1} (G)$	$P_{3 \rightarrow 4} (RG)$	$P_{6 \rightarrow 7} (G)$	$P_{9 \rightarrow 10} (RL)$
$P_{1 \rightarrow 2} (LG)$	$P_{4 \rightarrow 5} (RG)$	$P_{7 \rightarrow 8} (RG)$	$P_{10 \rightarrow 11} (LG)$



$P_0 (0/0)$	$P_{2 \rightarrow 3} (0/1)$	$P_{5 \rightarrow 6} (1/0)$	$P_{7 \rightarrow 9} (0/1)$
$P_{0 \rightarrow 1} (1/0)$	$P_{3 \rightarrow 4} (1/0)$	$P_{6 \rightarrow 7} (1/0)$	$P_{9 \rightarrow 10} (1/0)$
$P_{1 \rightarrow 2} (0/1)$	$P_{4 \rightarrow 5} (0/1)$	$P_{7 \rightarrow 8} (0/1)$	$P_{10 \rightarrow 11} (0/1)$

Epistasie messen:

1. Wenn in absoluten Koordinaten ein Punkt verschoben wird, verändern sich alle anderen ohne weiteres erstmal gar nicht!

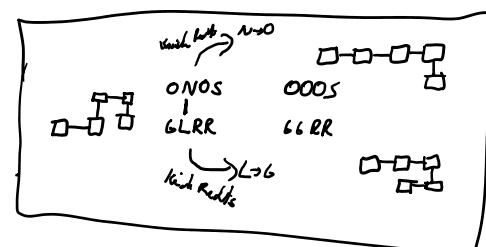
→ Es entsteht eine Lücke in der Proteinfaltung und eine Epistasie im Phänotyp ab der Position, an der die Verschiebung stattfindet.

Dies kann alle Positionen $n-2$ bis $n-(n-1)$ betreffen.

Epistasie = Abhängigkeit & Anzahl aller Gene nach verschobenem.

2. In relativen Gitterkoordinaten kann bei einer Mutation keine Lücke mehr entstehen, da Folgekoordinaten anhand vorselektierter und einer Himmelsrichtung erreicht werden.

Jedoch besteht die Möglichkeit, dass ab dem Mutationspunkt, durch die Folgekoordinaten mit falscher Richtung, die sich nicht auto. ändert, eine fälschlich andere Faltung entsteht und somit doch Epistasie entsteht. Keine funktionale "Abhopping" aber anderes Layout.

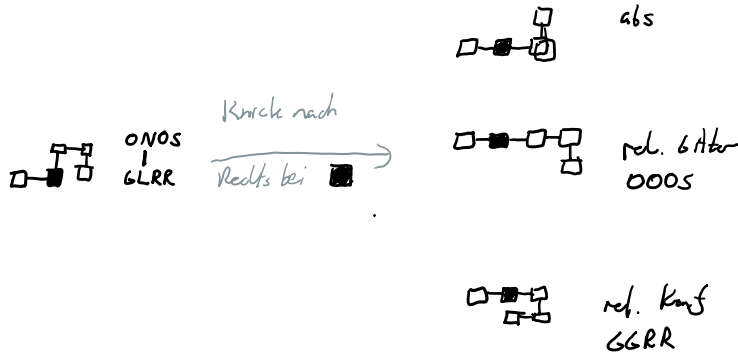


3. Außer auftretender Überlappungen, wie auch bei 1 & 2 möglich. Behält die Faltstruktur ab dem Mutationspunkt ihre Struktur und hat somit die ...

3. Außer auftretender Überlappungen, wie auch bei 1 & 2 möglich. Behält die Folgefolge ab dem Mutationspunkt ihre Struktur und hat somit die geringsten Epitussreduenzen, bzw. gar keine Änderung im Phänotypabschnitt nach dem Mutationspunkt

Geringste Epitase

Die relativen Konformationskoordinaten!



Algorithmus für Überlappung bei relativen Konformationen ohne Koordinaten!

Function checkOverlap (List <Directions> d) // use Stack to check overlaps for rel better.
 { if we move North 2 times and South 2 times same pos.

Array <Directions> a = []
 int overlaps = 0
 bool started = false;

for (int i=0; i < d.Count(); i++)

{

if (a.Count == 0)

a.push(d[i]);

if (started) // we started and found an overlap due to opposing directions
 overlaps++;

else

{ // if we were in opposing directions remove -> same place

if (a[a.Count-1] == NORTH && d[i] == SOUTH

|| a[a.Count-1] == EAST && d[i] == WEST

|| a[a.Count-1] == SOUTH && d[i] == NORTH

|| a[a.Count-1] == WEST && d[i] == EAST)

a.remove(a.Count-1);

else

a.push(d[i]); // other direction: push

}

started = true;

}

return overlaps;

}

Coding ohne overlaps

Koordinatensystem Klasse mit Feldern und bool besetzt.

Viel Speicherverbrauch ☹

Class Tabelle besetzt = true.

	7	8	9	
	6		10	11
4	5			
3	2	1		
		0		