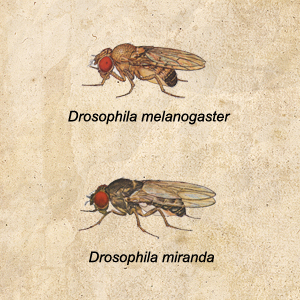
**FRUITVLIEGEN**

****

Arjan van Hoogdalem

Universiteit van Amsterdam

Daan Vial

Universiteit van Amsterdam

Timo Fernhout

Universiteit van Amsterdam

**Inleiding**

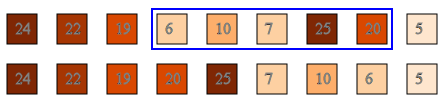


**Afbeelding 1: Hier weergegeven zijn de genomen van twee fruitvliegen.**

*Casus*

De casus gaat over gen mutaties bij fruitvliegjes en dan specifiek de snelste mutatie route tussen twee verschillende fruitvlieg genomen. De twee fruitvliegen zijn Drosophila Melanogaster en Drosophila Miranda, zie afbeelding 1.

Per generatie fruitvliegjes kan er één mutatie plaatsvinden deze mutatie is gedefinieerd als een inversie van een stuk genoom, zie afbeelding 2.



Afbeelding 2: Hier weergegeven zijn twee stukken genoom op de zelfde locatie van opvolgende generaties. Het omlijnde stuk genoom in de eerste generatie is bij de voortplanting omgedraaid.

Het hoofddoel van deze casus is om voor twee gedocumenteerde fruitvlieg genomen (Melanogaster & Miranda) te kijken wat de minimale hoeveelheid mutaties is die er nodig zijn om van genoom Melanogaster bij genoom Miranda te komen. Hiermee kan bepaald worden hoeveel generaties er tussen de twee fruitvlieg genomen zit.

Hiernaast wordt er ook gekeken of er een mutatieroute met zo min mogelijk verplaatsing van genen is. Wat betekent dat hoe kleiner de inversie, hoe minder gen verandering er plaats vinden. Dit wordt gedaan om te kijken of de uitkomsten overeenkomen met die van de hoofdcasus. Hiermee zou meer duidelijkheid kunnen komen over de oplossingspatronen.

*Toestandsruimte*

Omdat de lengte van het genoom 25 is, is het aantal mogelijke genomen 25! ≈ 1.55 \* 1025. Dit is echter nog niet de toestandsruimte, omdat bij deze casus niet het aantal mogelijke genomen van belang is, maar de transities tussen deze genomen. Bij elke mutatie zijn er 300 mogelijke mutaties. Dit valt te berekenen doordat er 1 mutatie is van lengte 25, er zijn 2 mutaties van lengte 24, er zijn 3 mutaties van lengte 23, enzovoorts tot en met 24 mutaties van lengte 2. Wanneer alle mutaties opgeteld worden krijg je het getal 300, 1+2+3+…+22+23+24 = 300. Hiermee kan een inschatting gemaakt worden van de toestandsruimte. Omdat een mutatie de ene kant op, ook een mutatie de andere kant op is, nemen we de helft van het aantal mogelijke mutaties. Een inschatting van de toestandsruimte is dan 150 \* 25! ≈ 2.33 \* 1027. Deze toestandsruimte is te groot om helemaal door te rekenen.

**Methodes casus deel 1**

Omdat de toestandsruimte te groot is, zal er gekeken moeten worden naar een lokaal optimum. Een lokaal optimum is te bereiken door telkens lokaal de ‘beste’ mutatie te kiezen. De vraag is dan echter wat een ‘beste’ mutatie is.

*Absolute positie*

Als eerst kan er gedacht worden aan absolute positie. Dit betekent dat een mutatie de ‘beste’ is, wanneer die mutatie zoveel mogelijk genen op de juiste doelpositie zet. Op deze manier kan er echter geen oplossing gevonden worden. Dit komt omdat er soms een ‘slechte’ stap nodig is om bepaalde genen naar de juiste positie te kunnen brengen. Dit is bijvoorbeeld goed te zien in Afbeelding 3. In groen staat aangegeven welke genen op de juiste positie staan, in rood welke genen nog niet op de juiste positie staan. Om gen 7 naar de juiste positie te kunnen brengen is het nodig om een swap te maken van een hele rij aan genen die al op de juiste positie staan. Wanneer er alleen gekeken wordt naar de absolute positie is zo een mutatie niet mogelijk.

1 2 3 4 23 21 25 8 5 20 18 12 13 14 15 16 17 7 10 6 19 22 11 24 9

**Afbeelding 3: Hier weergegeven is een voorbeeld van absolute positie.**

*Relatieve positie*

In plaats van absolute positie kan er ook gebruik worden gemaakt van relatieve positie. Relatieve positie houdt in dat er gekeken wordt naar de buren van elk gen en wanneer de buren de gewenste doel-buren zijn, dan is er sprake van een juiste relatieve positie. Wanneer gen 3 de genen 2 en 4 als buren heeft, dan heeft gen 3 een optimale relatieve positie. Zie Afbeelding 4 voor een illustratie. In groen is aangegeven welke genen een juiste relatieve positie hebben, in blauw welke genen een bijna-juiste relatieve positie hebben en in rood welke genen geen juiste relatieve positie hebben. Een ‘beste’ mutatie is dan een mutatie die zoveel mogelijk genen een juiste relatieve positie geeft, oftewel zoveel mogelijk ‘goede buren’. Met relatieve positie kan wel een oplossing gevonden worden. In eerste instantie hebben we hiermee een pad gevonden van 14 mutaties.

1 2 3 4 23 21 25 8 5 20 18 12 13 14 15 16 17 7 10 6 19 22 11 24 9

**Afbeelding 4: Hier weergegeven is een voorbeeld van relatieve positie.**

*Willekeurig ‘beste’ mutaties kiezen*

Bij de eerste implementatie van relatieve positie werd een pad gevonden van veertien mutaties. Dit hoeft echter niet het kortste pad te zijn, omdat er bij iedere stap vaak meerdere ‘beste’ mutaties zijn. In onze eerste implementatie wordt de mutatie gekozen die als eerst een ‘beste’ mutatie was. Om een idee te krijgen of er kortere paden te vinden zijn hebben we een simulatie geschreven waarbij er telkens willekeurig een van de ‘beste’ mutaties gekozen wordt totdat het doel-genoom gevonden is. Deze simulatie kan een aantal keer gedraaid worden, waarbij aan het eind het kortste pad dat gevonden is wordt teruggegeven. Wanneer deze simulatie 50 keer gedraaid wordt, wordt in de meeste gevallen een pad gevonden van dertien mutaties.

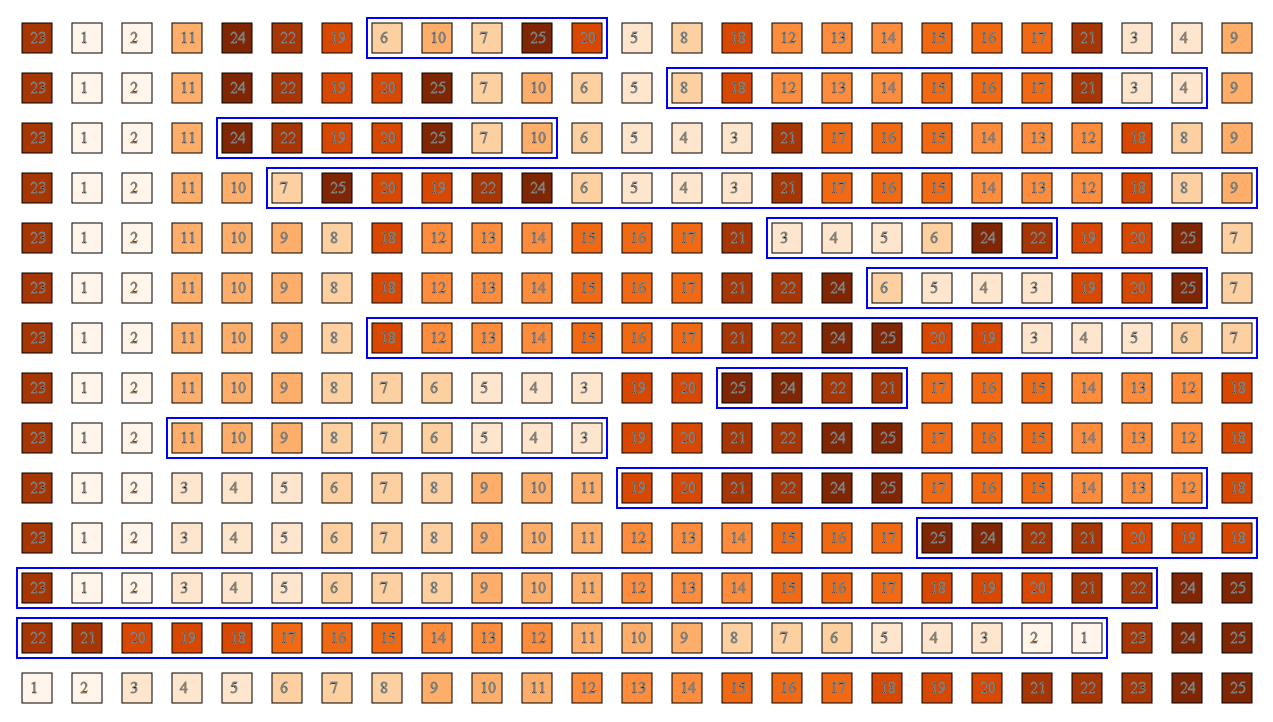
*Een Tree van ‘beste’ mutaties*

Om er zeker van te zijn dat er elke keer een pad van 13 mutaties gevonden zal worden hebben we een Tree gemaakt. Deze Tree wordt met behulp van een breadth-first search algoritme laag voor laag opgebouwd. Na elke laag wordt er gecontroleerd of het doel-genoom gevonden is. Een laag bestaat uit alle genomen die volgen uit alle gevonden ‘beste’ mutaties van alle genomen uit de vorige laag. Om de Tree te optimaliseren en ervoor te zorgen dat de Tree überhaupt te bouwen is, is er voor gekozen om alleen een genoom toe te voegen aan de Tree wanneer deze nog niet in de Tree zit. Doordat de Tree laag voor laag wordt opgebouwd wordt er hierbij niet een sneller pad weggegooid. Het is echter hierdoor niet mogelijk om meerdere kortste paden te vinden in de Tree, omdat er voor elk nieuw gevonden genoom maar één ouder-genoom wordt opgeslagen. Een indicatie van hoeveel sneller dit de Tree maakt, zal besproken worden bij de resultaten. Het duurt ongeveer 15 minuten om de Tree te bouwen.

**Resultaten**

*Kortst gevonden pad*

Door het gebruik maken van relatieve positie hebben we een pad kunnen vinden van 13 mutaties van Drosophila Melanogaster naar Drosophila Miranda, zie afbeelding 5 voor een visualisatie hiervan.



**Afbeelding 5: Hier weergegeven is een pad van 13 stappen van Melanogaster naar Miranda.**

*De Tree*

In totaal bestaat de Tree uit 42332 genomen. Verder is er onderzocht hoe vaak het voorkomt een genoom die al in Tree zit, nu niet opnieuw toegevoegd wordt. Hierbij maken we een onderscheid tussen duplicaten die zich in dezelfde laag bevinden, dus wanneer een genoom meerdere keren in dezelfde laag gevonden wordt en het totale aantal duplicaten, dus ook met genomen die zich al in eerdere lagen uit de Tree bevinden. Het totale aantal niet toegevoegde duplicaten is 106939 en aantal niet toegevoegde duplicaten uit dezelfde laag is 11549. Zie Tabel 1 voor meer gegevens over de Tree.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Aantal genomen per laag | Aantal duplicaten in dezelfde laag per laag | Totaal aantal duplicaten per laag |
| Laag 1 | 1 | 0 | 0 |
| Laag 2 | 1 | 1 | 1 |
| Laag 3 | 17 | 1 | 1 |
| Laag 4 | 103 | 13 | 41 |
| Laag 5 | 441 | 66 | 261 |
| Laag 6 | 1692 | 245 | 1397 |
| Laag 7 | 4629 | 755 | 5700 |
| Laag 8 | 8188 | 1577 | 14668 |
| Laag 9 | 9373 | 2454 | 23989 |
| Laag 10 | 7316 | 2475 | 24590 |
| Laag 11 | 4853 | 1915 | 17498 |
| Laag 12 | 3236 | 1311 | 10982 |
| Laag 13 | 2481 | 736 | 7810 |

**Tabel1: Hier weergegeven is het aantal genomen per laag, het aantal duplicaten in dezelfde laag per laag en het totaal aantal duplicaten per laag.**