Universitaire Instelling Antwerpen
Departement Wiskunde-Informatica

DNA computing en gen-assemblage in ciliaten.

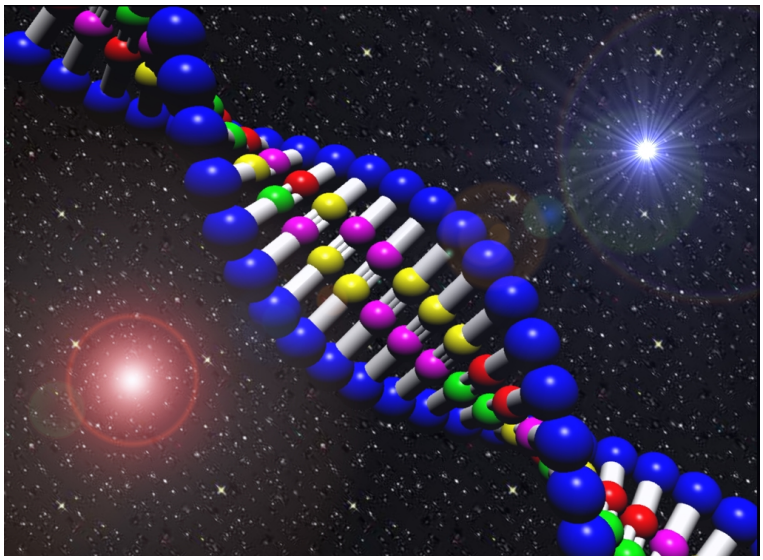
Sven Maerivoet
Erik Vanherck

2e licentie Informatica
Academiejaar 2000 - 2001

Opdracht voor het vak *Seminarie Informatica*
Titularis : Dirk Janssens

Samenvatting

In dit artikel wordt een korte bespreking gegeven van wat DNA computing juist inhoudt. Dit impliceert in eerste instantie een biologische benadering die de bouwstenen (DNA-moleculen) en enkele gebruikte technieken (bewerkingen met DNA-moleculen) toelicht. Vervolgens wordt de DNA-verwerking in een bepaalde soort van organismen (de ciliaten) besproken en hierop gebaseerd, wordt een meer formele beschrijving gegeven. Dit geeft aanleiding tot een computationele aanpak die de basis vormt voor reductiesystemen en onder andere een toepassing vindt in het oplossen van NP-harde problemen. Tot slot wordt er summier gekeken naar de toekomst op korte en lange termijn en wordt de relatie tussen DNA computing en quantum computing belicht.



Een artistieke impressie van een dubbele DNA-streng.

Inhoudsopgave

Lijst van figuren

1 Grondbeginselen van DNA computing

In dit deel wordt kort toegelicht wat DNA computing juist inhoudt en wordt de structuur van de essentiële bouwstenen (DNA-moleculen) beschreven. Verder wordt er ingegaan op de verschillende soorten bewerkingen die met DNA-moleculen kunnen gedaan worden, wat de praktische basis vormt voor het computationeel aspect dat aan DNA computing verbonden is.

De bespreking in dit deel is hoofdzakelijk gebaseerd op [HR01]. De beschrijving van de opbouw van DNA-moleculen is ook terug te vinden in [KKG00] en [Kar97].

1.1 Inleiding

DNA (*deoxyribo-nucleic acid*) computing is een onderdeel van het veld van Natural Computing, waarbij men concepten onleent aan de natuur. Bij DNA computing bekijkt men het berekeningsproces vanuit het standpunt van de moleculaire biologie. Men gaat voor de hardware, waarmee men algoritmes uitvoert, gebruik maken van DNA-moleculen en enzymen. Naar dergelijke hardware verwijst men dan ook dikwijls als *bioware*.

1.2 De opbouw van DNA

1.2.1 Wat is een DNA-molecule ?

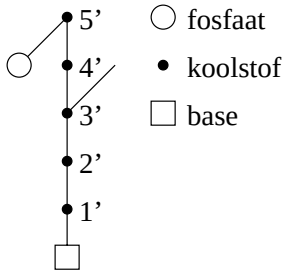
Een DNA-molecule is opgebouwd uit een aantal eenvoudige monomeren, nucleotiden genaamd. Een dergelijke reeks aan elkaar geregen enkelvoudige nucleotiden, vormt een enkelvoudige DNA-streng. Twee enkelvoudige DNA-strengen kunnen dan weer binden om een dubbele DNA-streng te vormen die in zijn natuurlijke vorm er als een dubbele helix uitziet. De beschrijving van DNA zoals ze hier gebeurt, is sterk vereenvoudigd maar voldoet voor de doeleinden van dit artikel.

1.2.2 Wat is een nucleotide ?

Een nucleotide bestaat uit een suikermolecule, die vijf koolstofatomen bevat, een fosfaat en een base. Ze zijn onderling verbonden, zoals weergeven in figuur 1. Er bestaan vier soorten nucleotiden in DNA : adenosine (dat een *adenine* base bevat), guanosine (dat een *guanine* base bevat), thymidine (dat een *thymine* base bevat) en cytidine (dat een *cytosine* base bevat). Deze basen worden afgekort met de letters A, G, T en C.

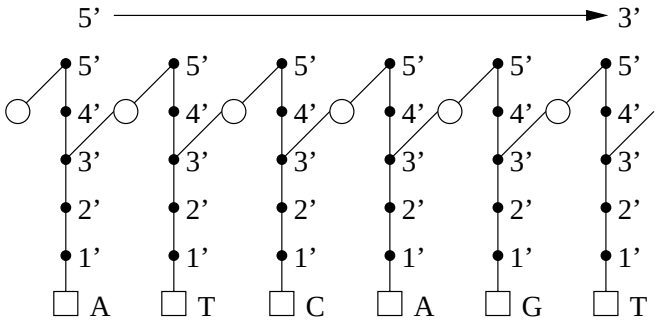
1.2.3 Bindingen

Een nucleotide kan zich binden met een andere nucleotide doordat er een (sterke) *covalente* binding (ook wel fosfodiëster-binding genoemd) ontstaat tussen het vijfde koolstofatoom (5')



Figuur 1: Een nucleotide.

met zijn fosfaat en het derde koolstofatoom (3’) van de andere nucleotide. Aldus wordt een enkelvoudige DNA-streng verkregen met langs de ene zijde een vrije 5’ en langs de andere zijde een vrije 3’ (zie figuur 2). Een dergelijke (korte) keten van een twintigtal nucleotiden wordt ook wel een *oligonucleotide* genoemd. Daar deze twee zijden andere chemische eigenschappen hebben, kan men ze van elkaar onderscheiden en aldus bezit een DNA-streng een zekere *polariteit*.

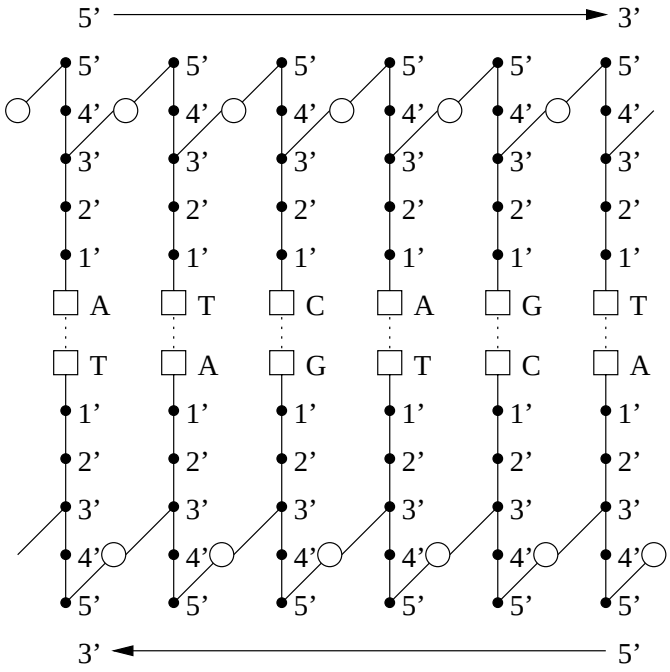


Figuur 2: Een enkelvoudige DNA-streng.

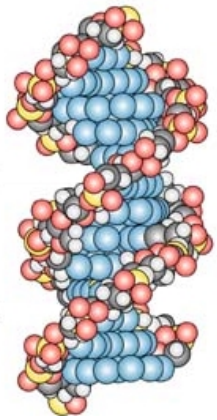
Nucleotiden verschillen qua opbouw enkel in hun base (zie paragraaf 1.2.2). Er kunnen (zwakke) bindingen (dit zijn combinaties van *Van der Waals krachten* en *waterstof*-bindingen[1]) ontstaan tussen twee basen en op die manier wordt een dubbele DNA-streng gevormd (zie figuur 3). Er is echter wel een beperking, de basen binden met name enkel op de volgende manieren : A met T en C met G, ook wel Watson-Crick complementair genoemd. Bovendien moeten beide enkelvoudige DNA-strengen tegengesteld georiënteerd zijn. De twee suiker-fosfaat ruggegraten winden zich rond de gebonden basen in de vorm van een dubbele helix (zie figuur 4), waarin ze anti-parallel lopen (ze hebben bij binding dus een verschillende polariteit). Dit model werd door James Watson en Francis Crick in 1952 bevestigd.

Merk op dat er ook nog RNA (*ribo-nucleic acid*) bestaat dat qua opbouw van nucleotiden lichtjes van DNA verschilt. De gebruikte basen zijn dezelfde, met uitzondering van thymine dat hier

[1]Tussen elk (A, T) paar worden *twee* waterstof-bindingen gevormd en tussen elk (C, G) paar worden *drie* waterstof-bindingen gevormd.



Figuur 3: Een dubbele DNA-streng.



Figuur 4: De dubbele helix die door DNA gevormd wordt.

wordt vervangen door *uracil* (afgekort als U). Er geldt nog steeds dat C en G complementair zijn, maar U is nu complementair met zowel A als G.

Volgens de laatste theorieën, bestaan er twee soorten menselijk DNA : *opvullend* DNA en *coderend* DNA (waarbij opvullend DNA ongeveer 95% inneemt). In het coderend DNA ligt de genetische informatie (gebruikt voor het produceren van eiwitten of RNA-moleculen) opgeslagen. Het is onder andere bij het extraheren van deze informatie uit een DNA-helix dat RNA een belangrijke rol speelt (de DNA-ketens mogen immers niet beschadigd worden, daarom dat ze met behulp van RNA worden gekopieerd).

1.2.4 Formele notatie

Omdat nucleotiden enkel verschillen qua base en omdat ze een polariteit hebben, neemt men de conventie aan om de natuurlijke oriëntatie van een DNA-molecule (5'-3') te schrijven van links naar rechts. Met andere woorden, $ATCAGT$ is de notatie voor de molecule in figuur 2. Voor een dubbele DNA-streng gebruikt men de conventie waarbij men de bovenste van beide strengen in de 5'-3' zin schrijft. Dit levert volgende notatie voor de molecule in figuur 3 :

$$\begin{matrix} ATCAGT \\ TAGTCA \end{matrix}.$$

1.2.5 Onvolmaaktheden

Het is echter zo dat niet alle DNA-moleculen perfecte dubbels zijn. Vaak komt het voor dat langs één van beide zijden een stuk ontbreekt. Bijvoorbeeld : bij

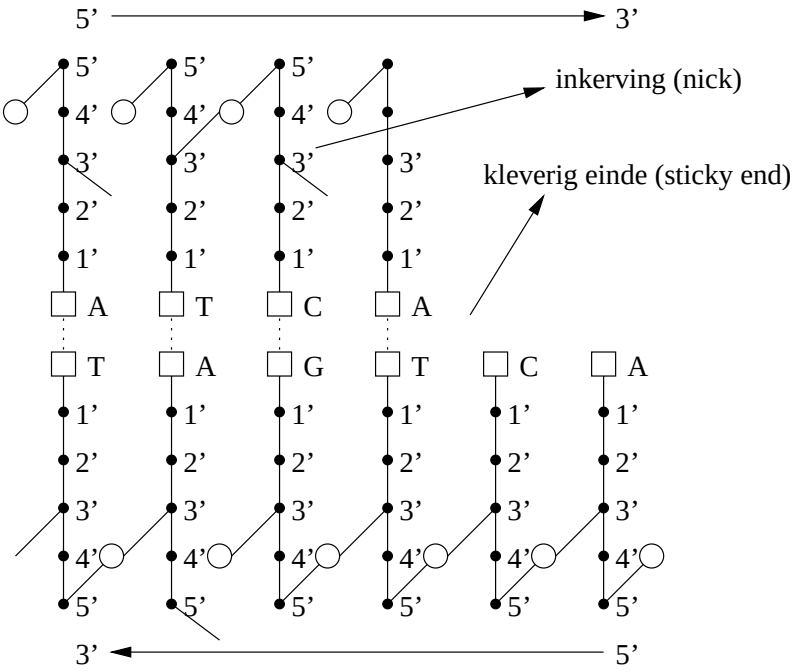
$$\begin{matrix} ATCA \\ TAGTCA \end{matrix}$$

ontbreekt een stukje van de bovenste DNA-streng. De CA op overschot van de onderste DNA-streng noemt men een *kleverig einde* (*sticky end*). De benaming komt van het feit dat deze molecule zich (tijdelijk) in een 'moleculaire soep' zal vasthechten aan een andere molecule die het complementair kleverig einde GT bezit. Ook bij de sterke covalente bindingen kan het voorkomen dat er een binding ontbreekt, dit noemt men dan een *inkerving* (*nick*). Beide onvolmaaktheden worden geïllustreerd in figuur 5.

1.3 Bewerkingen met DNA-moleculen

De zeer complexe structuur van levende wezens, is het resultaat van het toepassen van eenvoudige operaties op initiële informatie die door de DNA-sequenties gecodeerd wordt. De parallel met het computationele aspect is eenvoudig : het resultaat van het toepassen van een berekenbare functie op een argument, kan bekomen worden door het toepassen van een combinatie van simpele basisfuncties op dit argument. De DNA-strengen vormen dan de initiële informatie en men gebruikt *enzymen* om de eenvoudige berekeningen te simuleren.

Door de verschillende operaties – die met DNA-moleculen kunnen uitgevoerd worden – samen te stellen, kan men algoritmes coderen/vormen die problemen oplossen. Een voorbeeld hiervan is het oplossen van het Hamiltoniaans Pad Probleem (een afgezwakte versie van het Handelsreiziger Probleem (*Travelling Salesman Problem*)), dat in 1994 als eerste succesvolle toepassing werd uitgewerkt door Leonard Adleman (hij werkte met een graf met zeven knopen) en de mogelijkheden toonde die DNA computing herbergt.



Figuur 5: Kleverig einde (*sticky end*) en inkervingen (*nicks*) in dubbele DNA-strengen.

De bewerkingen die veelal met DNA-moleculen worden gedaan, zijn de volgende :

Uitgloeien (*annealing*) of hybridisatie : hierbij worden twee enkelvoudige moleculen samengevoegd tot een dubbele molecule doordat de basen (zwakke) waterstof-bindingen vormen.

Denatureren : dit is de omgekeerde operatie van uitgloeien. Deze operatie breekt met andere woorden een dubbele molecule op in zijn twee enkelvoudige moleculen (dit kan bereikt worden door middel van bijvoorbeeld verwarming of chemische reacties).

Verbinden (*ligation*) : wanneer er een (sterke) covalente binding ontbreekt in een molecule (bijvoorbeeld doordat twee moleculen via hun kleverige eindes aan elkaar gehecht zijn), dan is deze molecule niet echt stabiel. Dit komt omdat ze enkel samengehouden wordt door de zwakke bindingen tussen de basen. Het enzyme *ligase* herstelt echter de inkervingen in de gehele molecule.

Scheiden op basis van een gegeven patroon : men heeft dikwijls nood aan een deel van de moleculen met een specifiek patroon. Als men een bepaald patroon zoekt, kan men een grote massa complementaire moleculen aanmaken en die gebruiken als 'aas'. Daartoe bevestigt men deze moleculen aan microscopische glazen kraaltjes die men dan dicht opeenstapelt in een heel dunne glazen kolom. Vervolgens giet men de oorspronkelijke substantie door deze kolom, wat als gevolg heeft dat alle moleculen met het gewenste patroon aan hun complementen zullen blijven hangen terwijl al de overige gewoon wegvloeien (men noemt dit proces dan ook wel *affinity separation*). Dan kan men door middel van een stof die denatureren bevordert, de gevangen moleculen weer losmaken en in een apart schaaltje verzamelen. Merk op dat deze techniek niet geschikt is voor het scheiden van kleine patronen.

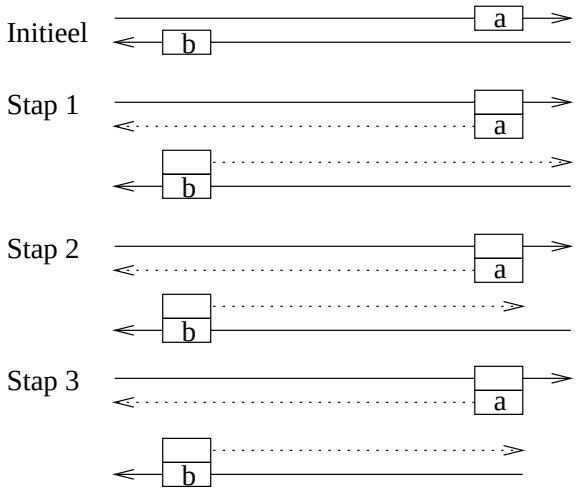
Scheiden op basis van lengte : de lengte van een enkelvoudige molecule is gewoon het aantal nucleotiden die ze bevat. Voor een dubbele molecule veronderstelt men dat ze compleet is en definieert men de lengte als het aantal paren basen. Een DNA-molecule is negatief geladen, dus als men ze tussen een positieve elektrode (*anode*) en negatieve elektrode (*kathode*) zet en de spanning activeert, dan zal ze aangetrokken worden door de positieve elektrode. In geval er geen obstakels zijn, zullen alle moleculen even snel deze elektrode bereiken (grotere moleculen hebben namelijk ook een grotere negatieve lading). Als men dit proces nu toepast in een gel, dan zullen de grotere moleculen moeilijker vooruit komen en dus ook trager vorderen. Wanneer de eerste moleculen de positieve elektrode bereiken, kan men de spanning afzetten en zullen de moleculen doorheen de gel verspreid zijn, al naargelang hun lengte. Men kan deze dan tellen door ze te bevuilen met bijvoorbeeld ethidium bromide wat maakt dat onder ultraviolet licht de moleculen van dezelfde lengte in de gel als banden worden weergegegeven. Men noemt dit proces ook wel *gel-elektrophorese*.

Vermenigvuldigen van DNA : wanneer men een bepaalde molecule zoekt in een gigantische hoop, dan is het interessant indien men deze kan vermenigvuldigen zodat er zeer veel van beschikbaar worden. *DNA polymerase* is een enzyme dat van een enkelvoudige streng een

dubbele molecule maakt door basen aan elkaar te hechten en verbindingen (ligaties) te vormen in geval van inkervingen. Maar dit enzyme heeft een startpunt nodig, met andere woorden, een klein stukje van het originele DNA moet dubbel zijn. Door middel van uitgloeien kan men een dergelijk klein stukje (dit wordt een *primer* genoemd) bevestigen aan het 3' einde van de molecule die men wilt vermenigvuldigen. Daarna zal polymerase de rest van de dubbele molecule vervolledigen met losse nucleotiden met de juiste basen. Om dit te kunnen gebruiken, moet men van de dubbele molecule twee kleine sequenties van basen weten langs de 3' zijden waarvan men dan de complementen kan aanmaken. Dan voegt men deze, tesamen met polymerase en een voldoende grote hoeveelheid losse nucleotiden, aan de substantie toe. Hierna itereert men volgende drie stappen in het proces :

1. denatureren door middel van opwarming,
2. afkoeling waardoor de primers zich bevestigen
3. en lichtjes opwarmen zodat polymerase zijn gang kan gaan.

Hierdoor worden er duplicaten gemaakt van elk van de enkelvoudige moleculen die een primer bezitten. Het vermenigvuldigen van DNA wordt geïllustreerd in figuur 6, wat de polymerase kettingreactie (*polymerase chain reaction*) – afgekort als PCR – voorstelt.



Figuur 6: Polymerase kettingreactie (*PCR*).

1.4 Conclusie

DNA computing probeert twee grote problemen met de op silicium gebaseerde computers op te lossen. De continue miniaturisatie van deze computers zal ergens een einde kennen. Daar men echter in DNA computing op een *nano-schaal* werkt, is dit hier niet direct een probleem. Het andere grote nadeel van de traditionele op silicium gebaseerde computers is dat deze zeer moeilijk op grote schaal te parallelliseren zijn. Ze zijn zo opgebouwd dat ze sequentieel algoritmes

uitvoeren. Bij DNA computing werkt men zelfs in de kleinste testube al met enorme hoeveelheden moleculen, zodat hier een operatie uitgevoerd wordt op miljarden moleculen tegelijk. Hier kan men dus spreken van *parallelisme op zeer grote schaal*. Het is dan ook mogelijk om het exponentiële algoritme voor het oplossen van het Hamiltoniaans Pad Probleem binnen een redelijke tijdspanne uit te voeren (dit omdat in de klasse NP problemen in polynomiale tijd met behulp van een niet-deterministisch algoritme kunnen opgelost worden), terwijl dit op een silicium gebaseerde machine onmogelijk zou zijn.

2 Gen-assemblage in ciliaten

2.1 Inleiding

Eén van de grote doorbraken op het vlak van DNA computing is de mogelijkheid om genetische systemen in bestaande organismen op formele wijze te beschrijven. In dit deel wordt de nadruk gelegd op de verwerking van DNA in een bepaalde soort van organismen, namelijk de *ciliaten*. Deze verwerking is zeer complex en duidelijk computationeel van aard (vergelijkbaar met Turing machines) wat aanleiding geeft tot enkele volledige formele beschrijvingen ervan (die allen equivalent met elkaar zijn). Daar waar het vorige deel van dit artikel handelde over DNA-verwerking *in vitro*, handelt dit deel over DNA-verwerking *in vivo*. Het is grotendeels gebaseerd op [EHP+01].

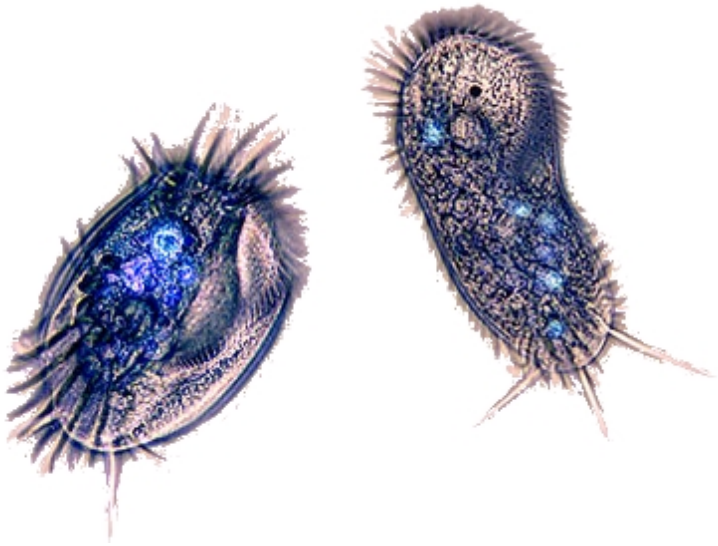
2.2 Ciliaten

2.2.1 Een korte biologische beschrijving

Ciliaten (zie figuur 7) zijn een verzamelnaam voor een zeer oude groep unicellulaire organismen (wel meer dan 7000). De naam ‘ciliate’ is afgeleid van het Latijn en betekent ‘wimper’. Een deel van het celoppervlak is immers bedekt met zeer korte harige structuren – de cilia – die ervoor zorgen dat de organismen zich kunnen voortbewegen (zie [WS94]). Er is sprake van nucleair dualisme, ze hebben namelijk twee soorten celkernen (*nuclei*) : een actieve macro-nucleus (*soma*) en een functioneel inerte micro-nucleus (*germline*) (zie [KKL99] en [KL00]). Deze laatste draagt enkel bij tot de seksuele voortplanting die optreedt wanneer een ciliate uitgehongerd is (en waarbij de micro-nucleus wordt geactiveerd, zie paragraaf 2.2.2).

2.2.2 De verwerking van DNA

Tijdens de ontwikkeling zal de somatische actieve macro-nucleus gevormd worden uit de micronucleus (na seksuele voortplanting). De micro-nucleus bevat chromosomen en wordt door mi-



Figuur 7: Twee hypotrichoïde ciliaten (*Euplotes* en *Stylonychia*).

tose[2] gedeeld. Ze ondergaat meiose[3] en er vindt uitwisseling van celkernen plaats wanneer cellen 'paren'. Let wel, aan het DNA wordt niet geraakt, de macro-nucleus daarentegen voorziet het nodige RNA voor de cel-operaties (het DNA moet immers uit de celkern gehaald worden zonder het te beschadigen) en wordt gevormd uit een tweede kopie van de micro-nucleus (zie [Pre99]).

De kopieën van sommige eiwit-gecodeerde genen in de micro-nucleus van hypotrichoïde ciliaten worden afgeschermd door de aanwezigheid van niet-eiwit-gecodeerde DNA-sequenties (men noemt deze interne geëlimineerde sequenties, IESs (*internally eliminating sequences*)). Deze IESs dienen verwijderd te worden alvorens de assemblage van een functionele kopie van een gen in de macro-nucleus (zie [KKL99] en [KL00]).

Verder geldt dat in de Oxytricha- en Stylonychia-ciliaten het kan gebeuren dat de eiwit-gecodeerde DNA-segmenten (*macronuclear destined sequences*, MDSs) voorkomen in een gepermuteerde volgorde (relatief ten opzichte van hun eindpositie in de macronucleaire kopie). Ook hier dient verwerking van het DNA te gebeuren opdat alle segmenten in de juiste, gebruikelijke volgorde worden gezet (zie [KKL99] en [KL00]).

[2]Bij mitose (deelproces van de celdeling) deelt een cel zich in twee dochtercellen waarvan de celkernen genetisch identiek zijn en hetzelfde aantal chromosomen als de ouderkern bevatten (het genetisch materiaal is dus verdubbeld), zie [dA01] voor meer details.

[3]Bij meiose treedt er geen verdubbeling van het aantal chromosomen op, men spreekt daarom ook wel van *reductiedeling*. Merk op dat het in beide dochtercellen aanwezige erfelijk materiaal *verschilt*, zie [dA01] voor meer details.

2.3 Formele voorstelling

De verwerking van DNA (van micro- naar macro-nuclei) in ciliaten kan formeel voorgesteld worden vermits deze computationeel van aard is. Het nucleotide-alfabet Σ bestaat uit de basen $\{A, C, G, T\}$. De (tekstuele) voorstelling van DNA-strengen die gebruikt wordt, is terug te vinden in paragraaf 1.2.4. Gegeven een string α (wat een enkelvoudige DNA-streng voorstelt), kan men de inversie $\overline{\alpha}$ ervan berekenen door eerst het Watson-Crick complement te nemen (vervang A door T en C door G en vice versa) en vervolgens de string omgekeerd te schrijven : $\alpha = ACATG$ wordt dan eerst $TGTAC$ wat vervolgens resulteert in de inversie $\overline{\alpha} = CATGT$. Voor een dubbele string γ (wat een dubbele DNA-streng voorstelt), wordt de inversie berekend door eerst de twee strings om te wisselen en vervolgens beide omgekeerd te schrijven. Een voorbeeld :

$$\begin{matrix} AACTGA \\ TTGACT \end{matrix} \Rightarrow \begin{matrix} TCAGTT \\ AGTCAA \end{matrix}.$$

Een gen τ uit een micro-nucleus bestaat uit een eindige reeks eenmalig voorkomende MDSs die worden gescheiden door IESs. In een dergelijke reeks $\{M_1, \ldots, M_k\}$ $(k \geq 2)$ MDSs in τ, heeft elke M_i $(i \in \{2, \ldots, k-1\})$ volgende structuur :

$$M_i = \left(\begin{matrix} p_i \\ \overline{p}_i \end{matrix}, \mu_i, \begin{matrix} p_{i+1} \\ \overline{p}_{i+1} \end{matrix} \right),$$

met M_1 en M_k van de vorm :

$$M_1 = \left(b, \mu_1, \begin{matrix} p_2 \\ \overline{p}_2 \end{matrix} \right), M_k = \left(\begin{matrix} p_k \\ \overline{p}_k \end{matrix}, \mu_k, e \right).$$

Hierin geldt dat de μ_i (met $i \in \{1, \ldots, k\}$) *volledige,* dubbele DNA-strengen zijn en men noemt ze ook wel het *lichaam* van M_i. Vermits ze volledig zijn, geldt dat hun inversies $\overline{\mu}_i$ dat ook zijn. De p_i en $\overline{p}_i$ (met nu $i \in \{2, \ldots, k\}$) zijn enkelvoudige DNA-strengen die samen een dubbele DNA-streng vormen. Het koppel $\langle p_i, \overline{p}_i \rangle$ wordt ook wel een wijzer (*pointer*[4]) genoemd en het is de inkomende wijzer voor een zeker MDS M_i en de uitgaande wijzer voor een zeker MDS M_{i-1}. Verder zijn b en e begin- en eindmarkeringen die gebruikt worden als symbolische markeringen van de plaatsen waar een beginnende macro-nucleaire DNA-molecule uit een micro-nucleair genoom wordt gesneden.

Merk op dat een wijzer $\langle p_i, \overline{p}_i \rangle$ altijd op het einde van een MDS gepositioneerd is (meestal op de grens tussen een MDS en een IES). Dezelfde sequentie kan ook elders in τ voorkomen maar in dat geval wordt er niet meer van een wijzer als dusdanig gesproken.

[4]Het is op dit punt dat men oppert dat ciliaten het concept 'gelinkte lijst' van nature uit in zich dragen (zie [EHP+01]).

Het verband tussen micro-nucleaire genen en macro-nucleaire genen is dat deze laatste bekomen worden door het 'aan elkaar plakken' van overlappende MDSs. Men spreekt hier van de *gebruikelijke volgorde* $M_1, \ldots M_k$. In de micro-nucleaire genen zitten er IESs tussen de MDSs en deze laatste kunnen zelfs door elkaar zitten (dus in een gepermuteerde volgorde) en sommige ervan kunnen dan ook nog eens geïnverteerd zijn (bijvoorbeeld : $M_2 I_1 M_4 I_2 \overline{M}_3 I_3 M_1 I_4 \overline{M}_5$ waarbij I_i de IESs zijn).

Opmerking : om de notatie wat leesbaarder te maken, worden de wijzers als positieve gehele getallen geschreven (bijvoorbeeld $2, 3, \ldots, k-1$ en de inversies $\overline{2}, \overline{3}, \overline{k-1}$). De begin- en eindmarkeringen worden nog steeds genoteerd als b en e (en $\overline{b}$ en $\overline{e}$ voor hun inversies). Een verzameling wijzers wordt nu genoteerd als $\Pi_{k-1} = \{2, 3, \ldots, k-1, \overline{2}, \overline{3}, \ldots \overline{k-1}\}$. Een wijzer $p \in \Pi_{k-1}$ heeft als inversie $\overline{p}$ (waarbij de operatie $\overline{X}$ idempotent is). De verzameling $\{p, \overline{p}\}$ wordt ook wel de wijzer-verzameling (*pointer set*) van p genoemd en dit wordt genoteerd met behulp van **pts**(p) (en **pts**$(\overline{p})$).

Het gevolg is dat de MDSs nu eenvoudiger kunnen geschreven worden : $M_1 = (b, \mu_1, 2), M_k = (k, \mu_k, e)$ en $M_i = (i, \mu_i, i+1)$ voor $2 \leq i \leq k-1$. De inversie van $M = (p, \mu, q)$ wordt genoteerd als $\overline{M} = (\overline{q}, \overline{\mu}, \overline{p})$.

Men zegt dat een wijzer p in τ *voorkomt* als hij óf de inkomende óf de uitgaande wijzer van een MDS van τ is. Men spreekt van twee *naaste voorkomens* van een wijzer p in τ als hij twee voorkomens in τ heeft die enkel door een IESs worden gescheiden, of als hij twee voorkomens op de uiteinden van de molecule τ heeft (één aan elke zijde). Twee wijzers p en q *overlappen* elkaar in τ indien ze elk twee voorkomens in τ hebben met juist één voorkomen van p tussen de twee voorkomens van q (of vice versa).

Tenslotte zegt men dat τ een *direct herhaald patroon* (*direct repeat pattern*) (p, p) van een wijzer p heeft indien p twee voorkomens in τ heeft. Men spreekt van een *geïnverteerd direct herhaald patroon* (*inverted repeat pattern*) $(p, \overline{p})$ van een wijzer p indien zowel p als de inversie $\overline{p}$ ervan een enkele keer voorkomen in τ én p voor $\overline{p}$ komt. Men spreekt van een *afwisselend direct herhaald patroon* (*alternating direct repeat pattern*) van de wijzers (p, q) indien p en q in τ overlappen en het eerste voorkomen van p voor het eerste voorkomen van q komt.

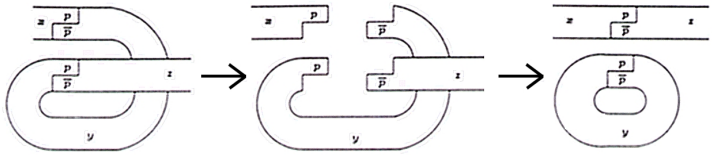
2.4 Moleculaire operaties

Gedurende de gen-assemblage worden de IESs dus verwijderd en de MDSs met elkaar verbonden in de gebruikelijke volgorde. Dit proces kan beschreven worden door middel van drie moleculaire operaties en het concept van de wijzers speelt hierin een belangrijke rol. De drie operaties worden *ld, hi* en *dlad* gedoopt en ze baseren zich alledrie op het maken van snedes in de moleculen.

2.4.1 ld-excision (*ld*)

De benaming *ld-excision* is afgeleid van *(loop, direct repeat)-excision*. Deze operatie *ld* wordt toegepast op moleculen die een direct herhaald patroon (p, p) van een wijzer p hebben. Een

dergelijke molecule wordt in een lus gevouwen (uitgelijnd door de directe herhaling). In figuur 8 wordt geïllustreerd wat er juist gebeurt : het afsnijden (*excision*) gebeurt via zogenaamde trapsgewijze snedes (*staggered cuts*) die kleverige eindes (*sticky ends*) opleveren. Het resultaat bestaat uit twee moleculen : een lineaire en een circulaire waarbij één ervan uit enkel IES bestaat (weliswaar met de aanwezigheid van een enkele kopie van de wijzer p) en waarbij in de andere een groter samengesteld MDS wordt gevormd. Het invoegen en verwijderen van kleine circulaire DNA-strengen bij lange lineaire DNA-strengen is een fenomeen dat in de natuur vrij vaak optreedt en het wordt dan ook als een aantrekkelijk paradigma aanzien om biomoleculaire berekeningen mee te doen (zie [DKGS99]).

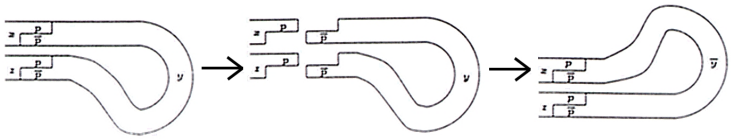


Figuur 8: De operatie 'ld-excision' (*ld*).

In paragraaf 2.4 werd reeds vermeld dat de drie operaties enkel snedes verrichten. Hieruit volgt dat er dus geen moleculen worden samengevoegd[5]. Een geslaagde gen-assemblage wordt slechts bereikt indien de door *ld* afgesneden circulaire moleculen geen MDS bevatten (en dus enkel uit een IES bestaan), of indien ze het gehele gen in micro-nucleaire of intermediaire vorm bevatten.

2.4.2 hi-excision/reinsertion (*hi*)

De benaming *hi-excision/reinsertion* is afgeleid van *(hairpin, inverted repeat)-excision/reinsertion*. Deze operatie is toepasbaar indien moleculen een geïnverteerd herhaald patroon $(p, \overline{p})$ van een pointer p bevatten. Dergelijke moleculen zijn in een haarspeld gevouwen. Ook hier worden, zoals in *ld*, trapsgewijze snedes gemaakt, maar het verschil is nu dat er herinvoeging (*reinsertion*) optreedt die resulteert in een enkele molecule. Een groter samengesteld MDS en een groter samengesteld IES worden nu gevormd (zie figuur 9).

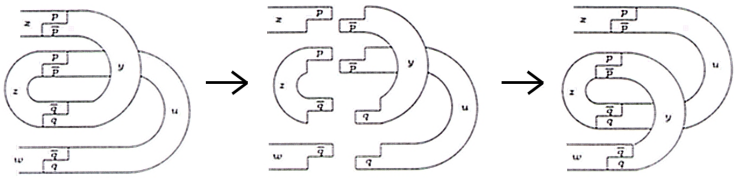


Figuur 9: De operatie 'hi-excision/reinsertion' (*hi*).

[5] Men spreekt dan ook wel van *intra-moleculaire* bewerkingen.

2.4.3 dlad-excision/reinsertion (*dlad*)

De benaming *dlad-excision/reinsertion* is afgeleid van *(double loop, alternating direct repeat)-excision/reinsertion*. Deze operatie wordt toegepast op moleculen die een afwisselend direct herhaald patroon van een koppel pointers (p, q) bevatten. Dergelijke moleculen worden nu in twee lussen gevouwen en er worden weerom trapsgewijze snedes gemaakt die na herinvoegen een enkele molecule opleveren (zie figuur 10).



Figuur 10: De operatie 'dlad-excision/reinsertion' (*dlad*).

2.5 Voorstelling MDS structuren

Gen-assemblage maakt gebruik van de in paragrafen 2.4.1, 2.4.2 en 2.4.3 besproken moleculaire operaties die ervoor zorgen dat de MDSs in de gebruikelijke volgorde $M_1, M_2, \ldots, M_k$ worden gezet. De structurele informatie over een gen kan beschreven worden door de volgorde van de MDSs alleen.

Het is dan ook belangrijk om een geschikte voorstelling van de MDS-structuur te geven voor de formalisatie van het proces van de gen-assemblage in ciliaten. In dit deel wordt hier kort op ingegaan.

2.5.1 Algemeen

De MDSs van een zeker gen worden genoteerd met het eindige alfabet $\mathcal{M}_k = \{M_{i,j} \mid i, j \in \mathbb{N}, i \leq j \leq k\}$. Men noemt de $M_{i,i}$ *elementair* en de $M_{i,j}$ *samengesteld* (met $i < j$). Een letter $M_{i,j}$, met $i < j$, wordt gebruikt om het samengestelde MDS voor te stellen dat ontstaat na samenvoeging (*aan elkaar plakken*) van de elementaire MDSs $M_i, M_{i+1}, \ldots, M_j$ door middel van hun wijzers. De inversies van de MDSs worden genoteerd met het alfabet $\overline{\mathcal{M}}_k = \{\overline{M}_{i,j} \mid M_{i,j} \in \mathcal{M}_k\}$. Stel $\Theta_k = \mathcal{M}_k \bigcup \overline{\mathcal{M}}_k$.

Een sequentie uit Θ_k wordt een *echte MDS structuur* (*real MDS structure*) genoemd indien ze kan gevormd worden door het permuteren van een sequentie in de gebruikelijke volgorde (met mogelijke inversies van een aantal elementen ervan).

2.5.2 Werkwijze

De bedoeling is nu om tijdens de gen-assemblage *enkel* de wijzers (en markeringen) bij te houden. De wijzers worden gebruikt om de vouwen – die in elk van de drie operaties ontstaan – juist uit te lijnen. Het proces van de gen-assemblage blijft nu doorgaan zolang er nog wijzers in de molecule aanwezig zijn (dus zolang er nog IESs aanwezig zijn die MDSs scheiden). Op het moment dat er nu geen IESs meer aanwezig zijn, zijn ook alle wijzers 'verdwenen'. Dit komt overeen met een macro-nucleair gen en het gen-assemblage proces is dan afgelopen.

Een verdere verfijning van de notatie, gebruikt op het einde van paragraaf 2.3, wordt nog doorgevoerd : een MDS $M = (p, \mu, q)$ wordt nu enkel genoteerd met behulp van zijn wijzers, dus als $M = (p, q)$. Er wordt nu een mapping ψ op Θ_k als volgt gedefinieerd :

- $\psi(M_{1,k}) = (b, e)$ en $\psi(\overline{M}_{1,k}) = (\overline{e}, \overline{b})$
- $\psi(M_{1,i}) = (b, i+1)$ en $\psi(\overline{M}_{1,i}) = (\overline{i+1}, \overline{b})$ $(\forall i \in \{1, \ldots, k-1\})$
- $\psi(M_{i,k}) = (i, e)$ en $\psi(\overline{M}_{i,k}) = (\overline{e}, \overline{i})$ $(\forall i \in \{2, \ldots, k\})$
- $\psi(M_{i,j}) = (i, j+1)$ en $\psi(\overline{M}_{i,j}) = (\overline{j+1}, \overline{i})$ $(\forall i, j \in \{2, \ldots, k-1\}$ met $i \leq j)$

Verder geldt nog dat $\psi(X_1 \ldots X_l) = \psi(X_1) \ldots \psi(X_l)$. Stellen we nu het uitgebreide alfabet voor door $\Pi_{ex,k}$ (dit is het alfabet uit paragraaf 2.3 aangevuld met $\{b, e, \overline{b}, \overline{e}\}$). De verzameling van alle geordende paren over $\Pi_{ex,k}$ noteren we met Γ_k en een string over deze verzameling wordt een *MDS beschrijving* (*MDS descriptor*) genoemd. Een MDS beschrijving wordt *realistisch* (*realistic*) genoemd indien voor een zekere echte MDS structuur x (over Θ_k, zie paragraaf 2.5.1) geldt dat $\delta = \psi(x)$.

Het formaliseren van de operaties tijdens de moleculaire gen-assemblage, gebeurt nu door het formaliseren van operaties op realistische MDS beschrijvingen. Er worden dan de regels **ld**, **hi** en **dlad** gebruikt (aangevuld met een hulpoperatie **rs** (*reversed switch*) die de inversie van een realistische MDS beschrijving vormt).

2.6 Reductiesystemen

Het omzetten van micro-nuclei naar macro-nuclei (en dus de gen-assemblage) is een proces dat op verschillende manieren formeel beschreven kan worden. Een populaire manier van werken is gebruik te maken van *reductiesystemen*. Uitgaande van een zekere beginsituatie, worden er een eindig aantal regels toegepast die zullen leiden tot een zekere eindsituatie. Het doel is om de gen-assemblage volledig te beschrijven met zo weinig mogelijk operaties (regels) op een zo efficiënt mogelijke manier. In dit deel worden kort twee mogelijke systemen toegelicht : een systeem gebaseerd op *strings* en een systeem gebaseerd op *grafen*.

2.6.1 String pointer reduction system (SPRS)

Inleiding In de voorgaande paragraaf werd het proces van de gen-assemblage beschreven door middel van realistische MDS beschrijvingen. Het formalisme steunde echter hevig op het gebruik van wijzers, markeringen en haakjes om de structuur van een MDS ten allen tijde te kunnen beschrijven. In deze paragraaf wordt dit model vereenvoudigd doordat er nu enkel nog maar strings van wijzers gebruikt worden. Door toepassing van een aantal regels op deze strings, kunnen deze laatste herleid worden (wat bijvoorbeeld resulteert in de gezochte gebruikelijke volgorde van MDSs).

Formele voorstelling Een *geldige string* over Π_k (zie paragraaf 2.3) is een string $\pi \in \Pi_k^*$ zodat voor elke $p \in \Pi_k$ geldt dat als π een enkel voorkomen uit $\mathbf{pts}(p)$ heeft, dan heeft π *juist* twee voorkomens uit $\mathbf{pts}(p)$.

Een realistische MDS beschrijving $\delta = (p_1, q_1) \ldots (p_m, q_m)$ (zie paragraaf 2.5.2) wordt genoteerd met behulp van een geldige string π_δ door enkel de sequentie van de wijzers neer te schrijven en de haakjes weg te laten (of nog : $\phi(\delta) = \pi_\delta$).

Er wordt nu een formeel systeem (een abstractie van het proces van de gen-assemblage) door **SPRS** gedefinieerd, waarin geldige strings over Π_k worden gereduceerd en waarbij elke reductie alle voorkomens van ofwel één of twee wijzer-verzamelingen verwijdert. Dit geeft aanleiding tot een herdefiniëring van de operaties **ld**, **hi** en **dlad**. Er wordt gebruik gemaakt van enkele hulp-operaties (**snr** (*string negative rule*), **spr** (*string positive rule*) en **sdr** (*string double rule*)).

Conclusie Als reductiesysteem slaagt **SPRS** er bijzonder goed in om op eenvoudige en efficiënte wijze het proces van gen-assemblage te beschrijven. Men kan aantonen dat dit systeem volledig equivalent is met het in paragraaf 2.5.2 gebruikte systeem van realistische MDS beschrijvingen. Het feit dat een biomoleculaire gen-assemblage in levende organismen (de ciliaten) volledig formeel beschreven kan worden, toont ondere andere aan dat DNA computing – vanuit computationeel standpunt gezien – bijzonder interessant is.

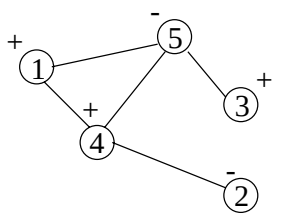
2.6.2 Graph pointer reduction system (GPRS)

Inleiding Equivalent met **SPRS** kan men nu *grafen* in plaats van *strings* gebruiken. De kunst bestaat erin om een zekere geldige string π in een graf γ_π om te zetten (waarbij de structuur van de wijzers door een geldige string wordt voorgesteld). Een wijzer-verzameling wordt nu voorgesteld door een knoop in de graf. Het reduceren van deze grafen (id est, het toepassen van operaties op deze grafen) heeft als gevolg dat elke reductie-stap overeenkomt met het verwijderen van óf een enkele knoop (dit in het geval het de operatie **ld** of **hi** betreft), óf twee knopen (wanneer het de operatie **dlad** betreft).

Formele voorstelling Een graf $\gamma = (V, E)$ bestaat uit een verzameling V van knopen (*vertices*) en een verzameling E van ongerichte verbindingen (*undirected edges*) die geen lussen zijn. Merk op dat er geen meervoudige verbindingen tussen twee knopen worden toegestaan. Men noteert deze verzamelingen ook wel als V_γ en E_γ.

Indien er een verbinding tussen twee knopen x en y bestaat dan worden ze *naburig* (*adjacent*) genoemd. De *omgeving* (*neighbourhood*) van een een knoop x in een graf γ bestaat uit alle knopen y in γ die ermee naburig zijn. De *gesloten omgeving* (*closed neighbourhood*) van een knoop x is de omgeving van x met x eraan toegevoegd. Een knoop x is *geïsoleerd* indien zijn gesloten omgeving $\{x\}$ zelf is.

Een *gesigneerde graf* (*signed graph*) $\gamma = (V, E, \theta)$ bestaat uit een graf $\gamma = (V, E)$, aangevuld met een *teken-functie* $\theta : V \rightarrow \{+, -\}$ die aan elke node een teken toekent (een node wordt dan *positief* of *negatief* genoemd). Figuur 11 illustreert dit concept.



Figuur 11: Een voorbeeld van een gesigneerde graf (*signed graph*).

Indien Δ nu een alfabet is en δ een string over Δ is, dan noemt men δ een *dubbel-voorkomen string* (*double occurrence string*) indien elke letter van Δ ofwel juist twee keer in δ voorkomt, ofwel helemaal niet voorkomt. Op basis hiervan kan de *overlap graf* van een dubbel-voorkomen string geconstrueerd worden. Deze overlap graf wordt intensief gebruikt bij het definiëren van de operaties **ld**, **hi** en **dlad**. Merk op dat in het formeel bewijzen men ook nog gebruik maakt van volgende hulp-operaties : **gir** (*local complementation*), **gnr** (*graph negative rule*), **gpr** (*graph positive rule*) en **gdr** (*graph double rule*).

Conclusie Het gebruik van **GPRS** is volledig equivalent met dat van **SPRS**, juist omdat aangetoond kan worden dat elke string in een graf kan omgezet worden (met de aangepaste versies voor de reductie-operaties). Gen-assemblage kan dus succesvol op verschillende (equivalente) manieren beschreven worden (en het gebruik van grafen heeft als voordeel dat het nog abstracter dan het gebruik van strings is).

2.7 Conclusie

Het proces van gen-assemblage in ciliaten observerend, heeft men getracht dit vanuit een computationeel standpunt te benaderen. Het verwijderen van IESs en aan elkaar plakken van MDSs kan beschreven worden met behulp van opeenvolgende moleculaire operaties (zie paragraaf 2.4). Deze operaties zijn *ld*, *hi* en *dlad* en samen vormen zij een *volledige* verzameling waarmee het

proces van DNA-verwerking beschreven kan worden. Allen baseren zij zich op het concept ‘wijzer’ dat impliciet in de ciliaten schijnt ingebakken te zitten.

In een poging tot steeds verdere abstractie, werd vooreerst een formeel systeem opgesteld (de realistische MDS beschrijvingen). Uitgaande van deze manier van werken, werden twee – meer abstracte – reductiesystemen uitgewerkt die zich enerzijds baseren op strings en anderzijds op grafen. Beide systemen zijn volledig equivalent (er gaat dus geen essentiële informatie verloren), al is het gebruik van grafen iets meer abstract dan het gebruik van strings. Ze bieden een uniforme verklaring voor het biologisch proces van DNA-verwerking en zijn als formeel systeem zeer handig om te redeneren.

3 DNA computing op korte en lange termijn

Het grote succes van DNA computing is mede te danken aan het feit dat genetische systemen in bestaande organismen op formele wijze beschreven kunnen worden. Computationeel gezien leent dit zich onder andere tot het oplossen van bepaalde problemen (zoals het Hamiltoniaans Pad Probleem). Verder kan DNA computing een leemte opvullen die gevormd zal worden vermits de op silicium gebaseerde computers ergens een einde qua miniaturisatie zullen kennen en zij beperkt zijn tot seriële verwerking. *Massaal parallellisme* is een groot voordeel van DNA computing om onder andere NP-harde problemen op te lossen.

Men mag evenwel niet uit het oog verliezen dat alhoewel op korte termijn er grote successen geboekt werden op het vlak van DNA computing, er nog veel werk nodig is om op lange termijn een concrete *DNA computer* te produceren. Daarenboven dient men ook in het achterhoofd te houden dat er een zekere ‘concurrentie’ is vanuit een ander – schijnbaar hiermee ongecorreleerd – domein dat zich baseert op de *quantummechanica*. Quantumcomputers beloven ook een serieuze kandidaat te zijn om bijvoorbeeld NP-harde problemen op efficiënte en snelle wijze op te lossen. DNA computing steunt in essentie nog steeds op klassiek rekenwerk, met dat verschil dat het massaal parallel kan gebeuren. Quantum computing daarentegen steunt op algoritmes die in polynomiale tijd exponentiële problemen kunnen oplossen (zie [Lab00]).

DNA computing kan weliswaar directe voordelen opleveren voor de medische gemeenschap (iets wat bij quantum computing niet zo voor de hand liggend is). Het is af te wegen wat de uiteindelijke voor- en nadelen van beide technieken (DNA computing en quantum computing) zullen zijn, alvorens een definitief oordeel te vellen over de inzetbaarheid ervan. Hoogstwaarschijnlijk zullen beiden naast elkaar bestaan en elkaar complementair aanvullen.

Referenties

[dA01] Dirk Van den Abeele. *Genetica bij Agaporniden*. Webextract, 2001. (URL : `http://home.worldonline.be/~vdadirk/mut/genetica.htm`).

[DKGS99] Mark Daley, Lila Kari, Greg Gloor, en Rani Siromoney. Circular contextual insertions/deletions with applications to biomolecular computation. Uit *Proceedings of SPIRE'99 - String Processing and Information REtrieval, Cancun, Mexico*. IEEE Press, september 1999.

[EHP+01] Andrzej Ehrenfeucht, Tero Harju, Ion Petre, David M. Prescott, en Gregorz Rozenberg. Formal systems for gene assembly in ciliates. *Theoretical Computer Science*, februari 2001. Ongepubliceerde voordruk.

[HR01] H.J. Hoogeboom en Gregorz Rozenberg. *DNA Computing*, 2001.

[Kar97] Lila Kari. DNA computing : arrival of biological mathematics. *The Mathematical Intelligencer*, 19(2):9–22, 1997.

[KKG00] Lila Kari, Rob Kitto, en Greg Gloor. *A Computer Scientist's Guide to Molecular Biology*, 2000.

[KKL99] Jarkko Kari, Lila Kari, en Laura F. Landweber. Reversible Molecular Computation in Ciliates. Uit *Jewels are Forever*, pagina's 353–363. Springer-Verlag, 1999.

[KL00] Lila Kari en Laura F. Landweber. Computational Power of Gene Rearrangement. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pagina's 203–212, 2000.

[Lab00] RSA Laboratories. *RSA Laboratories' Frequently Asked Questions About Today's Cryptography, Version 4.1*. Webextract – RSA Security Inc., 2000. (URL : `http://www.rsasecurity.com/rsalabs/faq`).

[Pre99] David M. Prescott. *Molecular Cellular and Developmental Biology*. Webextract – University of Colorado, 1999. (URL : `http://mcdb.colorado.edu/faculty/prescott99.html`).

[WS94] Ben Waggoner en Brian Speer. *Introduction to the Ciliata*. Webextract – The Museum of Paleontology of the University of California at Berkeley, the Regents of the University of California, 1994. (URL : `http://www.ucmp.berkeley.edu/protista/ciliata.html`).