Algoritmen voor identificatie van peptiden in massaspectrometrie

Suzanna CHA

promotor :
Prof. dr. Jan VAN DEN BUSSCHE

Eindverhandeling voorgedragen tot het bekomen van de graad
Licentiaat in de informatica
afstudeervariant multimedia

Universiteit Maastricht

universiteit hasselt

Algoritmen voor identificatie van peptiden in massaspectrometrie

Thesis voorgedragen tot het behalen van de graad van
master in de informatica/ICT/kennistechnologie,
afstudeervariant Informatica-Multimedia

Cha Suzanna

Promotor : Prof. dr. Jan van den Bussche
Begeleidster : Natalia Kwasnikowska

Academiejaar 2005-2006

School voor Informatietechnologie
transnationale Universiteit Limburg

Abstract

De identificatie van proteïnen is vaak een belangrijke fase in medische onderzoeken. Helaas is deze identificatie geen voor de hand liggende opdracht. De laatste jaren heeft men hiervoor steeds meer beroep gedaan op de informatica. Talloze algoritmen zijn ontwikkeld om aan de vraag te voldoen. Deze algoritmen kunnen onderverdeeld worden in twee grote groepen: de methoden die gebruik maken van een proteïnedatabase en de zogenaamde *de novo* algoritmen. Het zijn deze laatste die het eigenlijke onderwerp vormen van deze thesis. Deze algoritmen beperken zich meestal tot de identificatie van een peptide, wat een stukje van een proteïne is.
Deel I van de tekst bevat twee inleidende hoofdstukken. Het eerste hoofdstuk verklaart begrippen uit de moleculaire biologie en beschrijft technieken die gebruikt worden in de voorbereiding tot de identificatie van peptiden. Het tweede hoofdstuk is een inleiding op de algoritmen voor de identificatie van peptiden- en/of proteïnen.
Deel II beschrijft drie *de novo* algoritmen. Een eerste algoritme, Sherenga genaamd, maakt hiervoor gebruik van een graaf. De problemen die hiermee gepaard gaan leiden tot niet voor de hand liggende padalgoritmen die in hoofdstuk 4 aan bod zullen komen. Hoofdstuk 5 bespreekt een algoritme dat gebruik maakt van een heel ander aspect uit de theoretische informatica : de automaten. Het derde algoritme dat besproken wordt is, net als Sherenga, ook gebaseerd op de grafentheorie. We zullen zien dat dit algoritme de problemen die resulteren uit de biologische gegevens op een heel andere manier aanpakt. Speciaal aan dit algoritme is de integratie van een database zoekalgoritme, waardoor het resultaat van het algoritme niet beperkt blijft tot de identificatie van de peptide, maar verder gaat zoeken naar de proteïne waar deze peptide uit afkomstig is.

Voorwoord

De combinatie van biologie en informatica trok me sterk aan in dit thesisonderwerp. Het bracht heel wat bijkomende studie met zich mee over moleculaire biologie en massaspectrometrie, maar hiervoor kon ik rekenen op de steun van het Biomedisch Onderzoekscentrum Biomed op de campus. Vanaf het moment dat de studie zich begon te concentreren op het echte werk, de algoritmen, bleek al gauw dat dit niet van een leien dakje zou lopen. De literatuur over peptide identificatie in het algemeen is zeer uitgebreid, maar de literatuur over de specifieke algoritmen bleek zeer beperkt te zijn, waardoor het verre van eenvoudig was om de precieze werkwijze tot in detail uit te pluizen. Dit heeft verschillende oorzaken. Bioinformatica is een relatief nieuwe 'wetenschap' en is in volle ontwikkeling. De gebruikte literatuur is dan ook voor het grootste deel zeer recent gedateerd waardoor resultaten van verdergezet onderzoek nog niet voorhanden zijn. Omdat de identificatie van peptiden ook nog lang niet op punt staat worden er vooral nieuwe algoritmen ontwikkeld in plaats van bestaande algoritmen te optimaliseren, en blijft een verduidelijking bij een eerste publicatie veelal uit. Bovendien willen velen de details van hun algoritme ook niet prijs geven omwille van de commerciële waarde ervan.
Het gebruik van verschillende symbolen en benamingen in de talrijke literatuur over de identificatie van peptiden maakte het er niet eenvoudiger op. In deze thesistekst hebben we daarom eenduidige notaties en definities aangehouden doorheen de hele tekst.
Tijdens het tot stand brengen van deze thesis heb ik meermaals kunnen terugvallen op de theorie uit de opleiding, vooral begrippen uit de theoretische informatica en de statistiek.

Graag zou ik enkele mensen willen bedanken die zich mee ingezet hebben om dit werk tot een goed einde te brengen :
Promotor Prof. Dr. Jan van den Bussche en begeleidster Natalia Kwasnikowska voor de opvolging en begeleiding van de thesis.
Jean-Paul Noben en Debora Dumont van Biomed voor hun deskundige uitleg.
Mijn ouders en zus voor de steun en het geduld, niet enkel tijdens het verwezenlijken van deze tekst, maar gedurende de hele duur van mijn studies.
Mijn vrienden en zus voor de ontspannende momenten.
In het bijzonder : Bedankt mama en Mieke voor het vele nalees- en verbeterwerk.
Last but not least, special thanks to Albert Sickmann for the help on theoretical spectra.

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

Deel I

Inleiding

Hoofdstuk 1

Inleiding

Het onderwerp van deze thesis wordt gesitueerd in het domein van de bioinformatica, en er zijn vanzelfsprekend veel termen uit de moleculaire biologie aan bod gekomen. Een voorbereidende studie van de basisconcepten hiervan was uiteraard onontbeerlijk. Dit eerste hoofdstuk geeft een korte inleiding tot de basis van de moleculaire biologie, alsook de nodige kennis over *proteomica*[1] (sectie 1.2), als aanloop naar het eigenlijke onderwerp van de thesis : "algoritmen voor de identificatie van peptiden".

1.1 Basis van de moleculaire biologie

Deze inleiding is uiteraard zeer beperkt. Voor een meer gedetailleerde inleiding verwijzen we naar "The Cartoon Guide to Genetics" [1], "Introduction to Computational Molecular Biology" [2] en naar de referenties die in de tekst opgenomen zijn. Bibliografie waar niet naar verwezen wordt in de tekst is ook onderdeel van de literatuurstudie, maar draagt meer bij tot het geheel dan tot één bepaald onderdeel. Het Glossarium in bijlage A geeft een alfabetisch overzicht van de gebruikte biologische termen. Deze worden de eerste keer dat ze voorkomen in de tekst cursief weergegeven.

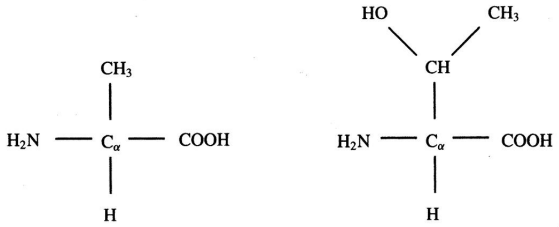
1.1.1 De biochemie van het leven

In de natuur komen zowel levende als niet-levende entiteiten voor. Levende entiteiten onderscheiden zich van de niet-levenden door hun actieve deelname in, en door het uitwisselen van energie met hun omgeving. De manier waarop dit gebeurt hangt af van een complexe opeenvolging van chemische reacties die plaats vinden binnenin het levend organisme. Zowel de zeer eenvoudige als de zeer complexe organismen hebben allen dezelfde moleculaire biologie, ook wel biochemie genoemd. De twee belangrijkste actoren in deze "chemistry of life" zijn *proteïnen* en *nucleïnezuren*.

Proteïnen

Proteïnen zijn verantwoordelijk voor wat een levend organisme is en doet, in de fysieke betekenis. Enkele voorbeelden van de vele verschillende soorten proteïnen die er bestaan zijn structurele proteïnen die te werk gaan als weefselbouwstenen, en *enzymen* die optreden als *katalysators* voor chemische reacties. Ook zuurstoftransport en *antigen*-detectie zijn voorbeelden van proteïnefuncties. Voor we het eigenlijke onderwerp van deze thesis kunnen aanvatten, is het noodzakelijk de structuur van een proteïne te kennen.
Een proteïne is een ketting van moleculen die *aminozuren* genoemd worden. Deze aminozuren hebben één centraal koolstof*atoom*, ook wel gekend als het *alpha carbon* of C_α. Hieraan zijn een waterstofatoom (H), een aminogroep (H_2N) en een koolstofgroep ($COOH$) toegevoegd. Verder is er ook nog een zogenaamde **side chain** die aan het C_α atoom vasthangt. Het is deze side chain die het ene aminozuur van het andere onderscheidt. Enkele voorbeelden van aminozuren

[1] Engelse term: proteomics



Figuur 1.1: Enkele aminozuren : alanine (links) en threonine [2]

	One-letter code	*Three-letter code*	*Name*
1	A	Ala	Alanine
2	C	Cys	Cysteine
3	D	Asp	Aspartic Acid
4	E	Glu	Glutamic Acid
5	F	Phe	Phenylalanine
6	G	Gly	Glycine
7	H	His	Histidine
8	I	Ile	Isoleucine
9	K	Lys	Lysine
10	L	Leu	Leucine
11	M	Met	Methionine
12	N	Asn	Asparagine
13	P	Pro	Proline
14	Q	Gln	Glutamine
15	R	Arg	Arginine
16	S	Ser	Serine
17	T	Thr	Threonine
18	V	Val	Valine
19	W	Trp	Tryptophan
20	Y	Tyr	Tyrosine

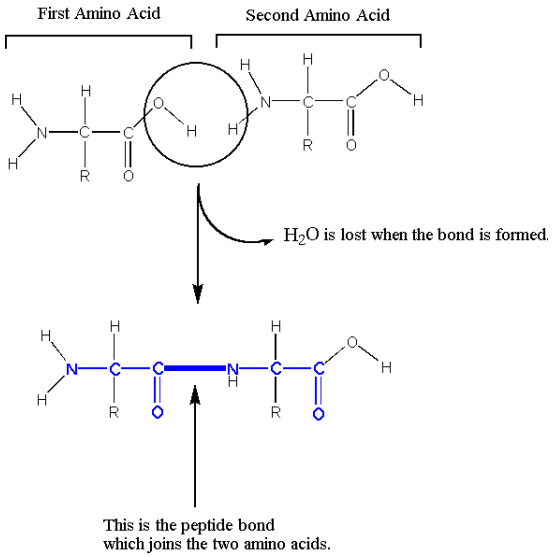
Figuur 1.2: Overzicht van de aminozuren [2]

zijn te zien in figuur 1.1. In de natuur zijn er zo twintig verschillende aminozuren gevonden die de bouwstenen vormen van proteïnen. Figuur 1.2 geeft hiervan een overzicht. Een deel van zulk een aminozurenketting wordt een *peptide* genoemd. Het is de identificatie van deze peptiden die het onderwerp van deze thesis vormt. Peptiden komen echter niet voor in organismen maar worden *in vitro* bekomen door een proteïne op te delen (sectie 1.2). De structurele en functionele informatie in deze eerste sectie heeft dan ook betrekking op de oorspronkelijke proteïnen.
De aminozuren die deel uitmaken van een proteïne zijn aan mekaar gehecht via *peptidebindingen.* Proteïnen worden daarom ook wel *polypeptide-ketens*[2] genoemd. Tijdens een peptidebinding wordt een watermolecule (H_2O) vrijgegeven, waardoor de polypeptide-keten in feite geen ketting van aminozuren is, maar een ketting van residu's van de originele aminozuren. Een schematische voorstelling van een peptidebinding is te zien in figuur 1.3. Het symbool R representeert een willekeurige side chain. Figuur 1.4 geeft de structuur weer van een proteïne. Hierin is het duidelijk zichtbaar dat, door de peptidebindingen, iedere proteïne een **backbone** heeft : $(-N-C_\alpha-(CO)-)$. Deze is zowel in figuur 1.3 als in figuur 1.4 in het blauw en vet weergegeven. De polypeptide-keten heeft ook een richting die begint bij de amino groep (de *N-terminal*) en eindigt aan de koolstof-groep (de *C-terminal*) (in de figuur van links naar rechts). Dit alles wordt ook wel de **primaire structuur** genoemd.
Proteïnen wegen typisch tussen de 30 en de 100 *kDa.* De tabel in bijlage C geeft een overzicht van de massa's van de aminozurenresidu's. Zo krijgt de lezer een idee van de enorme lengte[3] van een proteïne.

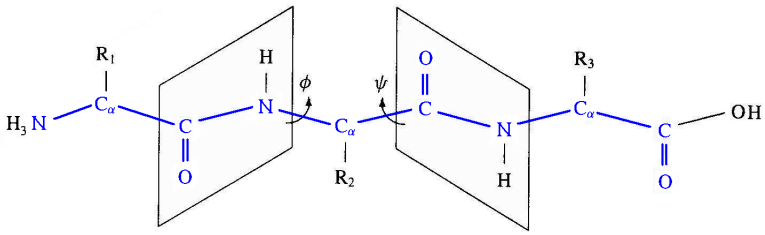
De secundaire, tertiaire en quaternaire structuur van een proteïne duiden op de vorm ervan. Deze is gerelateerd aan zijn functie en is vaak zeer complex en zonder symmetrie. Deze vorm kan daarom

[2] Engelse term: polypeptidic chains.
[3] In aantal aminozuren.

ook niet op een eenvoudige manier bepaald worden, maar het is, in de context van deze thesistekst, niet nodig om hier dieper op in te gaan. Een meer gedetailleerde uiteenzetting over de peptidebinding, de manier waarop de vorm van een proteïne zijn functie bepaalt, en andere details kunnen teruggevonden worden in [2].



Figuur 1.3: Peptidebinding [3]



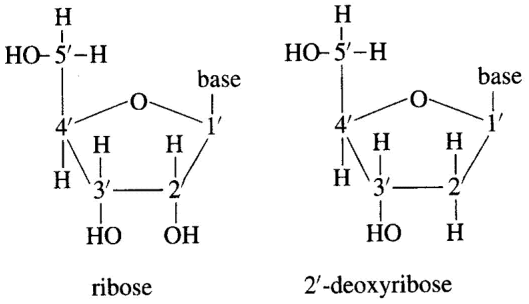
Figuur 1.4: Polypeptide-keten [2]

Nucleïnezuren

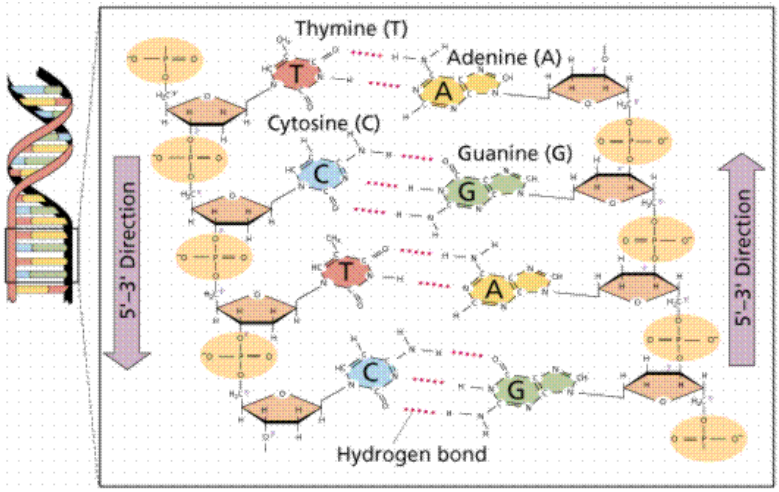
Nucleïnezuren coderen de informatie die nodig is om proteïnen aan te maken. Er zijn twee verschillende soorten nucleïnezuren; desoxyribonucleïnezuur[4] of *DNA*, en ribonucleïnezuur[5] of *RNA*.

- **DNA : desoxyribonucleïnezuur**
  Een DNA molecule is een **dubbele** ketting van eenvoudige moleculen. Eén enkele ketting wordt een **streng** genoemd. Zulk een streng heeft een backbone bestaande uit herhalingen van dezelfde basiseenheid. Deze basiseenheid wordt gevormd door een suikermolecule (desoxyribose) en een fosfaatresidu. De suikermolecule bevat vijf koolstofatomen die van 1' tot en met 5' gelabeld worden, zoals te zien is in figuur 1.5 (a). De backbone wordt gecreëerd door de binding tussen de 3' koolstof van een unit, het fosfaatresidu (niet op tekening),

[4] Engelse term: deoxyribonucleic acid.
[5] Engelse term: ribonucleic acid.



Figuur 1.5: Suikermolecule in DNA (a) en in RNA (b) [2]



Figuur 1.6: DNA [4]

en de 5' koolstof van de volgende eenheid. De oriëntatie van een DNA streng is als volgt gedefinieerd; beginnend bij het 5' einde, en eindigend aan het 3' einde. De twee strengen van een DNA molecule hebben een tegengestelde oriëntatie en worden daarom **antiparallel** genoemd.

De DNA-moleculen verschillen onderling in de base die aan de 1' koolstof van de basiseenheid vasthangt. Er zijn vier soorten bases : *adenine (A)*, *guanine (G)*, *cytosine (C)* en *thymine (T)*. Adenine en guanine behoren tot een grotere groep van substanties, *purines* genoemd. Cytosine en thymine behoren tot de groep van de *pyrimidines*. De basiseenheid samen met de onderscheidende base wordt een *nucleotide* genoemd. De twee strengen van een DNA molecule hechten aan mekaar omdat iedere base in de ene streng bindt met een base in de andere streng. Op die manier vormen ze een helix[6] die links in figuur 1.6 weergegeven wordt. Belangrijk hierbij is dat base A altijd met base T bindt, en base C met base G. Deze paren van complementaire basen worden ook Watson-Crick baseparen[7] genoemd, en worden verbonden door twee of drie *waterstofbruggen*. Het aantal base paren wordt vaak gebruikt als eenheid van lengte voor DNA-moleculen, afgekort tot *bp*. Een schematische voorstelling van een stukje DNA wordt rechts weergegeven in figuur 1.6.

Door de strikte bindingsregels tussen de basen is het, met één gegeven streng, mogelijk om de sequentie van de andere streng hieruit af te leiden. Dankzij dit is **replicatie** mogelijk, en kan

[6] Wenteltrapstructuur.
[7] Genoemd naar *James D. Watson* en *Francis Crick* die in 1953 de structuur van het DNA ontdekt hebben.

één enkele cel uitgroeien tot miljoenen andere cellen, die elk kopies van de DNA moleculen van de originele cel in zich dragen. De functie van DNA beperkt zich tot het coderen van informatie.

- **RNA : ribonucleïnezuur**
  Hoewel de basisstructuur hetzelfde is, zijn er toch enkele structurele verschillen tussen RNA en DNA. Een belangrijk verschil is het ontbreken van de base thymine (T) in RNA. In plaats daarvan vinden we een andere base, genaamd *uracil (U)*, die ook met adenine (A) bindt, net als thymine (T) doet in DNA. De suiker die voorkomt in RNA is ribose in plaats van desoxyribose in DNA. Figuur 1.5 (b) geeft de suiker weer die in RNA voorkomt. Verder vormt RNA geen dubbele helix en heeft het een drie-dimensionele structuur die veel gevarieerder is dan die van DNA. Ten slotte, in tegenstelling tot DNA, zijn er verschillende soorten RNA in een cel die allemaal een verschillende functie uitvoeren. In wat volgt zullen we ondermeer messenger RNA (mRNA), ribosomaal RNA(rRNA) en transfer RNA (tRNA) tegenkomen, drie soorten RNA elk met hun eigen specifieke functie.

1.1.2 Aanmaak van proteïnen

In deze sectie zal beschreven worden hoe proteïnen aangemaakt worden. Om dit proces te begrijpen is het nodig eerst wat kennis op te doen over o.a. chromosomen en genen, mRNA en ribosomen.

Genen en de genetische code

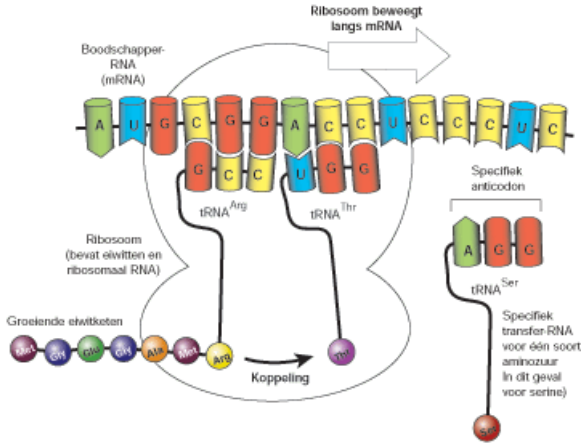
In een organisme heeft iedere cel een aantal zeer lange DNA moleculen. Zulke moleculen worden *chromosomen* genoemd. In een menselijke cel bijvoorbeeld komen 46 (23 paren) chromosomen voor. Chromosomen bestaan uit *genen*, aaneensluitende strengen DNA. Een gen codeert de samenstelling van één proteïne. Vermits een proteïne eigenlijk een aaneenschakeling van aminozuren is, zijn het juist die aminozuren die gekend moeten zijn voor de opbouw van een proteïne. Het DNA in een gen codeert deze informatie met behulp van *codons* of nucleotidetriplets, een opeenvolging van drie nucleotiden. Eén codon specificeert één aminozuur. De zogenaamde *genetische code* wordt weergegeven in de tabel in figuur 1.7. Hierin merken we op dat niet de DNA basen maar wel de RNA basen (dus met uracil i.p.v. thymine) gebruikt worden. De reden hiervoor is dat RNA moleculen de link vormen tussen het DNA en de proteïnesynthese (die zodadelijk aan bod komt). Ook valt het op dat er 64 mogelijke codons zijn, hoewel er maar 20 aminozuren gespecificeerd moeten worden. Hieruit volgt logischerwijs dat verschillende codons kunnen leiden tot hetzelfde aminozuur. In de tabel zien we ook meermaals het woord 'STOP' voorkomen. In de volgende sectie zal de rol van dit 'STOP'-codon duidelijk worden. Wat niet in deze tabel aangegeven is, is het 'START'-codon. Iedere proteïne begint met het aminozuur Methionine dat gecodeerd wordt door het codon AUG, wat daarom ook wel het 'START'-codon genoemd wordt.

Proteïnesynthese

Bij het eigenlijke proces van de *proteïnesynthese* komt veel meer kijken dan in deze korte uiteenzetting wordt vermeld, maar voor het verloop van deze thesistekst is begrip van het onderstaande voldoende. De eerste stap in de proteïnesynthese wordt **transcriptie** genoemd. Hierbij wordt er een kopie gemaakt van het nodige gen (dus het gen dat de gevraagde proteïne codeert) op een RNA molecule. Dit resulterend RNA is het *mRNA* of messenger RNA, en heeft exact dezelfde sequentie als één van de strengen van het gen, maar met base uracil i.p.v. thymine. Het kan voorkomen dat de genen *alternatieve splicing* ondergaan waardoor men een gewijzigde definitie van het gen krijgt, en daardoor ook ander mRNA. We zullen niet verder ingaan op de details maar onthouden wel dat dit zal leiden tot andere proteïnen op het einde van dit proces. Dit kunnen nieuwe, ongekende proteïnen zijn.
Vervolgens wordt dit mRNA in het *ribosoom* gebruikt om de proteïne aan te maken. Dit ribosoom is een celstructuur bestaande uit o.a. *rRNA* of ribosomaal RNA. In het ribosoom wordt, naast

First position	Second position G	A	C	U	Third position
G	Gly	Glu	Ala	Val	G
	Gly	Glu	Ala	Val	A
	Gly	Asp	Ala	Val	C
	Gly	Asp	Ala	Val	U
A	Arg	Lys	Thr	Met	G
	Arg	Lys	Thr	Ile	A
	Ser	Asn	Thr	Ile	C
	Ser	Asn	Thr	Ile	U
C	Arg	Gln	Pro	Leu	G
	Arg	Gln	Pro	Leu	A
	Arg	His	Pro	Leu	C
	Arg	His	Pro	Leu	U
U	Trp	STOP	Ser	Leu	G
	STOP	STOP	Ser	Leu	A
	Cys	Tyr	Ser	Phe	C
	Cys	Tyr	Ser	Phe	U

Figuur 1.7: Genetische code [2]



Figuur 1.8: De aanmaak van een proteïne binnen het ribosoom [5]

het mRNA dat als 'input' beschouwd wordt, ook *tRNA* of transfer RNA gebruikt. Dit soort RNA zijn de moleculen die het verband leggen tussen een codon en het aminozuur dat door deze codon gespecificeerd wordt. Dit proces wordt **translatie** genoemd. Wanneer het mRNA het ribosoom binnendringt, wordt er een tRNA aan gebonden dat overeenstemt met het huidige codon (dit is het codon dat zich binnen het ribosoom bevindt). Dit tRNA brengt het juiste aminozuur mee (dat in de cel 'rondzwemt'). Zodra het tRNA zich aan het codon gebonden heeft wordt het aminozuur losgelaten. Wanneer het volgende codon zich in het ribosoom bevindt wordt dit proces herhaald. Het laatst bijgekomen aminozuur wordt dan door middel van rRNA, dat als katalysator optreedt, aan het vorige aminozuur gebonden, waardoor een keten van aminozuren, oftewel de gevraagde proteïne ontstaat. Figuur 1.8 geeft schematisch weer hoe dit proces binnen het ribosoom verloopt. Zodra er een 'STOP'-codon het ribosoom binnendringt, wordt de aanmaak van de aminozurenketting beëindigd, en is de proteïne gevormd.
Na de translatie komt het vaak voor dat de aaneengeschakelde aminozuren van de proteïne andere moleculen aantrekken, zoals bijvoorbeeld zuurstofatomen. Dit fenomeen wordt *posttranslationele wijziging* genoemd. Hierdoor verandert zowel de massa als de lading van de proteïne. Bij de identificatie van peptiden moet er dus rekening gehouden worden met de mogelijkheid dat deze peptiden afkomstig kunnen zijn van gewijzigde proteïnen. Vermits men nooit zeker weet welke wijzigingen hebben plaatsgevonden, vormt dit een bijkomende moeilijkheid tijdens het identificatieproces.

1.2 Proteomica

Proteomica [6] is een relatief nieuwe term. Eerder biologisch onderzoek, *genomica*[8] genoemd [6], deed onderzoek naar het genoom van een organisme. Het *genoom* van een organisme is eigenlijk de volledige erfelijke informatie die dat organisme bij zich draagt. Deze informatie is, zoals wel bekend, gecodeerd door het DNA. Het is duidelijk dat een genoom een constant iets is, en daardoor relatief gemakkelijk te onderzoeken is. Het is echter gebleken dat er nog veel onbekenden bleven in het vakgebied, door enkel genomen te bestuderen. Uit sectie 1.1 is gebleken dat proteïnen een grote rol spelen in het leven. Daarom is men een stap verder gegaan na genomica. Om een analogie te creëren met de term genomica, werd deze volgende stap "proteomica" gedoopt. Proteomica bestudeert proteïnen op grote schaal, in het bijzonder hun structuren en functies. Een *proteoom* (naar analogie met de term genoom) kan gezien worden als de verzameling van proteïnen die in een bepaalde cel voorkomen, onder bepaalde omgevingsvoorwaarden. Proteomica bestudeert dus al deze proteïnen van een bepaald proteoom gelijktijdig, in plaats van ze elk apart te bestuderen. Hierdoor tracht men een meer globaal en geïntegreerd zicht te krijgen op de biologie. Een proteoom is, in tegenstelling tot het genoom, voortdurend aan het veranderen door zijn biochemische interacties met het genoom. Een organisme zal totaal verschillende proteïne-expressies hebben naargelang zijn weefsel- of celtype. Daardoor is proteomica een veel complexere studie dan genomica.
Het aspect van proteomica dat voor deze thesis van toepassing is, namelijk het identificeren van peptiden, zal in het vervolg van deze sectie toegelicht worden. Andere toepassingen van proteomica zullen hier niet aan bod komen, maar zijn voor de geïnteresseerde lezer terug te vinden in [6].

1.2.1 Inleiding

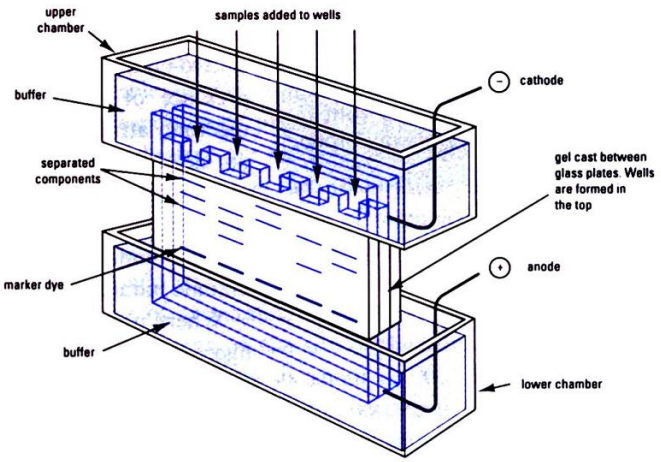
Uit het voorgaande is het duidelijk geworden dat proteïnen een zeer belangrijke rol spelen in het 'leven'. Onderzoek naar de werking van proteïnen is daarom een gebied van grote interesse. Niet zelden is een verstoorde werking van proteïnen de aanleiding tot een ziekte (bv. multiple sclerose) [7]. Door te achterhalen welke proteïnen onder welke condities (bv. gezond vs. ziek) aanwezig zijn, kan men trachten de oorzaak te vinden van deze ziektes. En als gevolg hiervan kan er ook gezocht worden naar remedies ertegen.
Wil men dit onderzoek voeren, dan moet er dus een manier bestaan om te achterhalen welke proteïnen er in een organisme voorkomen. Om dit proteoom te kunnen onderzoeken wordt hiervan een staal genomen. Het aantal proteïnen in een staal is afhankelijk van het species en het weefsel. Voor een gistcel bijvoorbeeld bevat een staal typisch enkele duizenden proteïnen (4000 à 5000). We zullen zien dat in werkelijkheid niet de proteïnen zelf geïdentificeerd worden, maar in eerste instantie wel de peptiden waaruit deze proteïnen bestaan. Na de identificatie van de peptiden kan men met de bekomen informatie de oorspronkelijke proteïnen identificeren met behulp van een analyse. Dit is echter geen onderdeel van deze thesis en we zullen ons dan ook voor het grootste deel beperken tot de identificatie van de peptiden.
Dit identificatieproces kan opgedeeld worden in drie grote delen, die in de volgende secties gedetailleerd aan bod zullen komen :

- scheiding van het staal : deze eerste stap gebeurt in het labo (***in vitro***),
- waarnemingen met de massaspectrometer : metingen die leiden tot MS/MS spectra,
- peptide-identificatie d.m.v. zoekalgoritmen : met behulp van de computer (***in silico***).

1.2.2 Scheiding van het staal

Het proteïnestaal is zo goed als altijd een complex mengsel van proteïnen. Vertrekkende hiervan zou het een bijna onmogelijke opdracht zijn om uiteindelijk in de identificatie te slagen. Daarom wordt het proteïnemengsel opgesplitst vooraleer de werkelijke gegevens worden ingewonnen over de samenstelling. Er bestaan verscheidene scheidingsmethoden om een complex proteïnemengsel op

[8] Engelse term: genomics.



Figuur 1.9: Gelconstructie waarin de proteïnenstalen ingebracht worden [8]

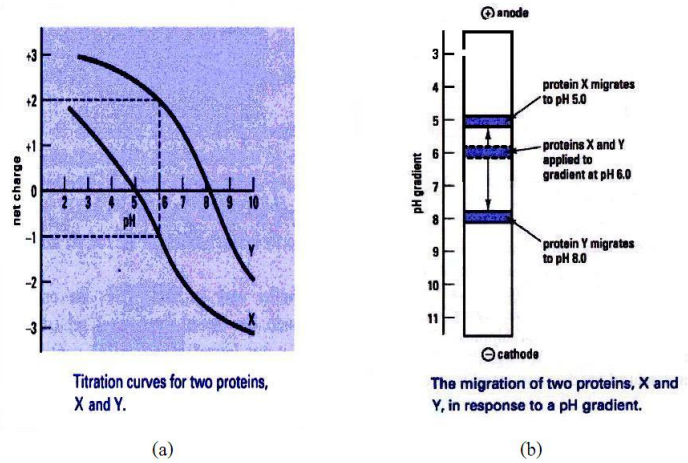
te splitsen in zijn individuele componenten. De twee voornaamste technieken die hiervoor gebruikt worden zijn **polyacrylamide gelelectroforese** en **chromatografie**.
Vooraleer we hier dieper op ingaan volgt er eerst een korte uitleg over *trypsine*, een enzyme dat in beide technieken gebruikt wordt.

Trypsine

Het enzyme trypsine heeft de eigenschap om proteïnen of peptiden op een zeer specifieke manier in stukjes te 'knippen'. Het splitst alleen peptidebindingen waarvan de koolstofgroep afkomstig is van de aminozuren lysine (K) of arginine (R). Een proteïne wordt hierdoor in peptiden geknipt. Een peptide bekomen door deze trypsinebehandeling zal dus als laatste aminozuur ofwel lysine ofwel arginine bevatten.
De eigenlijke taak van trypsine in dit proces is het katalyseren van de *hydrolyse* van deze specifieke peptidebindingen. Bij een hydrolyse reageert een chemische verbinding (hier zulk een specifieke peptidebinding) met een watermolecule H_2O waardoor de verbinding in twee gebroken wordt. Eerder zagen we dat bij de vorming van een peptidebinding een watermolecule vrijgegeven wordt. Deze hydrolyse kan aanzien worden als de inverse van de peptidebinding; de nieuwe peptide-uiteinden krijgen hun oorspronkelijke moleculen terug en zijn identiek aan de *terminals* van de oorspronkelijke proteïne.

Gelelectroforese

Polyacrylamide gelelectroforese, verder gewoon gelelectroforese genoemd, is een scheidingstechniek gebaseerd op de beweging van **geladen** moleculen in een **elektrisch veld** doorheen een gel. Na het aanmaken van de gel wordt het proteïnemengsel hierin ingespoten. Aan de kanten van deze gel worden respectievelijk een kathode (aan de kant waar het mengsel ingespoten wordt) en een anode aangelegd, oftewel een negatieve en een positieve pool. In figuur 1.9 zien we dat de gel bovenaan kanteelvormen heeft. Dit is een gevolg van de manier waarop deze gel aangemaakt wordt. We zullen niet verder ingaan op hoe deze gel gevormd wordt, maar een belangrijk voordeel van de kanteelvormen is dat, indien gewenst, verschillende proteïnestalen gelijktijdig in de gel ingebracht kunnen worden (in ieder kanteelvormpje één).
Er bestaan verschillende manieren om gelelectroforese toe te passen. Vooreerst is er de scheiding van het mengsel in één enkele richting, de zogenaamde 1-DE of één-dimensionale gelelectroforese. Hierbij wordt er rekening gehouden met een bepaalde eigenschap (die afhangt van de gebruikte soort 1-DE), op basis waarvan de scheiding in het proteïnemengsel gebeurt. Er wordt in de huidige proteïne onderzoeken echter gebruik gemaakt van 2-DE of twee-dimensionale gelelectroforese, een



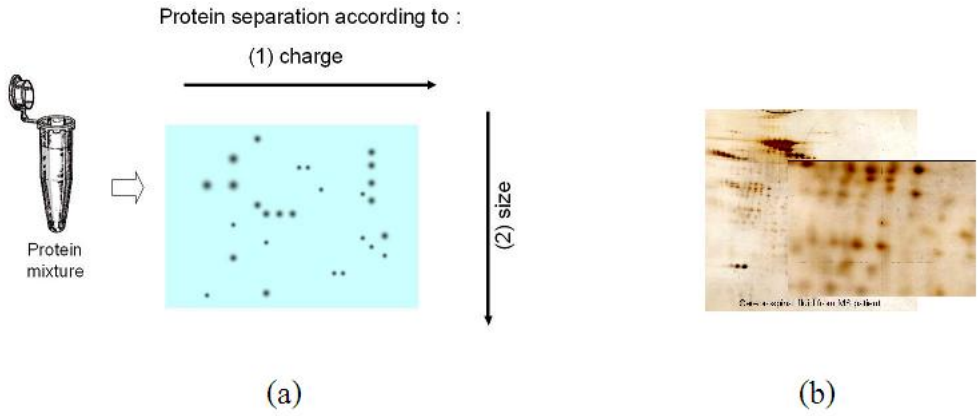
Figuur 1.10: Voorbeeld van Isoelectric Focusing [9]

veel krachtigere scheidingstechniek die het mengsel scheidt op basis van twee verschillende eigenschappen. Deze 2-DE is eigenlijk niet meer dan de combinatie van twee verschillende 1-DE technieken. Daarom zullen deze twee 1-DE technieken eerst apart besproken worden, waarna we ze samenvoegen tot 2-DE. Er bestaan nog andere 1-DE scheidingstechnieken, maar vermits 2-DE de meest gebruikte techniek is zullen we enkel die twee 1-DE technieken behandelen die deel uit maken van 2-DE.

- **IEF : 1-DE op basis van lading**
  IEF staat voor "Isoelectric Focusing". Om deze scheidingstechniek te begrijpen is de kennis van enkele begrippen nodig :
  - pH gradiënt of zuurtegraad : pH staat voor "potenz hydrogen", waarbij "potenz" staat voor sterkte of concentratie, en "hydrogen" voor het waterstofatoom H^+. De pH van een molecule is dus niets meer dan de meting van het aantal waterstofionen. Een pH gradiënt kan variëren van 0 tot 14. Een pH van 7 is neutraal, een pH kleiner dan 7 wordt als 'zuur' beschouwd, een pH groter dan 7 wordt als 'basisch' beschouwd.
  - pI = iso-elektrisch punt : iedere molecule heeft een iso-elektrisch punt. Dit is de pH-waarde waarbij de netto elektrische lading van de molecule gelijk is aan nul. Met netto elektrische lading wordt niet bedoeld dat er geen ladingen aanwezig zijn op de molecule, maar wel dat er net evenveel positieve als negatieve ladingen aanwezig zijn waardoor de netto lading nul wordt.

  IEF werkt met behulp van een IPG (Immobilized pH Gradiënt) gelstrip. De pH in zulk een gelstrip varieert lineair[9], bijvoorbeeld binnen een pH-bereik van 3 tot 10. Het proteïnemengsel wordt eerst aangebracht in striphouders. Daarbovenop worden de gelstrips geplaatst. Na een rehydratatieperiode van twaalf uur, waarbij de proteïnen in de gelstrip dringen, wordt een spanningsveld aangelegd. Aan de kant van de lage pH-waarden wordt een anode (positieve pool) aangelegd, aan de kant van de hoge pH-waarden een kathode (negatieve pool). Hierdoor gaan de componenten van dit mengsel (dus de proteïnen waaruit dit mengsel bestaat) bewegen doorheen de gelstrip. Op het moment dat een proteïne zijn iso-elektrisch punt bereikt heeft, stopt het met bewegen doorheen de gelstrip. Het proteïne heeft dan immers een netto lading gelijk aan nul, en wordt door geen van beide polen meer aangetrokken. Figuur 1.10 (b) geeft dit weer. De grafiek in figuur 1.10 (a) geeft weer hoe de netto lading van twee proteïnen X en Y verandert in functie van de pH-waarde.

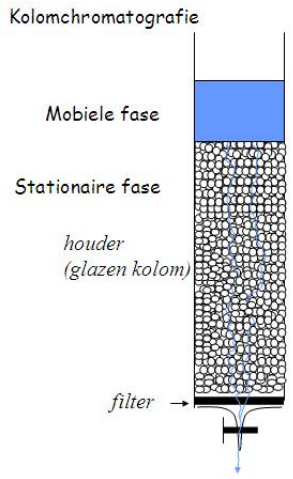
[9] Tegenwoordig wordt er ook soms gebruik gemaakt van niet-lineaire gelstrips. Hierin nemen de gradiënten waarvoor er veel moleculen verwacht worden een grotere oppervlakte in. Daardoor kunnen de moleculen beter onderscheiden worden.



Figuur 1.11: Schematische voorstelling van 2-dimensionale gel electroforese [10] (a) en een voorbeeld van een 2-DE op ruggenmergvocht van een multiple sclerose patiënt [8] (b)

- **SDS-PAGE : 1-DE op basis van massa**
  SDS-PAGE staat voor *sodium dodecyl sulfate-polyacrylamide gelelectrophoresis*. De term "polyacrylamide gelelectroforese" is ons reeds bekend, en wil niet meer zeggen dan dat er gebruik gemaakt wordt van een polyacrylamide gel om de electroforese op uit te voeren. De nieuwe term SDS is een detergent. In hun oorspronkelijke vorm hebben proteïnen meestal een 'opgerolde' structuur. Daarom worden ze eerst met SDS vermengd. Het detergent zorgt ervoor dat de proteïnen zich 'ontrollen' tot een lineaire structuur, waardoor ze zich erna makkelijker doorheen de gel gaan bewegen. Een bijkomend voordeel is dat de ontrolde proteïnen nu gemakkelijk van elkaar onderscheiden kunnen worden op basis van hun lengte (aantal aminozuren). Het belangrijkste is echter dat SDS ook effectief een binding aangaat met het proteïne, in een welbepaalde verhouding : 1.4g SDS bindt met 1.0g proteïne. Eén SDS molecule staat voor één negatieve lading. Hierdoor ontstaat er een uniforme massa/lading verhouding voor alle proteïnen die zich in het mengsel bevinden. Wordt nu het met SDS behandelde proteïnemengsel in een gel aangebracht waarover een spanning wordt aangelegd, dan zullen de proteïnen doorheen de gel gaan bewegen (dankzij hun negatieve lading, verkregen door SDS). Afhankelijk van hun grootte zal iedere proteïne anders doorheen de gel bewegen. Kleine proteïnen zullen immers makkelijk doorheen de gelporiën bewegen, de grote zullen het hier veel moeilijker mee hebben. Na een bepaalde tijdsspanne, typisch enkele uren, zal zichtbaar zijn dat de verschillende proteïnen verschillende afstanden hebben afgelegd doorheen de gel. De kleinste proteïnen zullen de grootste afstand hebben afgelegd, de grootste proteïnen zullen zich het dichtst bij de oorsprong bevinden. De proteïnen uit het complexe proteïnemengsel zijn nu dus van elkaar gescheiden op basis van hun grootte (en dus hun moleculaire massa M_r).

- **2-DE : Twee-dimensionale gelelectroforese**
  Zoals gezegd is 2-DE een combinatie van bovenstaande één-dimensionale scheidingsmethoden. Figuur 1.11 (a) geeft hiervan een schematische voorstelling weer. 2-DE verloopt effectief in twee aparte fasen, en dus niet in twee dimensies tegelijkertijd. In de eerste dimensie wordt IEF toegepast. De resulterende gelstrip van IEF wordt vervolgens met SDS behandeld en aangebracht op een nieuwe gel, net zoals dit zou gebeuren met een onbewerkt proteïnemengsel. Hierop wordt nu de electroforese toegepast.
  Een voorbeeld van een twee-dimensionale gelelectroforese, toegepast op een proteïnestaal (ruggenmergvocht) van een multiple sclerose patiënt is te zien in figuur 1.11 (b). Hierin zien we duidelijk dat er zich zogenaamde **gelspots** gevormd hebben. In het beste geval bevat zulk een gelspot proteïnen van slechts één enkele soort, maar hierover bestaat nooit zekerheid.



Figuur 1.12: Schematische voorstelling van kolomchromatografie [9]

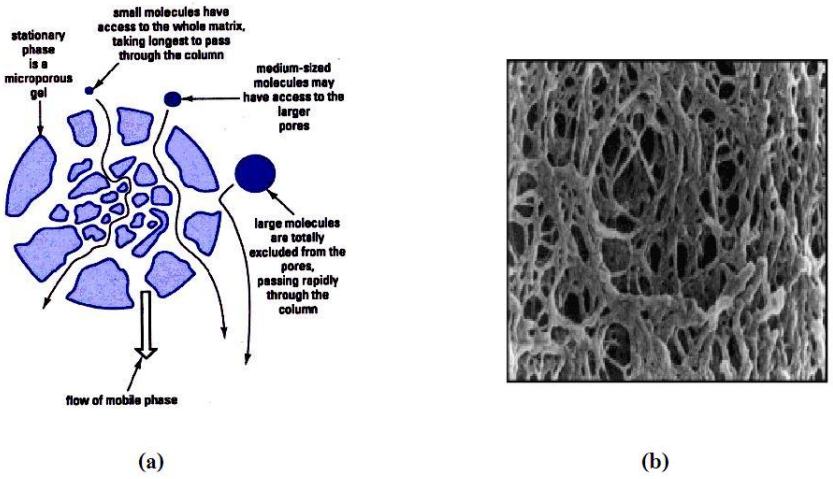
Het grootste voordeel van 2-DE (t.o.v. 1-DE) is de mogelijkheid om proteïnen te kunnen onderzoeken die één of andere vorm van posttranslationele wijziging hebben ondergaan. Dit is mogelijk omdat veel soorten proteïnewijzigingen een verandering teweeg brengen in zowel de lading als in de massa van de proteïne, die beiden door 2-DE worden weerspiegeld.
Na de gelelectroforese is het complexe proteïnestaal geëvolueerd naar een gel met gelspots, die telkens proteïnen van één soort[10] bevatten. In de volgende stap worden deze gelspots 'uitgeknipt', zodat deze allemaal afzonderlijk verder te behandelen zijn. Elke gelspot krijgt vervolgens een trypsinebehandeling en wordt een mengsel van peptiden.

Ondanks het feit dat 2-DE de vooraanstaande technologie is voor het scheiden en isoleren van proteïnen, blijven er toch nog een aantal problemen [6]. 2-DE is een werkintensief en tijdrovend proces dat niet gemakkelijk geautomatiseerd kan worden. Daarbij komt nog dat 2-DE beperkt is door het aantal en het soort van proteïne dat onderzocht kan worden. Vanzelfsprekend is er ook gezocht naar verbeteringen voor 2-DE, met betrekking tot pH gradiënten, minigel, het gebruik van fluorescerende kleurstoffen, enz. Hiernaar wordt gerefereerd in [6], maar kennis van deze technieken is niet noodzakelijk voor het verdere verloop van deze thesistekst.

Chromatografie

Een andere belangrijke scheidingstechniek is zogenaamde chromatografie. Bij chromatografie krijgt het staal vooraf een trypsinebehandeling. Zo kan men vertrekken van een complex peptidemengsel i.p.v. een complex proteïnemengsel. De peptiden in dit mengsel worden door de chromatografie gescheiden en 'gesorteerd'.
Er bestaat zowel **vloeibare** als **gasfase** chromatografie. In deze uiteenzetting zullen we ons beperken tot de beschrijving van de vloeibare chromatografie, maar de gasfase verloopt volledig analoog. Chromatografie is een tweefasen-systeem, bestaande uit een **stationaire** en een **mobiele** fase. De mobiele fase (een vloeistof bij vloeibare chromatografie, een gas bij gasfase chromatografie) bevat de moleculen, ook wel analieten genoemd, van het te scheiden staal. De stationaire fase zorgt voor de scheiding van de mobiele fase. Beide fasen hebben bepaalde eigenschappen op basis waarvan deze scheiding mogelijk is. Door deze scheiding wordt het inkomende peptidemengsel als het ware gesorteerd. Chromatografie kan opgedeeld worden in twee soorten, **gelfiltratie** chromatografie en **adsorptie** chromatografie.
Een veel gebruikte techniek om chromatografie toe te passen is kolomchromatografie. Figuur 1.12 geeft een schematische voorstelling weer van kolomchromatografie. Hier zien we ook duidelijk wat

[10] Of in minder gunstige gevallen van meerdere soorten.



Figuur 1.13: Gelfiltratie chromatografie (a) en een hiervoor gebruikte partikel (b) [9]

er bedoeld wordt met de mobiele fase en de stationaire fase. Onderaan is een 'kraantje' te zien. Hierdoor kunnen de uitkomende peptiden in groepjes opgevangen worden.

- **Gelfiltratiechromatografie**
  Bij gelfiltratiechromatografie worden de moleculen in de mobiele fase gescheiden op basis van hun moleculair gewicht. De (poreuze) partikels van de stationaire fase gedragen zich als een filter voor de mobiele fase; grote moleculen kunnen slechts op enkele plaatsen door of tussen de partikels. De kleine moleculen daarentegen kunnen zich een weg banen doorheen de kleinste 'gangen' van deze partikels, zoals wordt weergegeven in figuur 1.13 (a), waardoor ze sterk vertraagd worden. Figuur 1.13 (b) geeft een foto weer van zulk een partikel. Hier wordt het duidelijk wat er met de zogenaamde gangen bedoeld wordt. De grote moleculen zullen dus eerst doorheen de stationaire fase geraken, de allerkleinste moleculen als laatste.

- **Adsorptiechromatografie**
  Adsorptiechromatografie scheidt de analieten in de mobiele fase op basis van de moleculaire eigenschappen van zowel deze mobiele fase, de stationaire fase als van de analieten zelf. Deze factoren bepalen de zogenaamde verdelingscoëfficiënt K voor alle analieten:

$$K = \frac{[analiet\ gebonden\ aan\ stationaire\ fase]}{[analiet\ in\ mobiele\ fase]} \tag{1.1}$$

  Hoe groter K, hoe groter de affiniteit is van een analiet voor de stationaire fase, en dus hoe groter de vertraging van deze analiet zal zijn door de stationaire fase. Uiteraard kunnen alleen analieten met een verschillende verdelingscoëfficiënt van elkaar gescheiden worden. De eigenschappen van de analieten bepalen de aard van de stationaire fase en de samenstelling van de mobiele fase. Hierdoor zullen er interacties plaatsvinden tussen de analieten en de stationaire fase. Er zijn vier combinaties van eigenschappen en interacties die gebruikt worden om de analieten te scheiden. Tabel 1.1 geeft hiervan een overzicht. Deze vier soorten adsorptiechromatografie worden respectievelijk **polaire chromatografie** (ook wel normale fase genoemd), **apolaire chromatografie** (of omgekeerde fase), **ionenuitwisselingschromatografie** en **affiniteitschromatografie** genoemd. Het principe blijft hetzelfde bij deze verschillende methoden; de analieten worden van elkaar gescheiden op basis van een bepaalde eigenschap. Het scheiden zelf kan plaatsvinden dankzij de juiste keuze van stationaire en mobiele fase, waardoor er een bepaalde interactie zal optreden tussen de analieten en de stationaire fase (zie ook tabel 1.1).

chromatografie	analiet	stationaire fase	interactie	mobiele fase
Polaire	polair	polair	H-brug, dipool-dipool	organische solventen
Apolaire	hydrofoob	hydrofoob	Van der Waals	waterige buffer
Ionenuitwisselings-	geladen	geladen	electrostatisch	zoutarme buffer
Affiniteits-	vorm	biospecifiek	sleutel-slot	waterige buffer

Tabel 1.1: Adsorptie chromatografie : de 4 soorten [9]

Opmerking

Een belangrijk verschil tussen gelelectroforese en chromatografie is dat de peptiden bekomen door electroforese afkomstig zijn van één of hoogstens enkele soorten proteïnen. Bij chromatografie echter krijgen we een peptidemengsel afkomstig van alle proteïnen in het oorspronkelijke staal. Vanzelfsprekend zal de identificatie van de proteïnen moeilijker verlopen bij deze laatste methode, maar hieraan zullen we verder geen aandacht besteden.

1.2.3 Waarnemingen met de massaspectrometer

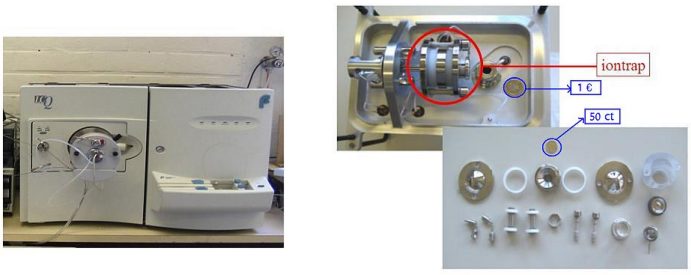
Met massaspectrometrie of kortweg MS kan men structurele informatie, zoals de massa of de aminozurensequentie van een peptide bekomen. Dit gebeurt met behulp van een massaspectrometer. Er bestaan veel verschillende types massaspectrometers, maar ze hebben allen een gelijkaardige werking, nl. het meten van de massa/lading (m/z) verhoudingen van ionen (geïoniseerde peptiden) en de intensiteiten waarmee deze verhoudingen voorkomen. Ook kan er een interval ingesteld worden (bv. intensiteit, m/z). Het doel hiervan komt aan bod in de sectie over het opbreken van parent ionen. De werking van een massaspectrometer is als volgt :

- ionisatie van de peptiden,
- m/z detectie van de ionen,
- MS spectrum,
- opbreken van parent ionen,
- m/z detectie van de nieuwe, kleinere ionen,
- MS/MS spectrum als output.

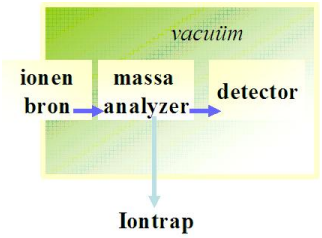
Aangezien de verscheidenheid in toestellen te groot is om ze allen tot in detail te bespreken, zullen we ons in deze tekst beperken tot de beschrijving van de massaspectrometer van Biomed (figuur 1.14 (a)), een LCQ ThermoFinnigan Classic [11]. LC staat voor *vloeibare chromatografie kolom - hydrofobe fase*, wat verderop duidelijk zal worden. Het interval dat voor dit toestel ingesteld kan worden is een m/z waarde interval. De algemene werking van de massaspectrometer zal aan de hand van dit toestel verduidelijkt worden.
De output van een scheidingsmethode (dus een peptidemengsel) wordt de input voor deze massaspectrometer. De informatie die de massaspectrometer verschaft over het peptidemengsel wordt dan in de volgende stap gebruikt om deze peptiden (*in silico*) te identificeren. Figuur 1.15 geeft een schematische voorstelling weer van de massaspectrometer, en laat de stappen zien die binnenin gebeuren. Deze stappen bepalen hoe de input van de massaspectrometer omgevormd wordt in zijn output. Wat deze output precies is, zal gaandeweg duidelijk worden.

Ionisatie van de peptiden

Om de peptiden in de massaspectrometer in te brengen wordt er gebruik gemaakt van vloeibare kolomchromatografie (LC). Op basis van zijn hydrofobe eigenschappen wordt het peptidemengsel nog eens extra gescheiden om op een 'geordende' manier de massaspectrometer binnen te gaan.



Figuur 1.14: Massaspectrometer "LCQ ThermoFinnigan Classic" van Biomed (a) en de iontrap ervan (b) [10]

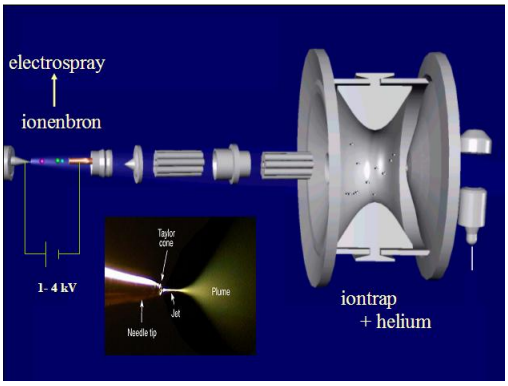


Figuur 1.15: Schematische voorstelling van de werking van een massaspectrometer [10]

Deze extra kennis over de hydrofobe eigenschappen van de peptiden kan nuttig zijn indien men later twijfelt tussen verschillende resultaten van de identificatie (*in silico*), of indien men twijfelt aan de correctheid hiervan.
Het inkomende peptidemengsel wordt nu met behulp van een **electrospray** in de massaspectrometer 'geneveld'. Op figuur 1.16 zien we dat er zich in de electrospray een ionenbron bevindt. Deze ioniseert de peptiden, m.a.w., de peptiden krijgen een lading. Onderaan op de figuur zien we een foto van het nevelen van de ionen. De reden voor deze ionisatie is dat de massaspectrometer in de volgende stappen enkel kan werken met geladen moleculen. Vanaf nu zullen we daarom spreken over (peptide-) ionen i.p.v. over peptiden. Wat er werkelijk gebeurt, is het toevoegen van protonen bij de inkomende peptiden waardoor er een positieve lading ontstaat op elke peptide. Afhankelijk van de gebruikte massaspectrometer ontstaat er een éénwaardige of een meerwaardige (typisch twee- of driewaardige) lading op de peptiden die zo ionen worden. Bij de bespreking van de algoritmen zullen we zien dat hiermee rekening gehouden moet worden. Merk op dat het toevoegen van één of meerdere protonen aan een peptide niet enkel de lading maar ook de massa ervan verandert. De massa van een proton is gelijk aan 1 *Da*.
We zien in de hoofdfiguur van figuur 1.16 ook de **iontrap** die in het volgende stadium zal zorgen voor de detectie van de m/z van de ionen.

m/z detectie van de ionen

De detectie van m/z waarden gebeurt in de beschreven massaspectrometer met een iontrap. In de iontrap is een 3D-elektrisch veld aangelegd waar de ionen als het ware in gevangen worden. De kokervormige onderdelen tussen de ionenbron en de iontrap op figuur 1.16 (niet benoemd op de figuur) geleiden de ionen tot aan de iontrap. Bij het voorkomen van een m/z lading wordt zijn intensiteit verhoogd. Deze gegevens worden opgeslagen in de massaspectrometer. Een groot voordeel van de iontrap t.o.v. andere detectie methoden, is dat de ionen er in 'gevangen' kunnen worden. Vervolgens kan er uit alle ionen gekozen worden welke ionen geëjecteerd zullen worden en welke niet, aan de hand van het ingestelde interval. De geselecteerde ionen zijn de zogenaamde **parent ionen**.



Figuur 1.16: Onderdelen van een massaspectrometer: de electrospray met ionenbron en de iontrap [10]

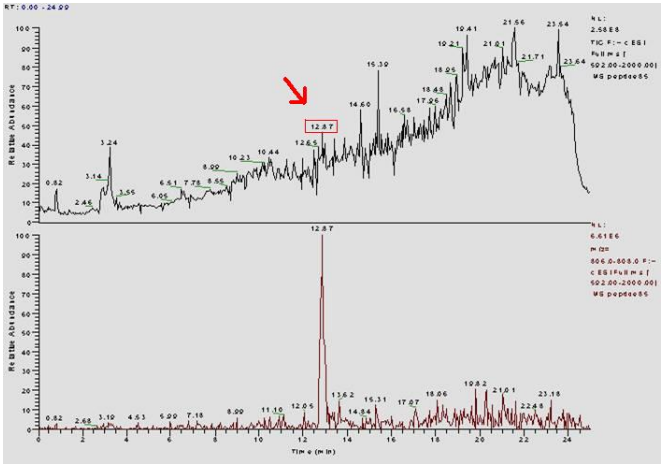
Een interessant detail is de manier waarop de m/z gemeten wordt. De massaspectrometer meet immers een m/z waarde en niet de massa en de lading apart om daaruit de m/z waarde te berekenen. Het is niet nodig om in te gaan op de technische kant van deze meting, maar dit impliceert wel dat de lading van een ion eigenlijk niet gekend is. Enkel indien de ionisatiemethode enkel éénwaardige ionen voortbrengt kent men ook de massa van de ionen. Bij meerwaardig geladen ionen zijn de mogelijkheden meestal beperkt tot tweewaardig of driewaardig geladen ionen, maar is niet bekend over welke lading het precies gaat. Hierdoor kan de massa van de ionen niet met zekerheid bepaald worden en moet het verdere onderzoek meerdere mogelijkheden in acht nemen. We beperken deze tekst veelal tot éénwaardig geladen ionen en zullen hier niet verder op ingaan.
Om een idee te krijgen van het uitzicht en de grootte van een iontrap is er één te zien in figuur 1.14 (b). De muntstukken laten toe de ware grootte in te schatten.

MS spectrum

Als we alle gegevens in deze fase zouden verzamelen hebben we niets meer dan een lange reeks getallenparen (massa/lading - intensiteit). Om dit visueel voor te stellen kan dit uitgetekend worden in grafiekvorm. De x-as geeft hierbij de massa/lading verhoudingen weer, de y-as de intensiteit. Figuur 1.17 (boven) geeft hiervan een voorbeeld. Zowel de ruwe data op zich als de grafiek worden het massaspectrum genoemd. In wezen stellen ze dan ook exact hetzelfde voor.
We kunnen in het voorbeeld duidelijke pieken zien. Deze weerspiegelen de veel voorkomende ionen; de ionen met een hoge intensiteit. In figuur 1.17 (onder) zien we een uitvergroting van een geselecteerde piek. Hier is het heel duidelijk zichtbaar dat de pieken geen 'zuivere' pieken zijn. De kleinere pieken rondom zijn vaak ruis, veroorzaakt door de beperkingen van de massaspectrometer. Bij de m/z meting van het parent ion kunnen we op een eenvoudige manier de oorspronkelijke massa van de te identificeren parent *peptide* berekenen. Is de lading gelijk aan 1, dan is de massa van de parent peptide gelijk aan (m-1). De éénwaardige lading heeft immers de massa van de parent peptide met 1 Da verhoogd. Is de lading gelijk aan 2, dan is m/z gemeten door de massaspectrometer gelijk aan $(m_p + 2)/2$, met m_p de massa van de parent peptide en $(m_p + 2)$ de massa van het parent ion. Dan krijgen we $m_p = 2 * (m/z) - 2$. Analoog voor hogere ladingen.

Opbreken van parent ionen

Figuur 1.17 (boven) geeft de selectie van een mogelijke parent ion weer. Iedere geselecteerde parent ion wordt vervolgens verder opgesplitst door het bloot te stellen aan elektrische energie. De gebruikte methode hiervoor is CID, Collision Induced Dissociation, een proces waarbij het ion in kwestie gefragmenteerd wordt door de interactie met een neutraal gas. Dit gebeurt doordat een deel van de translatie-energie van het ion omgezet wordt in interne energie, waardoor de fragmentatie plaatsvindt en het ion in twee *fragmenten* opbreekt. De lading van het parent ion wordt



Figuur 1.17: Voorbeeld van een massaspectrum (boven) en een vergroot beeld (onder) van het geselecteerde parent ion [10]

daarbij verdeeld over de twee fragmenten. Indien er maar een éénwaardige lading aanwezig is zal er ook maar één fragment geladen worden. Het andere fragment is dan niet geladen (neutraal fragment) waardoor het niet door de massaspectrometer gemeten kan worden. Bij meer-waardige ladingen kan de lading zowel verdeeld worden over de twee fragmenten als dat de volledige lading op slechts één fragment aanwezig is. Ook dan gaat het neutraal fragment verloren voor de verdere metingen. De geladen fragmenten worden *fragmentatie-ionen* genoemd. Het zijn deze ionen die door de massaspectrometer gemeten zullen worden in de volgende stap.
Het grote verschil met het knippen van proteïnen en peptiden is het ontbreken van één van de twee *terminals* bij de fragmentatie-ionen. In tegenstelling tot het knippen met trypsine is de fragmentatie géén *hydrolyse*. We krijgen dus telkens een fragment met enkel een *N-terminal* en/of een fragment met enkel een *C-terminal*, wat leidt tot zogenaamde N-terminale en/of C-terminale ionen.
Het 'breekpunt' van het parent ion kan op verschillende plaatsen binnen het ion voorkomen, waardoor er verschillende soorten fragmentatie-ionen ontstaan. Dit zal verderop aan bod komen in sectie 1.2.4.
Heel belangrijk om weten is dat we vertrokken zijn met een peptidemengsel waarin alle verschillende peptiden meermaals voorkomen. Als we hier spreken over een parent peptide of parent ion, dan spreken we eigenlijk over alle peptiden uit het mengsel met dezelfde massa. Alle ionen met dezelfde massa worden dus gefragmenteerd. We krijgen daarom niet twee fragmentatie-ionen, maar een heleboel fragmentatie-ionen die onderling kunnen verschillen in de manier waarop ze gefragmenteerd zijn (sectie 1.2.4).

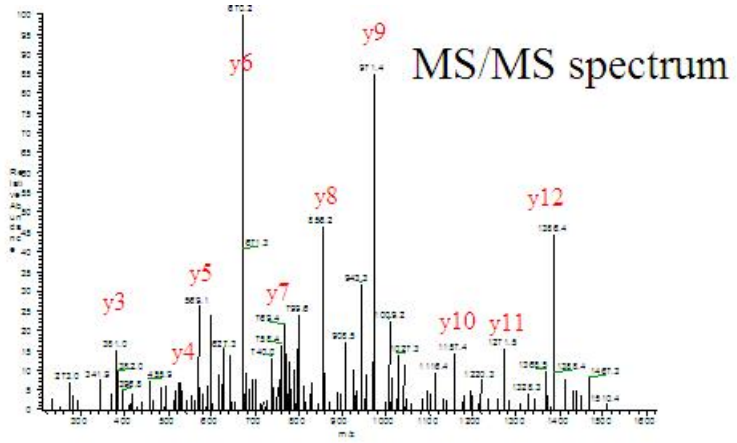
m/z detectie van de nieuwe ionen

De m/z verhoudingen van de resulterende fragmentatie-ionen worden vervolgens gemeten, net zoals bij het parent ion gebeurd is[11].

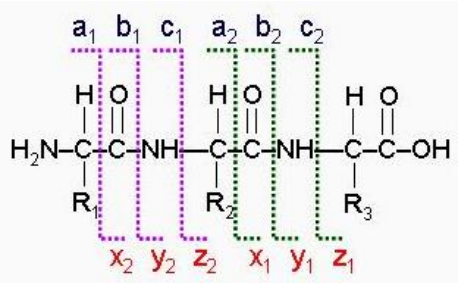
MS/MS spectrum als output

Ook de intensiteiten van de voorkomende m/z verhoudingen worden gemeten. Het resultaat van zo'n meting, van één enkele (soort) parent ion, is te zien in figuur 1.18 en wordt het **MS/MS spectrum** of ook wel **tandem massaspectrum** genoemd. Dit proces gebeurt voor alle gekozen parent ionen waardoor we een verzameling van MS/MS spectra bekomen. Een voorbeeld van deze uiteindelijke data file is te vinden in bijlage B. Bij het identificatieproces worden de verschillende

[11] Ook hier wordt er enkel een m/z waarde opgemeten, en niet de massa en de lading afzonderlijk.



Figuur 1.18: MS/MS spectrum van het parent ion uit figuur 1.17 [10]



Figuur 1.19: Fragmentatie van een peptide volgens de Biemann nomenclatuur [10]

parent ionen één voor één geïdentificeerd, door in eerste instantie de fragmentatie-ionen waaruit ze bestaan te identificeren.
Om deze fragmentatie-ionen te kunnen identificeren zullen we zien dat de algoritmen proberen te achterhalen wat de massa is van deze fragmenten indien ze niet geïoniseerd zouden zijn. In tegenstelling tot de massaberekening van de parent peptide is de massaberekening van de niet geïoniseerde fragmenten niet zo voor de hand liggend. De wijze waarop de fragmentatie gebeurt beïnvloedt de massa van deze fragmentatie-ionen, waardoor er geen eenvoudige formule bestaat die tot de juiste oplossing leidt. Het verschil in massa tussen het fragmentatie-ion en de som van de aminozuren waaruit het bestaat wordt benoemd met het symbool δ. Deze δ is verbonden met het type fragmentatie-ion. De volgende sectie belicht deze verschillende types fragmentatie-ionen. De δ-waarden zullen in het volgende hoofdstuk aan bod komen.
Van het parent ion weten we welke lading het bevat vermits deze door de massaspectrometer meegegeven wordt. Bij de fragmentatie-ionen kan men enkel uitsluiten welke ladingen *zeker niet* kunnen voorkomen[12]. Daarenboven kunnen er vele verschillende fragmentatie-ionen (sectie 1.2.4) gevormd worden uit één bepaalde parent ion. Het MS/MS spectrum geeft hieromtrent echter geen informatie.

1.2.4 Fragmentatie van parent ionen

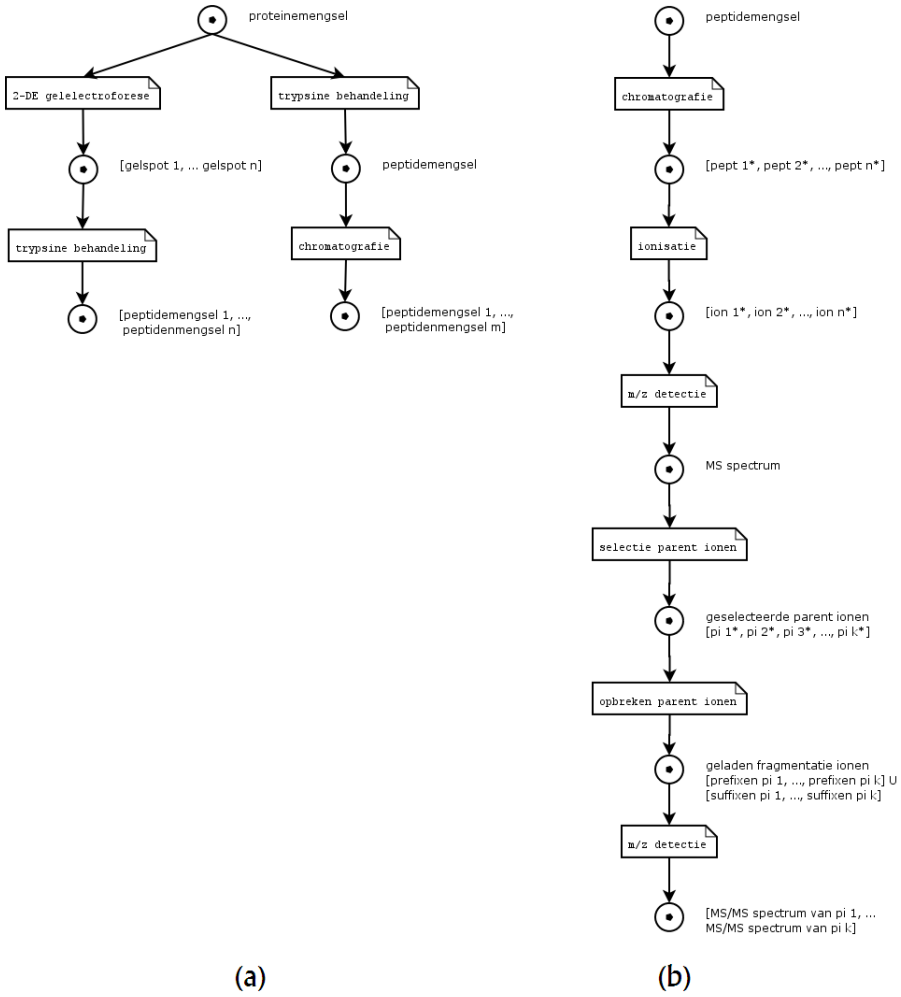
Het fragmenteren van parent ionen kan op verschillende manieren gebeuren. Deze verschillende mogelijke fragmentatie-ionen maken het identificatie proces uiteraard niet eenvoudiger. Meestal gebeurt dit echter op een manier die leidt tot N-terminale en C-terminale ionen, ook wel de *prefix* en de *suffix* genoemd. Figuur 1.19 geeft de mogelijke fragmentatie-ionen weer die hieraan voldoen.

[12] Bv.: een tweewaardige lading als het parent ion slechts éénwaardig geladen is, is uitgesloten.

Deze ionen worden volgens de 'Biemann nomenclatuur' met de letters a, b en c benoemd indien het om N-terminale ionen gaat, en met de letters x, y en z indien het om C-terminale ionen gaat. We zien duidelijk dat een a-ion complementair is met een x-ion, een b-ion met een y-ion en een c-ion met een z-ion. Zoals reeds aangehaald krijgt slechts één van de twee fragmentatie-ionen een lading indien het parent ion een éénwaardige lading heeft. Heeft het parent ion een tweewaardige lading, dan krijgen beide fragmentatie-ionen meestal elk één proton, maar het kan ook voorkomen dat slechts één fragmentatie-ion een tweewaardige lading krijgt en het andere fragment neutraal blijft. Meer dan tweewaardige ladingen komen minder voor, maar hiervoor geldt hetzelfde principe. Aangezien de fragmentaties vaak maar één enkel ion opleveren is het duidelijk dat het aantal fragmentatie-ionen van complementaire ion types niet met elkaar overeen moet komen. Bij splitsing volgens de b- en y-ionen-regel komt het bijvoorbeeld vaker voor dat er een y-ion gevormd wordt dan een b-ion.
Het kan echter ook voorkomen dat een parent ion helemaal niet gefragmenteerd wordt en dus gewoon het parent ion blijft. Vermits de massaspectrometer de massa en de lading van het parent ion reeds gemeten heeft zullen we na de tweede meting van de massa geen extra informatie krijgen. Er wordt in de identificatie algoritmen dan ook geen rekening gehouden met de mogelijke aanwezigheid van een niet-gefragmenteerde parent ion in het MS/MS spectrum.
We zien op de figuur dat de genoemde fragmentatieplaatsen tussen elke twee aminozuren kunnen voorkomen. De plaats waar de fragmentatie gebeurt, verschilt van parent ion tot parent ion. Dit is duidelijk zichtbaar op figuur 1.18 waar de aanwezige y-ionen op zijn benoemd. We zien ook dat het spectrum niet alle mogelijke y-ionen bevat. Er hebben bijvoorbeeld geen fragmentaties plaatsgevonden die geleid hebben tot de vorming van het y_2-ion.
Interne fragmentatie-ionen kunnen ontstaan indien het parent ion in drie i.p.v. twee stukken opbreekt, maar dit komt zelden voor aangezien de gebruikte low-energy CID de typische 'Biemann'-fragmentatie als gevolg heeft. De twee meest voorkomende fragmentatie-ionen zijn de b- en de y-ionen. Bij deze fragmentatie breekt het parent ion op de oorspronkelijke peptidebinding tussen twee aminozuren. Er wordt in een aantal zoekmethoden daarom enkel rekening gehouden met deze twee soorten fragmentatie-ionen. De hoofdreden hiervoor is dat op deze manier het zoekproces eenvoudiger wordt, en dus computationeel haalbaar. Toch onstaan er ook vaak andere dan b- en y-ionen en is het negeren van deze minder voorkomende gevallen uiteraard een tekortkoming. Men streeft hierbij naar een evenwicht tussen het behouden van de nodige informatie en een computationeel haalbaar algoritme.
De mogelijke fragmentatie-ionen hangen af van de gebruikte fragmentatiemethode (bij de voorgestelde massaspectrometer : CID). We zagen ook reeds dat de gebruikte ionisatiemethode bepaalt welke lading de parent ionen krijgen. Kennis van de gebruikte massaspectrometer is dus van groot belang om de meetresultaten op de juiste manier te kunnen interpreteren. Aangezien de identificatie algoritmen deze taak hebben, zijn de meeste dan ook specifiek geschreven om de data te identificeren afkomstig van een bepaald type massaspectrometer (bv. met iontrap). Enerzijds moeten deze algoritmen daarom enkel rekening houden met de mogelijke ladingen en fragmentatie-ionen. Anderzijds is het verre van optimaal dat de algoritmen 'instrument-afhankelijk' zijn, waardoor ze slechts beperkt toepasbaar zijn. Bij de beschreven algoritmen in de volgende delen van deze thesistekst zal dit euvel aan bod komen.
Tot slot kunnen er na de fragmentatie nog wijzigingen plaatsvinden in de fragmentatie-ionen. Een vaak voorkomend voorbeeld hiervan is het verlies van een watermolecule, waardoor het fragmentatie-ion een kleinere massa krijgt. Het is deze kleinere massa die wordt gemeten door de massaspectrometer. Deze wijzigingen worden aanzien als 'andere' ion types zoals bv. een $b - H_2O$-ion. Aangezien enkele van deze wijzigingen vaak voorkomen tracht men hier rekening mee te houden in de identificatie algoritmen.

1.2.5 Overzicht : van proteïnen tot MS/MS spectra

Om het overzicht te bewaren geven we een schematische voorstelling van het hele voorbereidende proces, vertrekkende van een proteïnestaal om uiteindelijk te komen tot MS/MS spectra, die gebruikt kunnen worden voor de identificatie van de peptiden. Figuur 1.20 (a) geeft het proces



Figuur 1.20: Schematische voorstelling : van proteïnemengsel tot peptidemengsels (a) en van één peptidemengsel tot MS/MS spectra (b)

weer van één poteïnestaal dat gescheiden wordt in verscheidene peptidemengsels. Figuur 1.20 (b) vertrekt van één peptidemengsel, gaande van de input van de massaspectrometer tot de output als verzameling van MS/MS spectra. De cirkels stellen data voor. De aard van deze data staat steeds rechts van het symbool. Data omsloten door rechte haken stelt een verzameling of array voor van soortgelijke data. De rechthoeken zijn de bewerkingen.

1.2.6 Identificatie van peptiden

De gegevens bekomen uit de massaspectrometer worden nu gebruikt voor de eigenlijke identificatie van de peptiden. Er bestaan twee grote groepen algoritmen om hiertoe te komen. Een eerste groep wordt *de novo* peptide sequencing genoemd en voert de identificatie uit enkel op basis van de gegevens die het massaspectrum bevat.
De andere groep algoritmen maakt gebruik van proteïnedatabases. Er bestaan wereldwijd een aantal databases die toegankelijk zijn via het internet. Deze databases bevatten structurele gegevens van gekende proteïnen, zoals de sequentie van aminozuren en de massa. Een voorbeeld van zo'n database is SwissProt [12]. Eerder werd het gebruik van trypsine toegelicht. Dankzij de specifieke knip-eigenschap hiervan kan er op eenvoudige wijze bepaald worden welke peptiden afkomstig kunnen zijn van een bepaalde proteïne uit de database. Een gedetailleerde uitleg volgt in hoofdstuk 2.

De werking van een aantal *de novo* algoritmen vormt het eigenlijke onderwerp van deze thesis (deel II). In hoofdstuk 6 wordt een algoritme besproken dat de *de novo* werkwijze combineert met een database search.
Een factor die een rol speelt in het bemoeilijken van de identificatie zijn de beperkingen van de massaspectrometer. Ook het meest geavanceerde toestel is niet perfect, en er zal altijd een hoeveelheid ruis in de resultaten aanwezig zijn. Belangrijk is dus om deze ruis van de eigenlijke data te onderscheiden. Meestal wordt hiervoor een bepaalde grens ("threshold") bepaald zodat alle bovenliggende waarden als data worden beschouwd, en de onderliggende waarden als ruis. Vanzelfsprekend is dit geen ideale methode en zal er zowel data verloren gaan als dat er ruis in de als data geklasseerde gegevens overblijft. Hoe men deze en nog meer moeilijkheden tracht op te lossen komt uitgebreid aan bod bij de bespreking van verschillende identificatie algoritmen. Vooreerst volgt er een inleidend hoofdstuk.

Hoofdstuk 2

Inleiding tot de algoritmen voor de identificatie van peptiden

De algoritmen die gebruikt worden voor de identificatie van peptiden kunnen opgedeeld worden in twee grote groepen : *de novo* algoritmen (ook *de novo* peptide sequencing genoemd) en algoritmen die voor de identificatie gebruik maken van een proteïne database. Dit hoofdstuk geeft een korte inleiding op deze twee groepen algoritmen. Deze thesis legt vooral de nadruk op *de novo* algoritmen. Hierbij komen technieken uit de theoretische informatica aan te pas. Aangezien er talloze algoritmen bestaan hebben we de keuze gemaakt om twee *de novo* algoritmen te bespreken die een totaal verschillende aanpak hebben (hoofdstukken 3 en 5). Ook een combinatie algoritme dat zowel een *de novo* gedeelte bevat als gebruik maakt van een database komt aan bod. We zullen zien dat het *de novo* gedeelte de basisstructuur van ons eerste algoritme anders aanwendt. De output van dit *de novo* gedeelte wordt vervolgens gebruikt als *tag* om een database te doorzoeken (hoofdstuk 6).
Alle algoritmen hebben als *input* een experimenteel bekomen tandem massaspectrum, het resultaat van de verwerking van een proteïnestaal zoals beschreven in het vorige hoofdstuk. Het 'inputspectrum' wordt daarom ook wel het *experimenteel spectrum* genoemd, dit om een onderscheid te maken met het *theoretisch spectrum*, een theoretische berekening van het spectrum van gekende peptiden (sectie 2.4). Indien er verwarring kan ontstaan, zullen we een duidelijk onderscheid maken tussen experimentele spectra en theoretische spectra. We benadrukken nog even dat we met een spectrum S één spectrum van de verzameling MS/MS spectra bedoelen dat overeenkomt met één parent peptide. Concreet bevat spectrum S dus de m/z en de lading van deze parent peptide en de m/z en intensiteit van alle opgemeten fragmentatie-ionen van deze parent peptide. De algoritmen voor de identificatie kunnen op alle parent ionen uit de verzameling MS/MS spectra toegepast worden zodat men uiteindelijk al deze geïoniseerde peptiden kan identificeren. Zoals in het vorige hoofdstuk werd aangegeven tracht men met deze geïdentificeerde peptiden de oorspronkelijke proteïnen te identificeren, maar dit vormt geen onderdeel van dit thesisonderwerp.
De *output* van een algoritme is de identificatie van de input, m.a.w., de identificatie van de ongekende parent peptide.

2.1 Probleemstelling

Tandem massaspectrometrie "peptidesequentieprobleem" :

- Gegeven :
  - spectrum S (MS/MS) van één te identificeren parent peptide[1];
  - massa m en lading z van de parent ion, gemeten door de massaspectrometer.

[1] Onderdeel van de output van de massaspectrometer, zie ook Bijlage B.

- Gevraagd : vind een peptide met massa m, met een maximale match voor spectrum S, d.w.z., verklaar het spectrum S door de peptide te identificeren die dit spectrum gevormd heeft.

2.2 *De novo* peptide sequencing

De novo peptide sequencing bestaat in het algemeen uit twee grote stappen. Ten eerste wordt er, gebruik makend van het gegeven spectrum S, gezocht naar aminozurensequenties die overeen kunnen komen met dit spectrum. Er wordt dus gezocht naar peptiden waarvoor het spectrum S opgemeten zou kunnen worden door een massaspectrometer. Op deze manier wordt er getracht voor het parent ion alle mogelijke peptiden te reconstrueren. In de tweede stap worden deze peptiden heuristisch gerangschikt. De peptide met de hoogste score wordt meestal aanzien als de mogelijke identificatie voor het spectrum S.
We kunnen hier reeds opmerken dat we spreken over *alle mogelijke peptiden*. Dit duidt erop dat een spectrum door meerdere peptiden verklaard kan worden. Uiteraard zijn we enkel op zoek naar die peptide die het spectrum gegenereerd heeft, waardoor de tweede stap noodzakelijk is. We zullen zien dat het verre van eenvoudig is om de juiste oplossing te vinden.
Hoewel het gebruik van databases een veel gebruikte methode is om peptiden te identificeren, is *de novo* vaak van nut voor specifieke situaties [13]. Indien de peptide die men wil identificeren nog niet in de bestaande databases aanwezig is, zal het uiteraard ook niet geïdentificeerd kunnen worden door er in deze databases naar te zoeken. Dit kan zich voordoen wanneer de database onvolledig is, wat vaak het geval is: van nog maar enkele organismen zijn de meeste proteïnen, en dus ook peptiden, gekend. Een andere oorzaak van deze onvolledigheid is dat genen *alternatieve splicing* kunnen ondergaan bij de vorming van *mRNA*, waardoor er andere proteïnen ontstaan. Uit de inleiding in hoofdstuk 1, sectie 1.1.2, kan men afleiden dat een wijziging in genen een wijziging in codons betekent, wat kan leiden tot de vorming van andere aminozuren en dus andere, nieuwe proteïnen die mogelijk nog niet in de databases aanwezig zijn. Een identificatie methode die gebruik maakt van een database zal in dat geval geen of een foutieve identificatie voortbrengen.

2.3 Proteïne databases

In vele gevallen is een database zoekmethode zeer efficiënt, ondermeer omdat deze methode vaak sterk geautomatiseerd kan worden [14]. De algoritmen trachten de ongekende peptiden te identificeren door in een proteïne database op zoek te gaan naar mogelijke antwoorden, wat neerkomt op het principe van "looking up the answer in the back of the book" [15].
We spreken al geruime tijd over proteïne databases, hoewel we geen proteïnen maar wel peptiden moeten identificeren. Dankzij het gebruik van trypsine (sectie 1.2.2) bij het voorbereidende proces kan men precies bepalen welke peptiden uit een (gekende) proteïne kunnen bekomen worden. De database bevat immers de aminozurensequenties van de proteïnen die we nu 'virtueel' kunnen knippen met de knipeigenschap van trypsine[2]. Aangezien een peptide niets anders is dan een aminozurensequentie (en deze massa's gekend zijn (Bijlage C)) kunnen ook de massa's van deze peptiden op een eenvoudige manier berekend worden. In een te identificeren spectrum komen echter niet de massa's van peptiden, maar wel de massa's van de fragmentatie-ionen van deze peptiden voor. Meer details hierover volgen in de volgende sectie waar het *theoretisch spectrum* aan bod komt. Het theoretisch spectrum wordt opgesteld om de massa's van de fragmentatie-ionen, afkomstig van de 'virtueel' bekomen peptiden, te berekenen. Eens het theoretisch spectrum van een peptide gekend is, kan dit vergeleken worden met het spectrum S om zo te bepalen in hoeverre er een match bestaat tussen deze twee spectra, en dus of het theoretische spectrum een mogelijke identificatie is voor S.
De meeste proteïne databases bevatten voor iedere aanwezige proteïne twee soorten data; de kerninformatie en de aantekeningen. De kerninformatie bevat o.a. de sequentie zelf, maar ook gegevens zoals de biologische oorsprong van de proteïne, bv. afkomstig van de mens of van een bacterie. De aantekeningen bevatten een heleboel andere informatie zoals de functie(s) van de proteïne,

[2] Voor iedere proteïne in de database kunnen we dus bepalen welke peptiden hiervan afkomstig kunnen zijn.

posttranslationele wijzigingen en de interne structuren. Uiteraard is de proteïnesequentie het belangrijkste voor het identificatieproces. De aanwezigheid van de overige informatie verschilt van database tot database.

Ook het formaat waarin de gegevens in een database zijn opgeslagen verschilt van database tot database. Dit kan van belang zijn met betrekking tot het zoekalgoritme dat men wenst te gebruiken. Sommige algoritmen vereisen een bepaald formaat en kunnen enkel op die databases toegepast worden die dit formaat hanteren [16].
Een veel voorkomend en zeer eenvoudig formaat is het FASTA-formaat [17]. Een sequentie in FASTA-formaat bestaat uit een beschrijving van één lijn, gevolgd door een aantal lijnen sequentiedata. De beschrijving begint steeds met het symbool '>', en geen enkele lijn mag langer zijn dan 80 karakters. Spaties worden genegeerd (en wordt dus enkel gebruikt voor de leesbaarheid).
Een voorbeeld van een database entry in FASTA-formaat :

Listing 2.1: Voorbeeld van een database entry in FASTA-formaat

```
>LCA_HUMAN
      MRFFVPLFLV GILFPAILAK QFTKCELSQL LKDIDGYGGI ALPELICTMF HTSGYDTQAI
      VENNESTEYG LFQISNKLWC KSSQVPQSRN ICDISCDKFL DDDITDDIMC AKKILDIKGI
      DYWLAHKALC TEKLEQWLCE KL
```

2.4 Het theoretisch spectrum

Het theoretisch spectrum van een peptide wordt vooral gebruikt in de database zoekmethoden. Er komen bij de berekening ervan echter aspecten aan bod die *in het algemeen* deel uit maken van het peptide identificatieproces. Kennis van theoretische spectra draagt bovendien bij tot het inzicht in experimentele spectra. We laten daarom het theoretisch spectrum uitgebreid aan bod komen.

2.4.1 Enkele notaties

In deze sectie geven we reeds een aantal definities en notaties die nodig zijn voor het berekenen van een theoretisch spectrum. Sectie 2.5 geeft een volledig overzicht van alle definities en notaties die gelden voor het hele verdere verloop van de tekst.

- Zij P een peptide,
- een partiële peptide P' van P is een substring $p_i...p_j$ van P met massa $m(P') = \sum_{i \leq t \leq j} m(p_t)$, de som van de massa's van de aminozurenresidu's,
- een δ-ion van een partiële peptide P' is een wijziging van P', zó dat het ion een massa heeft die gelijk is aan $m(P') - \delta$, bv. $\delta = -1$ voor een b-ion (meer details omtrent δ-waarden volgt in de volgende secties).

2.4.2 Berekening van een theoretisch spectrum

In sectie 1.2.4 hebben we kennis gemaakt met de meest voorkomende types fragmentatie-ionen in tandem massaspectrometrie, namelijk de b- en de y-ionen. Om een theoretisch spectrum te berekenen wordt er meestal enkel met deze twee types fragmentatie-ionen rekening gehouden. Het grote aantal mogelijke fragmentatie-ionen is immers te omvangrijk om allemaal op te nemen in een theoretisch spectrum. We zullen in de volgende sectie zien dat een zeer eenvoudig voorbeeld met enkel b- en y-ionen al uitgroeit tot een spectrum van redelijke omvang.
Vaak beperkt men zich tot alle of de meest voorkomende ion types die kunnen voorkomen in de gebruikte massaspectrometer, nl. de massaspectrometer die gebruikt werd om het experimenteel spectrum te verkrijgen dat men met het theoretisch spectrum wilt vergelijken.

Ion type	Composition	m/z ratio	δ
a	$\Sigma + H - CO$	S - 27	27
b	$\Sigma + H$	S + 1	-1
c	$\Sigma + H + NH + H + H$	S + 18	-18
x	$\Sigma + OH + CO$	S + 45	-45
y	$\Sigma + OH + H + H$	S + 19	-19
z	$\Sigma + OH - NH$	S + 21	-21
$b - H_2O$	$\Sigma + H - H_2O$	S + 1 - 18	17
$y - H_2O$	$\Sigma + OH + H + H - H_2O$	S + 19 - 18	-1
$b - NH_3$	$\Sigma + H - NH_3$	S + 1 - 17	16
$y - NH_3$	$\Sigma + OH + H + H - NH_3$	S + 19 - 17	-2
$a - NH_3$	$\Sigma + H - CO - NH_3$	S - 27 - 17	44

Tabel 2.1: Berekening van ionenmassa's [18]

Om een theoretisch spectrum te berekenen vertrekken we dus van een gekende peptide, bekomen door een proteïne uit een database een virtuele trypsinebehandeling te geven. Het doel van dit berekend spectrum is een idee te krijgen welk massaspectrum door deze peptide gegenereerd zou kunnen worden, indien het onderwerp zou zijn van een identificatie proces.
Voor de berekening hebben we de massa's nodig van de residu's van deze aminozuren. Deze zijn gekend en terug te vinden in Bijlage C. Van alle mogelijke fragmentatie-ionen van de gegeven peptide, rekening houdend met de gekozen beperking, wordt nu de massa bepaald. In de volgende sectie zullen we dit proces verduidelijken aan de hand van een voorbeeld. Het berekenen van de massa van een ion is niet zo voor de hand liggend. Simpelweg optellen van de massa's van de aminozurenresidu's is niet voldoende. Buiten het feit dat er ook rekening gehouden moet worden met de extra moleculen aan de N- of aan de *C-terminal* (zie figuur 1.3), gebeurt de vorming van ieder type fragmentatie-ion ook op een eigen, specifieke manier. Het gaat niet op om simpelweg de massa van de lading op te tellen bij de massa van de aminozurenresidu's en de extra *terminal*-atomen. Tabel 2.1 geeft weer hoe de massa's van een aantal éénwaardig geladen ion-types op een eenvoudige manier berekend kunnen worden. Het Σ-symbool staat voor alle aminozurenresidu's waar het ion uit bestaat. Symbool S staat voor de totale massa van deze aminozuren. De verhouding van de massa over de lading m/z, voor een éénwaardige lading met $z = 1$, kan berekend worden door de formule in de voorlaatste kolom toe te passen; dit zijn dus de waarden die men zou kunnen waarnemen in een experimenteel spectrum van de peptide in kwestie. De gebruikte gehele getallen zijn afrondingen. Het is voor deze tekst onnodig de massaberekening tot in detail te bespreken. Voor de geïnteresseerde lezer is een gedetailleerde uitleg van de berekening voor b- en y-ionen terug te vinden in bijlage D.
We zien in de tabel naast de 'gewone' ion types (ons reeds bekend uit figuur 1.19) ook ion types zoals $b - H_2O$. Dit zijn b-ionen die een watermolecule verloren hebben en daardoor een kleinere massa hebben. Er zijn verscheidene mogelijke wijzigingen van ionen mogelijk. In de tabel zijn er enkele opgenomen die regelmatig voorkomen.
In de laatste kolom zijn de δ-waarden weergegeven volgens de formule $m(\delta\text{-}ion) = m(P') - \delta$. In de literatuur worden verscheidene formules en daarbij horende δ-waarden gebruikt om de ion types aan te geven. In deze tekst zullen we ons houden aan deze formule die zowel voor N- als voor C-terminale ionen zal toegepast worden.

2.4.3 Voorbeeld

Vertrekkende van een gegeven peptide, gaan we nu het theoretisch spectrum hiervan berekenen. We zullen enkel rekening houden met b- en y-ionen, en we gaan in dit voorbeeld voor de eenvoud uit van een één-waardige lading op het parent ion, wat als gevolg heeft dat de resulterende fragmentatie-ionen enkel één-waardig geladen kunnen zijn. We kunnen dus gebruik maken van de gegevens in tabel 2.1. Voor de massa's van de aminozurenresidu's doen we beroep op de tabel

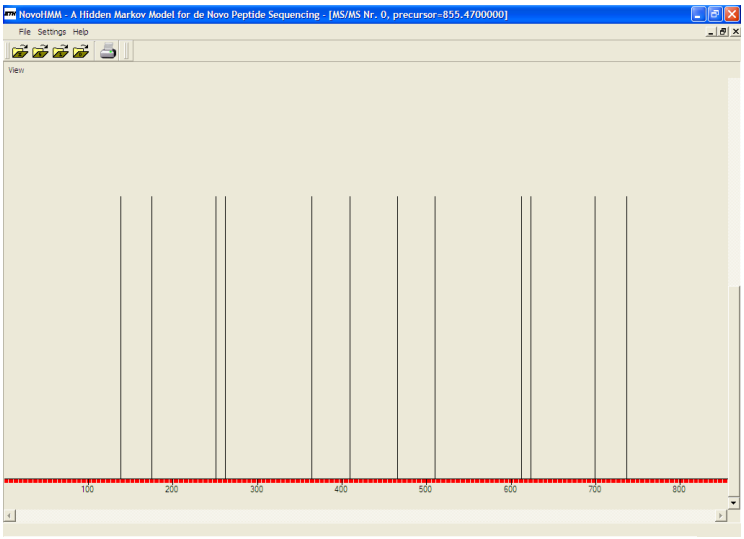
in Bijlage C. In het theoretisch spectrum worden alle mogelijke b-ionen en alle mogelijke y-ionen berekend. In een experimenteel spectrum zullen meestal niet al deze b- en y-ionen voorkomen, maar aangezien men niet weet welke wel en welke niet voorkomen houdt het theoretisch spectrum hier geen rekening mee. Er zullen echter hoogstwaarschijnlijk ook andere ion types dan b- en y-ionen voorkomen. Andere veel voorkomende wijzigingen zoals bv. het verlies van een watermolecule worden ook niet in acht genomen in dit theoretisch spectrum. Rekening houden met zelfs maar enkele van deze aspecten zou al onmiddellijk een grote uitbreiding van het theoretisch spectrum met zich meebrengen, waardoor het computationeel niet meer zomaar op een aanvaardbare manier te berekenen valt. Niet enkel de berekening van het theoretisch spectrum zal meer rekenkost vergen, ook het gebruik van dit spectrum tijdens het zoeken naar een *match* met het experimenteel spectrum zal meer tijd in beslag nemen.

- **Gegeven :**
  - peptide : HLITFSR (gebruik makende van de éénlettercode (zie figuur 1.2))
  - massa's van de aminozuren : zie Bijlage C
  - berekeningsformules voor de massa van b- en y-ionen : zie tabel 2.1

- **Mogelijke b-ionen :**
  - b1-ion : 137.06 Da voor H (Histidine) + 1 Da = 138.06 Da
  - b2-ion : 137.06 Da voor H + 113.08 Da voor L (Leucine) + 1 Da = 251.14 Da
  - enz...
  - b6-ion : 137.06 Da voor H + 113.08 Da voor L + 113.08 Da voor I + 101.05 Da voor T + 147.07 Da voor F + 87.03 Da voor S = 699.37 Da

- **Mogelijke y-ionen :**
  - y1-ion : 156.10 Da voor R (Arginine) + 19 Da = 175.1 Da
  - y2-ion : 87.03 Da voor S (Serine) + 156.10 Da voor R + 19 Da = 262.13 Da
  - enz...
  - y6-ion : 113.08 Da voor L + 113.08 Da voor I + 101.05 Da voor T + 147.07 Da voor F + 87.03 Da voor S + 156.10 Da voor R = 718.41 Da

- **Resulterend spectrum :** deze berekende fragmentatie-ionen vormen het theoretisch spectrum; we weten nu welke massa's kunnen voorkomen na fragmentatie van de peptide. Figuur 2.1 geeft dit spectrum weer, gegenereerd met NovoHMM, een software programma dat later aan bod zal komen. Er wordt enkel *binair* rekening gehouden met de massa's, m.a.w. enkel het al dan niet voorkomen van een massa wordt opgeslaan (niet de intensiteit zoals bij een experimenteel spectrum). Alle pieken hebben daarom dezelfde hoogte. Bijlage E geeft de volledige data file weer van dit theoretisch voorbeeld.
  Theoretisch gegenereerde spectra kunnen vergeleken worden met experimentele spectra, om zo tot een identificatie te komen. We spreken hier echter niet voor niets van een 'theoretisch' spectrum. Indien dezelfde ionen door een massaspectrometer gemeten zouden worden, zouden we zeker niet dezelfde massawaarden verkrijgen. Door de beperkingen van de massaspectrometer en ruis in de metingen wijken de gemeten waarden af van de theoretische waarden. Simpelweg vergelijken tussen theoretische en experimentele spectra is dus uit den boze. Ook hier wordt nog eens duidelijk dat het identificeren van peptiden een gecompliceerd proces is.



Figuur 2.1: Voorbeeld van een theoretisch spectrum, gegenereerd met NovoHMM [19]

2.4.4 Pseudo-code

Om alle algoritmen overzichtelijk te houden zullen er zowel voor de berekening van het theoretisch spectrum als voor de specifieke algoritmen in de volgende hoofdstukken telkens één of meerdere secties 'pseudo-code' zijn. Deze secties beschrijven het besproken algoritme in pseudo-code zodat de lezer een gestructureerd overzicht heeft.
Bij deze pseudo-code gaan we er vanuit dat er een datastructuur bestaat om het berekende spectrum in op te slaan.

Listing 2.2: Pseudo-code theoretisch spectrum

```
bepaal alle b-ionen van de peptide;
voor alle b-ionen do
  bereken de massa;
  if (de massa is nog niet in het spectrum aanwezig) then
    voeg de massa toe aan het spectrum;

bepaal alle y-ionen van de peptide;
voor alle y-ionen do
  bereken de massa;
  if (de massa is nog niet in het spectrum aanwezig) then
    voeg de massa toe aan het spectrum;

teken histogram;
```

2.5 Notaties en formules

Tot slot geven we in deze sectie een overzicht van notaties en formules die zullen gelden voor het hele verdere verloop van deze thesistekst.

Algemene notaties en formules :

- zij A de set van de 20 aminozuren;
- de massa van een aminozuur $p \in A$ wordt genoteerd als $m(p)$;

- een peptide $P = p_1...p_n$ is een sequentie van n aminozuren;
- de massa van de parent peptide wordt genoteerd als $m(P)$;
- $m(P) = \sum_{i=1}^{n} m(p_i)$ = de som van de massa's van de aminozurenresidu's;
- een partiële peptide $P' \subseteq P$ is een substring $p_i...p_j$ van P met massa $m(P') = \sum_{i \leq t \leq j} m(p_t)$.

Notaties en formules i.v.m. peptide fragmentatie in tandem massaspectrometrie :

- partiële N-terminale peptide $P_i = p_1...p_i \quad i = 1, ..., n-1$;
- partiële C-terminale peptide $P_j^- = p_j...p_n \quad j = 2, ..., n$;
- ion types (bv. b-ion, y-ion) worden gerepresenteerd door een δ-waarde die integere waarden aanneemt die zowel positief als negatief kunnen zijn;
- een δ-ion van een partiële peptide P' is een wijziging van P', zó dat het ion een massa heeft die gelijk is aan $m(P') - \delta$ (vb.: $\delta = -1$ voor een b-ion);
- de verzameling ion types die kunnen voorkomen in een bepaalde massaspectrometer wordt gerepresenteerd door de set $\Delta = \delta_1, ..., \delta_k$, met k het aantal verschillende iontypes die door de massaspectrometer gegenereerd kunnen worden en dus in het spectrum kunnen voorkomen.

Merk op dat zowel bij de parent peptide als bij de partiële peptiden enkel rekening gehouden wordt met de aminozuren en niet met de N- en de C-*terminal*. Bij de berekening van de massa worden de massa's van deze *terminals* dan ook weggelaten. De massa van een δ-ion wordt opgemeten door de massaspectrometer en houdt dan weer net wèl rekening met de massa van de *terminals*; de massaspectrometer meet immers het volledige fragment. De waarde van een δ houdt dus niet enkel de massaverandering in ten gevolge van de ionisatie, maar houdt ook rekening met de *terminals* (Bijlage D).

Overige notaties :

- het *theoretisch spectrum* van peptide P wordt bekomen door rekening te houden met alle mogelijke ion types $\delta_1, ..., \delta_k$ (zie ook 2.4);
- het *spectrum* $S = \{s_1, ..., s_m\}$ is een verzameling van massa's van fragmentatie-ionen, bekomen uit de metingen van de massaspectrometer en afkomstig van één parent ion;
- een *match* tussen spectrum S en peptide P is het aantal massa's dat zowel bij het experimenteel als bij het theoretisch spectrum voorkomt;
- een *compleet spectrum* S is een spectrum dat voor alle mogelijke partiële peptiden van het parent ion minstens één ion type bevat.

In de praktijk zullen niet alle N- en C-terminale partiële peptiden en niet alle types ionen voorkomen in een MS/MS spectrum. De meest voorkomende ionen zijn b- en y-ionen. Een compleet spectrum is dan ook iets wat niet in de praktijk voorkomt. We zullen bij de algoritmen zien dat het compleet spectrum vaak wordt aangewend om de algoritmen in ideale omstandigheden te omschrijven.

Deel II

De novo algoritmen voor de identificatie van peptiden

Hoofdstuk 3

De spectrumgraaf : Sherenga

Sherenga [20][21][22] is een *de novo* algoritme dat gebruik maakt van de spectrumgraaf. Een graaf is een structuur bestaande uit knopen, die verbonden zijn door gerichte of ongerichte bogen. De spectrumgraaf is een gerichte graaf die opgebouwd wordt uit de gegevens van het experimenteel spectrum.

3.1 Opstellen van de spectrumgraaf

Voor de eenvoud veronderstellen we even dat een spectrum van een tandem massaspectrometer enkel uit N-terminale ionen bestaat, waarop zich een éénwaardige lading bevindt. De aanpassingen voor een tweewaardige lading zullen, daar waar van toepassing, telkens op het einde van een sectie beschreven worden. Hogere ladingen worden niet beschreven maar hiervoor gebeuren de aanpassingen volledig analoog.
We gaan er tevens van uit dat geweten is welke ion types, en dus welke δ-waarden, voorkomen in het spectrum en dat $\Delta = \{\delta_1, ..., \delta_k\}$, met k het aantal verschillende ion types die in de massaspectrometer kunnen voorkomen. Met deze vereenvoudigingen in het achterhoofd gaan we nu de spectrumgraaf $G_\Delta(S)$ opstellen.

3.1.1 Knopen

De knopen van de graaf zijn integer waarden die *potentiële massa's* van partiële peptiden voorstellen. Deze worden bekomen door iedere piek $s \in S$ om te zetten in k knopen : $V(s) = \{s + \delta_1, ..., s + \delta_k\}$. Met piekwaarde bedoelen we de massawaarde die deze piek representeert (dus de waarde op de x-as), niet de intensiteit (of y-waarde) van de piek. Indien we over intensiteit spreken, dan wordt dit expliciet benoemd als zijnde de intensiteit van een piek. Rekening houdend met de vereenvoudiging tot enkel N-terminale ionen kunnen we de formule voor de berekening van de massa van een ion type van een bepaalde partiële peptide (sectie 2.5) als volgt schrijven :

$$m(\delta\text{-}ion) = m(P_i) - \delta \tag{3.1}$$

Oftewel :

$$m(P_i) = m(\delta\text{-}ion) + \delta \tag{3.2}$$

Aangezien de pieken s in het spectrum S de massa's van deze fragmentatie-ionen voorstellen, kunnen we de formule schrijven als :

$$m(P_i) = s_j + \delta \tag{3.3}$$

met s_j de piek uit het spectrum die overeenkomt met het δ-ion van de partiële peptide P_i.
De gemeten massawaarde van het ion wordt dus terug herleid tot de som van de aminozuren waaruit deze massa bestaat (zie notaties sectie 2.5) en wordt ook wel *residu-massa* genoemd.

Opdat deze vergelijking op zou gaan moet δ natuurlijk de waarde hebben van het ion type van de piek s_j, willen we uitkomen bij de massawaarde van de niet-geïoniseerde overeenkomstige partiële peptide. Aangezien we niet weten welk ion type een bepaalde piek representeert, is het onmogelijk om telkens net de juiste δ te gebruiken bij de berekening van de knopen. Vandaar dat er, voor een bepaalde piek s, voor elke δ een knoop wordt gemaakt. Voor een spectrum S resulteert dit in een groot aantal knopen waarvan enkel de knoop die gevormd is met de *juiste* δ ook werkelijk de partiële peptide representeert die bij de piek hoort. Toch kan het zijn dat andere knopen ook toevallig een reële massawaarde hebben van een mogelijke partiële peptide. Dit zijn de zogenaamde valse knopen. Merk op dat we nog geen rekening houden met de onnauwkeurigheden in de metingen van de massaspectrometer die nog meer aanleiding kunnen geven tot valse knopen. Alle aspecten omtrend onnauwkeurige metingen komen later aan bod in sectie 3.5.
De volledige set knopen voor een spectrum met m pieken wordt gegeven door :

$$\{s_{initial}\} \cup V(s_1) \cup ... \cup V(s_m) \cup \{s_{final}\} \tag{3.4}$$

Hierbij is $s_{initial} = 0$ en $s_{final} = m(P)$, met $m(P)$ de massa van het parent ion.

Voor tweewaardig geladen fragmentatie-ionen moeten we rekening houden met het feit dat de gegevens in het spectrum niet de massa voorstellen maar wel de massa/lading. De werkelijke massa bij tweewaardig geladen fragmentatie-ionen is dan ook dubbel zo groot dan de piekwaarde s. Om de massa van de niet-geïoniseerde N-terminale partiële peptide te achterhalen (en dus de knopen te bepalen) moet de formule lichtjes gewijzigd worden :

$$m(P_i) = 2s_i + \delta \tag{3.5}$$

Indien het spectrum zowel één- als tweewaardig geladen ionen bevat worden de knopen opgesteld door toepassing van beide formules 3.3 en 3.5. De formule voor éénwaardig geladen ionen wordt toegepast met de deelverzameling van de set Δ die éénwaardig geladen ion types representeert. De formule voor tweewaardig geladen ionen wordt toegepast met δ-waarden die tweewaardig geladen ion types voorstellen.

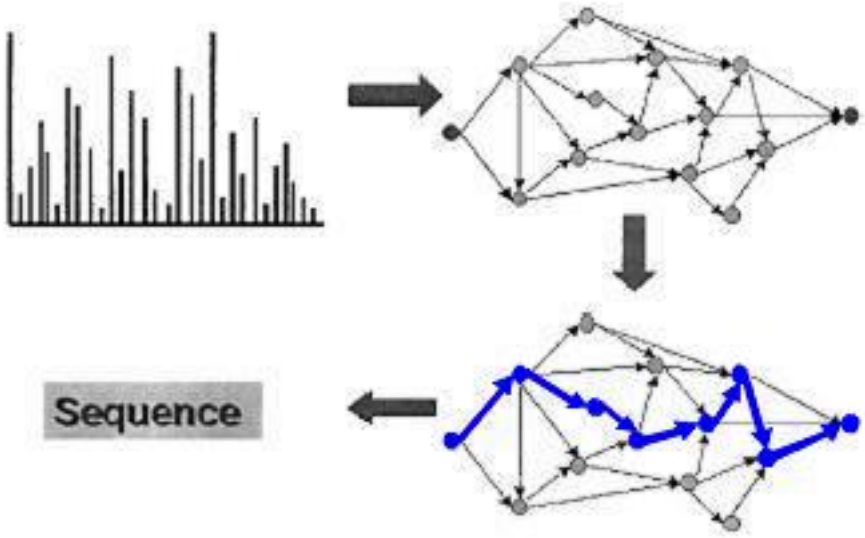
3.1.2 Bogen

Twee knopen u en v worden door een gerichte boog verbonden van u naar v indien het verschil v-u de massa is van een aminozuur. De boog wordt dan met dit aminozuur gelabeld. We hebben dus een *gerichte* graaf. Aangezien de massa van een aminozuur strikt positief is kan een boog nooit terugkeren naar een knoop met een kleinere waarde. Hierdoor is de graaf *acyclisch*.
Vermits we de vereenvoudiging hebben aangenomen dat we enkel met N-terminale ionen te doen hebben impliceert zulk een boog van u naar v dat de sequentie bij v[1] bekomen kan worden door de sequentie bij u met precies één aminozuur te verlengen, en wel met dát aminozuur waarmee de boog gelabeld is. We zullen zien dat op deze manier een pad gevormd kan worden doorheen de graaf.
Zodra de bogen getrokken zijn kan de definitieve vorm van de spectrumgraaf bepaald worden. Alle knopen die niet bereikbaar zijn van waaruit $s_{initial}$ en alle knopen vanwaar s_{final} niet bereikbaar is worden uit de graaf verwijderd. Zouden ze toch deel blijven uitmaken van de graaf, dan zou een pad ofwel nooit in zulk een knoop kunnen geraken, ofwel er nooit weggeraken. Deze knopen zijn dus zinloos voor het vinden van een pad van $s_{initial}$ naar s_{final} en worden om die reden verwijderd.

3.1.3 Pad in de spectrumgraaf

Indien het spectrum S compleet is, wat wil zeggen dat iedere mogelijke partiële peptide P_i in het spectrum aanwezig is onder de vorm van minstens één ion type, dan bestaat er in $G_\Delta(S)$ een pad van lengte n van $s_{initial}$ tot s_{final}, gelabeld met peptide P (bestaande uit n aminozuren) [20].

[1] Een knoop stelt een partiële peptide voor wat niets anders is dan een sequentie van aminozuren



Figuur 3.1: Voorbeeld : opstellen van een spectrumgraaf en het vinden van een pad [20]

Deze bewering is gebaseerd op waarnemingen en de literatuur [20][21][22] geeft hiervoor geen bewijs, maar intuïtief is deze bewering correct. Door de compleetheid is er voor iedere mogelijke N-terminale partiële peptide een knoop aanwezig in de spectrumgraaf. Voor een parent peptide bestaande uit n aminozuren zijn dit dus $n-1$ knopen die overeenkomen met de N-terminale partiële petiden P_1 tot en met P_{n-1}. Door de eigenschappen van de bogen[2] kan men dus de volledige oorspronkelijke parent peptide herconstrueren indien men het juiste pad in de graaf volgt. Het hele proces wordt schematisch weergegeven in figuur 3.1. In de onderste graaf is het gevonden pad in het blauw en vet weergegeven. Het peptide sequencing probleem is nu dus herleid tot ***het vinden van het correcte pad in een gerichte acyclische graaf***.
We zien in de tekening dat er ook andere paden te vinden zijn in de graaf. Al deze paden worden aan een *scoring* algoritme onderworpen. Het aangegeven pad is het pad met de hoogste score. Sectie 3.7 licht dit toe.

3.1.4 Voorbeeld

Vooraleer we al de vereenvoudigingen laten vallen, geven we eerst een eenvoudig voorbeeld van het opstellen van de spectrumgraaf. De gemaakte veronderstellingen zetten we nog even op een rij :

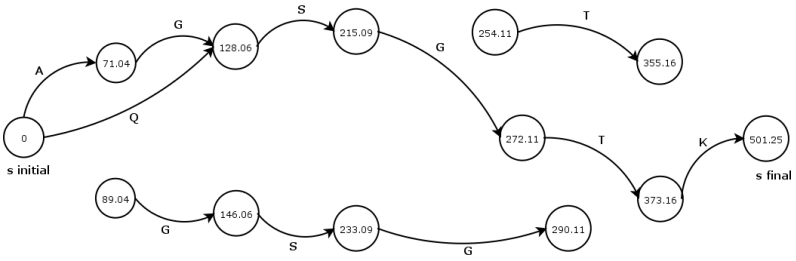
- de pieken uit het spectrum stellen N-terminale, éénwaardig geladen ionen voor;
- de voorkomende ionen zijn b-ionen en $b - H_20$-ionen wat neerkomt op de set $\Delta = \{-1, 17\}$;
- het spectrum is compleet;
- er zitten geen fouten op de meetresultaten van de massaspectrometer.

Zij S het (fictieve) experimentele spectrum in kwestie. Tabel 3.1 geeft de data weer van dit spectrum. Aangezien we in de spectrumgraaf geen rekening houden met de intensiteiten van de pieken zijn deze weggelaten uit de tabel. We kunnen aflezen dat de parent massa gelijk is aan 520,25 Da en de lading van deze parent massa +1 bedraagt. We berekenen eerst de waarde van $m(P)$: de gemeten waarde 520,25 Da $= m(P) + m(H) + m(OH) + m(lading) = m(P) + 1Da + 17Da + 1Da$. Hieruit volgt dat $m(P) = 501,25$ Da.

[2] Een boog 'volgen' komt overeen met het verlengen van de huidige sequentie met één aminozuur

	520.25	**1**
s_1	72.04	
s_2	129.06	
s_3	216.09	
s_4	255.11	
s_5	273.11	
s_6	356.16	

Tabel 3.1: Fictief voorbeeld van een MS/MS spectrum dat enkel N-terminale ionen bevat



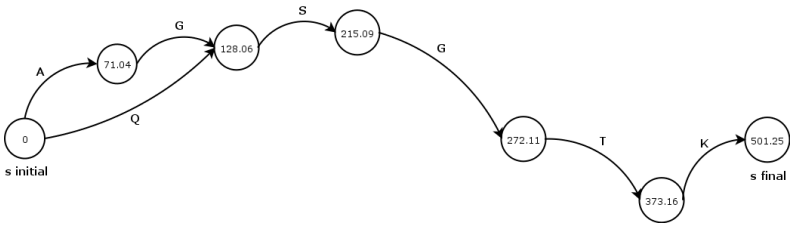
Figuur 3.2: Spectrumgraaf met alle knopen en bogen

Berekening van de knopen

We berekenen de knopenverzamelingen volgens $V(s) = \{s + \delta_1, s + \delta_2\}$ met $\delta_1 = -1$ en $\delta_2 = 17$.

- $V(s_1) = \{s_1^1, s_1^2\} = \{71.04, 89.04\}$
- $V(s_2) = \{s_2^1, s_2^2\} = \{128.06, 146.06\}$
- $V(s_3) = \{s_3^1, s_3^2\} = \{215.09, 233.09\}$
- $V(s_4) = \{s_4^1, s_4^2\} = \{254.11, 272.11\}$
- $V(s_5) = \{s_5^1, s_5^2\} = \{272.11, 290.11\}$
- $V(s_6) = \{s_6^1, s_6^2\} = \{355.16, 373.16\}$

We zien in de knopenverzamelingen dat de waarde van knoop s_4^2 gelijk is aan die van knoop s_5^1. Voor deze waarde wordt er dus slechts één knoop aangemaakt. Deze waarde impliceert dat de partiële peptide in kwestie onder twee (in dit voorbeeld dus alle) ion types voorkomt die in het spectrum voorkomen.
Nu de knopen bepaald zijn kunnen we de bogen trekken. Figuur 3.2 geeft de spectrumgraaf weer met alle berekende knopen en de mogelijke bogen. Opdat de lezer de knopen eenvoudig kan verifiëren met de knopenverzamelingen zijn de knopen die overeenkomen met de eerste $\delta = -1$ bovenaan in de graaf weergegeven, de andere knopen onderaan. De zonet vermelde knoop met waarde 272.11 wordt in het midden weergegeven. Figuur 3.3 geeft de graaf weer zonder de knopen en bogen die niet volledig in de graaf opgenomen konden worden. In deze finale graaf zien we dat er twee mogelijke paden zijn van $s_{initial}$ naar s_{final}. Zowel de sequentie $AGSGTK$ als $QSGTK$ zijn paden doorheen de graaf. Om te bepalen welke sequentie het spectrum identificeert gaat men op zoek naar het *langste pad*. "Het langste pad is het pad dat overeenkomt met een peptide P die het spectrum het beste *verklaart*" [20]. In dit voorbeeld is $AGSGTK$ de peptide die spectrum S heeft voortgebracht. We gaan hier nu niet verder op in. In sectie 3.7 zullen we zien dat er aan elke knoop een score gegeven wordt. Aan de hand van deze scores wordt telkens de score van het gevonden pad berekend. Het pad met de hoogste score is het zogenaamde *langste pad*.
Het langste pad in een gerichte acyclische graaf is een gekend probleem in de theoretische informatica, en kan opgelost worden door een snel, lineaire tijd, dynamisch algoritme [23]. We zullen



Figuur 3.3: Finale spectrumgraaf waaruit de paden afgeleid kunnen worden

zien (sectie 3.8) dat dit algoritme spijtig genoeg niet kan toegepast worden in de praktijk. Zelfs het gevonden langste pad zal niet altijd de juiste identificatie weergeven (hoofdstuk 4). Op dit moment hoeven we ons hier echter nog geen zorgen over te maken, en gaan we er van uit dat er een algortime bestaat om het langste pad te zoeken in onze spectrumgraaf.
We hebben voor dit voorbeeld een aanzienlijk aantal vereenvoudigingen gemaakt en hebben een MS/MS spectrum geïdentificeerd met een minieme hoeveelheid *correcte* data. De volgende sectie overloopt alle vereenvoudigingen en haalt de oplossingen aan die Sherenga hiervoor gebruikt. Vanaf sectie 3.2 worden deze technieken gedetailleerd besproken. Vaak worden er meerdere langste paden gevonden, waardoor een *scoring algoritme* noodzakelijk is (sectie 3.7). Tot slot (sectie 3.8) komt het algoritme aan bod dat de paden zoekt doorheen de spectrumgraaf.

3.1.5 Tekortkomingen

Er zijn tot hier toe al heel wat vereenvoudigingen gebeurd om de spectrumgraaf te kunnen opstellen. Bij het introduceren van het compleet spectrum zal de lezer onmiddellijk de bemerking hebben gemaakt dat dit niet in overeenstemming is met wat voordien gezegd is geweest over de output van massaspectrometers. Experimentele spectra zijn niet enkel niet-compleet, er is immers geen enkele garantie dat er minstens één ion in het spectrum aanwezig is van alle mogelijke partiële peptiden, maar bovendien bevatten spectra een heleboel ruis die kan leiden tot zogenaamde 'valse' knopen en bogen, die toch in de finale graaf opgenomen kunnen worden. De bogen worden immers enkel bepaald op basis van de aminozurenmassa's, en de kans dat twee knopen toevallig in net zo'n aminozuur verschillen is zeer reëel. De massawaarden van de aminozuren liggen ook vaak zeer dicht bij elkaar (zie bijlage C) waardoor er al gauw een verkeerd label aan de bogen kan gegeven worden.
Door het niet compleet zijn van spectra kan een pad al niet meer op de eenvoudige manier gevonden worden. Er zullen dan knopen ontbreken (niet-aanwezige fragmentatie-ionen in het spectrum) waardoor de wél gevonden knopen onderling niet meer in de massa van slechts één enkel aminozuur gaan verschillen. Vooraleer de overtollige knopen en bogen verwijderd worden, zal de graaf daarom eerst vervolledigd worden met *gap* en *bridge edges*.
Een ander groot probleem is dat verschillende soorten massaspectrometers voor éénzelfde peptide heel andere spectra als output kunnen geven. De gebruikte ionisatiemethode en de manier waarop de massa analyse gebeurt heeft een grote invloed op de output, vermits deze leiden tot andere types fragmentatie-ionen. Iedere massaspectrometer heeft, naast het vormen van andere fragmentatie-ionen, zijn eigen set Δ. Om niet instrument-afhankelijk te zijn heeft Sherenga de *offset frequentie functie* gedefinieerd om de set Δ van een bepaalde massaspectrometer te achterhalen (sectie 3.2). Aan de hand van deze offset frequentie functie worden ook *intensiteit thresholds* bepaald. Op basis daarvan wordt bepaald welke pieken uit het spectrum als ruis en welke als spectrumgegevens beschouwd worden. Enkel deze laatste zullen gebruikt worden om de spectrumgraaf op te bouwen.
Tot slot moet men in de praktijk wel rekening houden met de C-terminale ionen; een experimenteel spectrum bevat immers de massa's van zowel N-terminale als C-terminale ionen. Theoretisch gezien zou men, met behulp van de parent massa, de massa van deze C-terminale ionen kunnen omrekenen naar de massa van het complementaire N-terminale ion, zodat men na deze vereenvoudiging enkel met N-terminale ionen kan verderwerken. Het is echter niet gekend welke massagegevens uit het spectrum betrekking hebben op N-terminale ionen en welke op C-terminale

ionen, en welk ion type gepaard gaat met een bepaalde piek. In de praktijk is deze berekening daarom niet zo vanzelfsprekend. Bovendien zal de waarde van de parent massa, door de beperkingen van de massaspectrometer, afwijken van zijn theoretische (correcte) massawaarde, net als de massawaarden van de ionen. De berekening zou dan op basis van deze onnauwkeurige gegevens moeten gebeuren waardoor de fout op de omgerekenende waarden aanzienlijk groot wordt. Secties 3.4 en 3.6 beschrijven de gebruikte oplossingen hiervoor.

3.1.6 Overzicht

De hierboven vermelde oplossingen voor de tekortkomingen zullen voor, tijdens en na het opstellen van de graaf plaatsvinden. Om de lezer niet in verwarring te brengen volgt eerst een schematisch overzicht van het opstellen van de spectrumgraaf hoe dit in de praktijk gebeurt (zonder vereenvoudigingen). De onbekende begrippen zullen vanaf de volgende sectie uitgebreid besproken worden. De lezer kan altijd teruggrijpen naar dit schema om de beschreven technieken te situeren. Stappen één tot en met vier zijn bepalend voor de knopen van de graaf. Stappen vijf en zes voor de bogen. Alle knopen en bogen die op dit punt niet voldoen aan de omschrijving in sectie 3.1.2 maken geen deel uit van de finale spectrumgraaf. Tot slot worden de paden doorheen de graaf bepaald en wordt hun score berekend.

1. Offset frequentie functie : bepaling van de set Δ
2. Intensiteit thresholds
3. C-terminale ionen met herberekening van de parent massa
4. Merge algoritme
5. Bogen
6. Gap en bridge edges
7. Zoeken naar paden en het scoring algoritme

3.2 De offset frequentie functie

Eén van de aannames die we in het begin gemaakt hebben is de kennis van de set Δ, de ion types die voorkomen in het te identificeren, experimenteel spectrum. Aangezien deze set voor zowat iedere massaspectrometer anders is, zijn *de novo* algoritmen vaak specifiek voor bepaalde massaspectrometers bedoeld. Sherenga echter heeft een functie die kan achterhalen welke ion types door een bepaalde massaspectrometer geproduceerd worden. Hierdoor kan het algoritme zonder problemen op gegevens van alle soorten massaspectrometers toegepast worden. De enige vereiste is een *training set*. Een *training set* is een verzameling geïdentificeerde experimentele spectra, in dit geval afkomstig uit de massaspectrometer waarvoor men de set Δ wilt bepalen.

3.2.1 Basis

Vertrekkende met een *training set* is het mogelijk om de set Δ te leren kennen. Eens deze set gekend is kan men hiermee verder werken om de spectrumgraaf op te stellen en ongekende sequenties, bekomen na analyse door zulk een massaspectrometer, te identificeren. Voor de eenvoud gaan we nog steeds uit van een éénwaardige lading op N-terminale fragmentatie-ionen.
Gegeven P en S_t, met S_t één spectrum uit de *training set* en P de identificatie van dit spectrum, definiëren we de *offsets*

$$x_{ij} = m(P_i) - s_j \tag{3.6}$$

en nauwkeurigheid ϵ, die meestal gelijkgesteld wordt aan 0,5. P_i stelt de N-terminale partiële peptiden van de gekende sequentie P voor. De massa's van deze partiële peptiden kunnen berekend worden met behulp van de tabel in bijlage C. De massawaarden uit het spectrum S_t worden voorgesteld door s_j.

3.2.2 Offset frequentie functie *H(x)*

Vertrekkende van de gegevens uit voorgaande sectie kunnen we nu de offset frequentie functie definiëren :

- zij x de variabele van de functie H die enkel integere waarden aanneemt;
- zij $\hat{H}(x, S_t)$ gelijk aan het aantal (P_i, s_j) zó dat de corresponderende offset x_{ij} binnen een nauwkeurigheid ϵ van x ligt.

Dan definiëren we de offset frequentie functie als volgt :

$$H(x) = \sum_{S} \hat{H}(x, S_t) \tag{3.7}$$

De *offsets* representeren de mogelijke iontypes, dus de δ-waarden die in de *training set* voorkomen. Dit is gemakkelijk te zien indien we de formules 3.3 en 3.6 met elkaar vergelijken. De waarde van $\hat{H}$ is niets anders dan het aantal ionen van het type $\delta = x$ dat aanwezig is in spectrum S_t.
De waarde van $\hat{H}$ zal, voor de x-waarden die overeenkomen met δ-waarden van ion types die voorkomen in het spectrum, veel groter zijn dan de waarde van $\hat{H}$ op willekeurige waarden van x. Voor willekeurige waarden van x zal de functie $\hat{H}$ toch waarden hebben groter dan nul, een gevolg van de aanwezige ruis in het spectrum. Deze waarden zijn echter verwaarloosbaar klein ten opzichte van de telling bij de 'echte' δ-waarden voor x. Als we de functie $\hat{H}$ uittekenen in functie van x, dan zullen er op de x-waarden die overeenkomen met de δ-waarden van aanwezige ion types duidelijke pieken zichtbaar zijn. Figuren 3.4 en 3.5 zijn hier voorbeelden van.

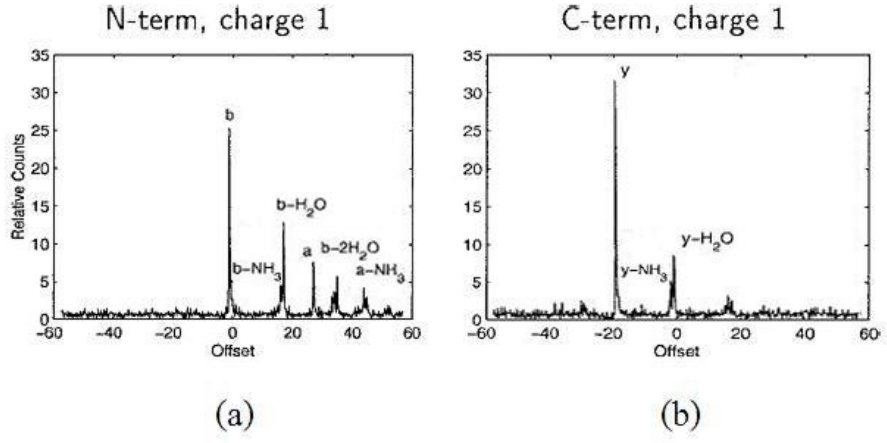
Offset	Integer offset	Count	Terminal	Ion
-18.85	-19	604	C	y
-0.85	-1	568	N	b
17.05	17	338	N	$b - H_2O$
-0.90	-1	248	C	$y - H_2O$
27.15	27	204	N	a
-20.05	-20	183	C	y^2
16.15	16	159	N	$b - NH_3$
-1.90	-2	131	C	$y - NH_3$
35.20	35	151	N	$b - H_2O - H_2O$
34.20	34	134	N	$b - H_2O - NH_3$
44.25	44	129	N	$a - NH_3$
45.15	45	107	N	$a - H_2O$
-2.30	-2	102	C	$y^2 - H_2O$
16.10	16	97	C	$y - H_2O - NH_3$
17.15	17	91	C	$y - H_2O - H_2O$

Tabel 3.2: Voorbeeld : gegevens verkregen uit een *training set* met behulp van de offset frequentie functie [20]

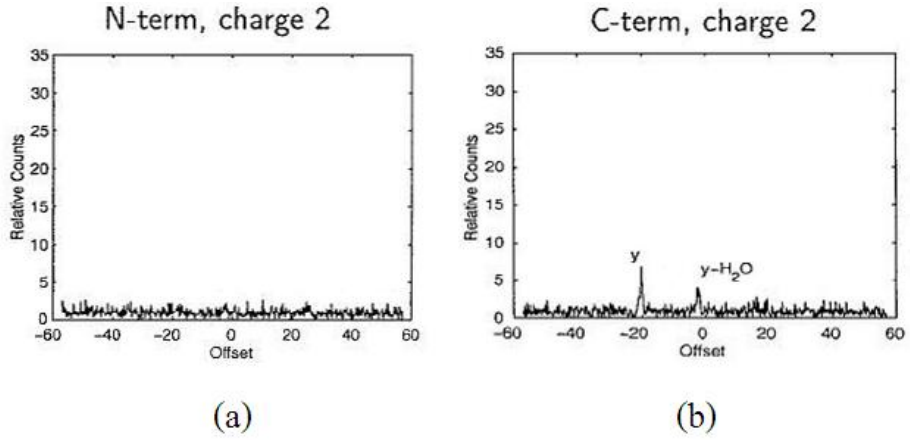
De offset frequentie functie is niets meer dan een sommatie van de functie $\hat{H}$ over alle spectra S_t van de *training set*. Door het gebruik van meerdere spectra zullen de pieken van veel voorkomende ion types bij elkaar opgeteld worden waardoor ze nog meer zichtbaar worden. Voor minder vaak voorkomende ion types zullen de kleinere pieken groter worden in verhouding tot de ruispieken (die willekeuriger zijn en daardoor niet noodzakelijk dezelfde offsets genereren in de verschillende spectra), waardoor ook deze ion types beter te onderscheiden zijn. Het resultaat van een toepassing van de offset frequentie functie wordt weergegeven in tabel 3.2. Deze tabel geeft, naast de x-waarde (*integer offset*), ook de werkelijk gemeten offset weer[3]. Verder bevat de tabel, naast het

[3] De nauwkeurigheid ϵ werd gebruikt om het verschil in deze waarden (meetonnauwkeurigheden in het spectrum) op te vangen.

aantal ionen gevonden voor een bepaalde x (*count*), ook gegevens over het ion type en of het gaat om N-terminale of C-terminale ionen. Hoe de waarden voor C-terminale ionen bekomen werden komt aan bod in sectie 3.4.2.
Enkel de waarden van de telling die overeenstemmen met bepaalde berekende offsets zijn gegevens die voortvloeien uit de offset frequentie functie. De overige informatie is toegevoegd en zijn gekende gegevens die overeen komen met de integer offset in kwestie. In de tabel staan ook gegevens voor twee tweewaardig geladen ionen, y^2 en $y^2 - H_2O$. Hier komen we later op terug.
Figuur 3.4 (a) geeft van dit voorbeeld een grafiek weer voor de ion types van N-terminale partiële peptiden (met lading 1). In de grafiek zijn een aantal pieken benoemd. Hierin zien we goed het verschil in intensiteit tussen de verschillende ion types, en de ruispieken. Voor de volledigheid is ook de grafiek van de C-terminale partiële peptiden te zien in figuur 3.4 (b). Wat opvalt is de uitgesproken aanwezigheid van y-ionen, en het feit dat er slechts twee andere ion types gevonden zijn, in tegenstelling tot de verscheidenheid aan ion types voor N-terminale partiële peptiden, wat meteen verklaart waarom algoritmen, zoals Sherenga, als vereenvoudiging de beperking tot N-terminale ionen of tot enkel b- en y-ionen aannemen.



Figuur 3.4: Voorbeeld : grafieken van een offset frequentie functie (éénwaardige lading) [20]



Figuur 3.5: Voorbeeld : grafieken van een offset functie (tweewaardige lading) [20]

Voor ionen met een tweewaardige lading wordt de functie $\hat{H}(x, S_t)$ omgezet in : $\hat{H}^{+2}(x, S_t)$ met $x_{ij} = m(P_i) - 2s_j$. Figuren 3.5 (a) en (b) geven respectievelijk de offset frequentie functies weer

voor N-terminale en C-terminale ionen met een tweewaardige lading. We zien duidelijk dat de gebruikte ionisatie methode in de massaspectrometer vooral éénwaardig geladen ionen heeft voortgebracht. Ionen y^2 en $y^2 - H_2O$ in tabel 3.2 geven de meetresultaten voor tweewaardig geladen ionen weer. Voor nog hogere ladingen gaan we analoog te werk.

Zowel uit de tabel als uit de grafieken is het duidelijk leesbaar welke ion types door de gebruikte massaspectrometer voortgebracht worden. Dankzij de offset frequentie functie hebben we deze ion types gemakkelijk kunnen achterhalen op basis van een *training set*, waardoor het algoritme Sherenga machine onafhankelijk is geworden. De gevonden set Δ kan nu verder gebruikt worden om de graaf op te stellen.

3.2.3 Pseudo-code

We beperken de pseudo-code tot éénwaardige ladingen op de ionen.

Listing 3.1: Pseudo-code Sherenga : offset frequentie functie

```
voor alle spectra S in de training set do
  voor alle partiële N-terminale ionen do
    voor alle x variërend met integere waarden do
      bereken H(x,S);
      H(x) = H(x) + H(x,S);
  voor alle partiële C-terminale ionen do
    voor alle x variërend met integere waarden do
      bereken H(x,S);
      H(x) = H(x) + H(x,S);
```

3.3 Intensiteit thresholds

Pieken in een spectrum hebben verschillende intensiteiten. Er is echter ook een hoeveelheid ruis aanwezig in zo'n spectrum. Om een onderscheid te maken tussen deze twee kan men thresholds instellen. Het kiezen van de thresholds is echter niet voor de hand liggend. Zijn de thresholds te laag, dan leidt dit tot een extreme groei van de spectrumgraaf. Zijn ze langs de andere kant te hoog, dan zullen we niet meer in staat zijn om de spectrumgraaf volledig op te bouwen. Indien er te weinig pieken in beschouwing genomen worden kan het immers zijn dat er veel relevante informatie verloren gaat en er hierdoor onvoldoende bogen getrokken kunnen worden in de spectrumgraaf. Hierdoor kunnen 'gaten' onstaan in de graaf waardoor er geen pad kan gevonden worden.
Het bepalen van intensiteit thresholds komt niet enkel in dit algoritme voor. De meeste algoritmen zijn afgestapt van het idee van één globale threshold en gebruiken lokale thresholds. Sherenga echter bepaalt de intensiteit thresholds met behulp van de offset frequentie functie en aan de hand van een experimenteel vastgesteld feit : de intensiteit van de pieken in een spectrum hangt samen met de ion types. We weten dat bv. b- en y-ionen het meeste voorkomen. Vanzelfsprekend zullen de intensiteiten van pieken die b- of y-ionen voorstellen groter zijn dan van andere ion types. Sherenga heeft een methode ontwikkeld om de threshold op een zeer specifieke manier te bepalen naargelang het ion type in kwestie.
Om te weten te komen welke ion types in welke intensiteiten voorkomen, werken we verder met de *training set* en de offset frequentie functie. Gegeven een spectrum S_t, worden de pieken gegroepeerd in *bins* met grootte $K = (parentmassa)/100$ volgens hun intensiteit. De K pieken met grootste intensiteit krijgen rang 1, de volgende K pieken rang 2, enz. Door dit toe te passen op de spectra uit de *training set* heeft men opgemerkt dat de offset frequentie functie, in functie van de rang, een zeer interessant verloop vertoont. Ion types komen vooral voor in enkele aaneensluitende rangen,

maar zodra de offset frequentie functie in een lagere rang komt, daalt de grafiek zeer snel, wat inhoudt dat het ion type niet veel voorkomt in deze kleinere intensiteiten. Er is dus een duidelijk verband tussen de rang of intensiteit van een piek en het bijhorende ion type. Sherenga gaat dit verband benutten om betere thresholds te gebruiken. Aangezien de offset frequentie functie betrekking heeft op *alle* spectra uit de *training set* worden al deze spectra opgedeeld in bins zodat ook uit alle spectra de gegevens omtrent het verband tussen ion type en rang benut kunnen worden.

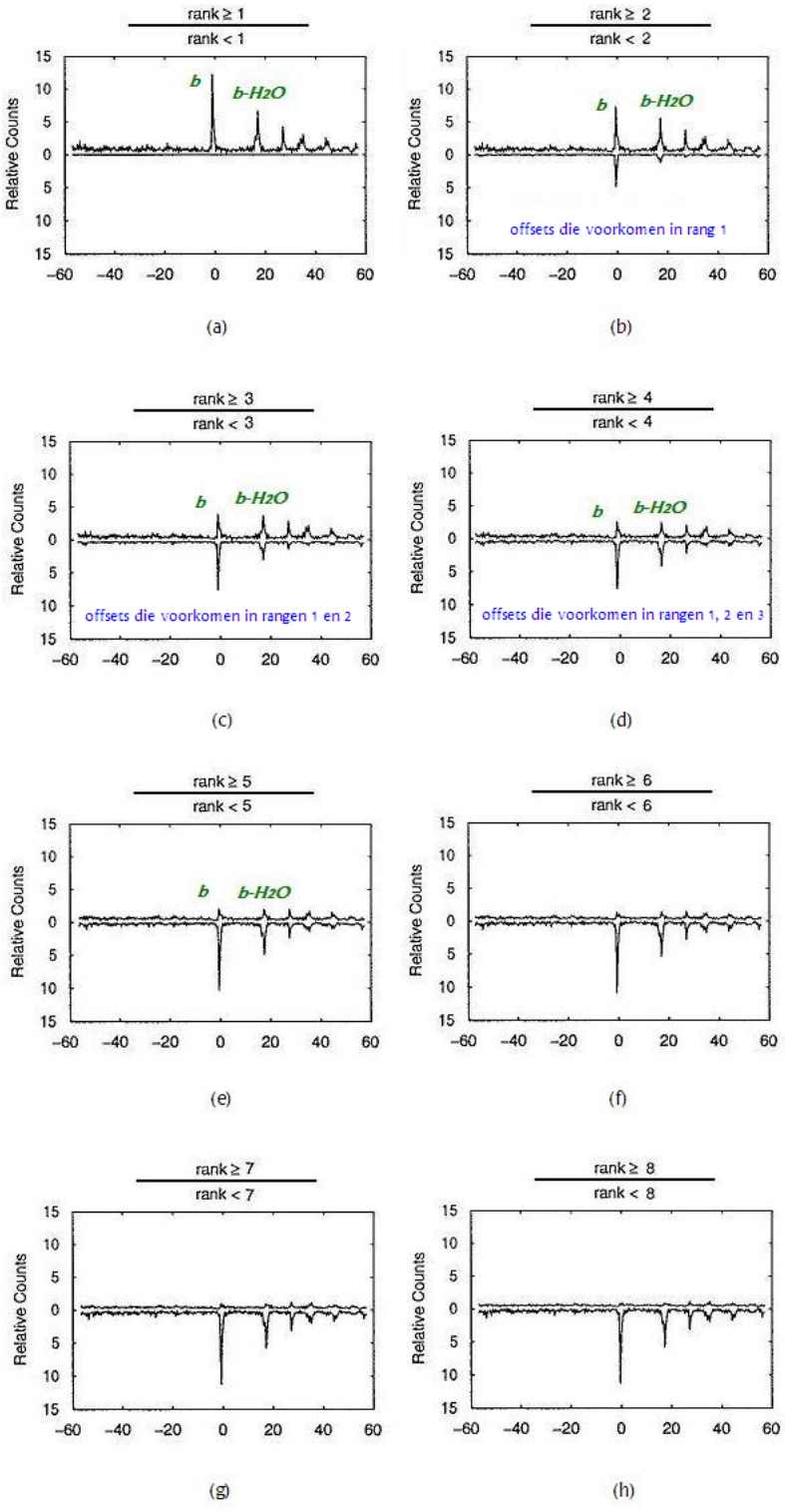
3.3.1 De offset frequentie functie in functie van de rang

Figuur 3.6 geeft voor iedere rang i de bijhorende offset frequentie functie weer voor N-terminale, éénwaardig geladen ionen (we werken nog steeds met hetzelfde voorbeeld). Het bovenste gedeelte van de grafieken geeft de offset frequentie functie weer voor ionen met een rang $\geq i$, het onderste gedeelte voor de ionen met rang $< i$. In deze figuur kunnen we zien dat bij intensiteiten met een rang lager dan 5, de offset frequentie functie hiervoor (het bovenste gedeelte) zo goed als geen pieken vertoont. Dit wil zeggen dat de pieken die in het spectrum een rang lager dan 5 kregen zo goed als geen bijdrage leverden tot de telling van de ion types voor de offset frequentie functie. Deze pieken zullen heel waarschijnlijk geen data voorstellen maar ruis. We kunnen de knopen van de spectrumgraaf daarom al beperken tot de pieken uit rangen één tot en met vijf.

3.3.2 Verband tussen ion types en intensiteit

Uit de grafieken kunnen we ook afleiden welke ion types in welke rangen voorkomen. Het onderste deel van de grafieken kunnen we als volgt beschouwen. Neem als voorbeeld grafiek (b) waarin het onderste gedeelte de offset frequentie functie voorstelt voor de rangen hoger dan twee, oftewel de offset frequentie functie beperkt tot rang 1. Analoog voor grafiek (c) onderaan waarin de offset frequentie functie beperkt wordt tot de rangen één en twee.
Om nu te bepalen welke ion types in welk rangen voorkomen wordt er gekeken naar de bijdrage van een bepaalde rang tot de piek in het onderste deel van de grafieken. Als voorbeeld zullen we de ion types b en $b - H_2O$ beschouwen (deze pieken zijn aangeduid op de grafieken). Voor alle andere N-terminale ion-types verloopt dit analoog. Voor de C-terminale ionen werken we op dezelfde manier, maar dan uiteraard met de offset frequentie functie van de C-terminale ionen.
Voor b-ionen[4] zien we dat de piek in het onderste gedeelte sterk toeneemt op de grafieken (b) tot en met (d). Voor grafiek (b) kunnen we de 'groei' van de piek in de onderste grafiek als volgt formuleren : het aantal b-ionen dat geteld werd onder pieken met rang 1. De grote toename impliceert dat er in de rangen één, twee en drie veel b-ionen aanwezig zijn in de spectra. Vanaf grafiek (e) neemt de piek zo goed als niets meer toe, zeker indien we de toename vergelijken met de vorige grafieken. Pieken uit het spectrum met rang vier zullen dus niet veel b-ionen meer bevatten. Om de spectrumgraaf op te bouwen zullen we enkel voor pieken s uit de rangen één, twee en drie de knoop berekenen met $\delta = -1$, wat overeenkomt met de δ voor het b-ion.
Doen we hetzelfde voor ion type $b - H_2O$, dan bekomen we een ander resultaat. Op het onderste gedeelte van grafiek (b) zien we voor dit ion type slechts een hele kleine waarde van de offset frequentie functie verschijnen. We noemen dit hier bewust geen piek, want er is nog geen piek te onderscheiden van de omringende ruis. In grafiek (c) is er slechts een kleine piek zichtbaar. Vanaf grafiek (d) echter is er een duidelijke toename van de piekwaarde voor dit ion type. We kunnen dus stellen dat vooral pieken uit de rangen drie, vier en vijf ionen van het type $b - H_2O$ representeren. Deze methode hangt sterk af van de kwaliteit van de spectra uit de *training set* en is min of meer subjectief in het beoordelen van het verband tussen rang en ion type. Deze manier van werken is echter meer regel dan uitzondering in dit vakgebied, daar men niet kan terugvallen op exacte gegevens omwille van de beperkingen die de huidige massaspectrometers hebben.

[4] De waarden van de offsets (of δ-waarden) worden weergegeven op de horizontale as. Voor b-ionen is deze gelijk aan -1.



Figuur 3.6: Offset frequentie functies voor de verschillende rangen van intensiteiten [20]

3.3.3 Voor- en nadelen voor de spectrumgraaf

Na het bepalen van de intensiteit thresholds kunnen we de spectrumgraaf opbouwen door in de eerste plaats enkel de pieken die tot de rangen 1 tot en met 5 behoren om te zetten in knopen. Vervolgens worden de thresholds toegepast, afhankelijk van het ion type. Beschouwen we nu een piek s uit het spectrum die we willen omvormen tot knopen. Stel dat deze piek tot rang twee behoort. Dan weten we nu dat we zeker de δ-waarde moeten optellen die overeenstemt met het b-ion, en niet de δ-waarde die overeenstemt met $b - H_2O$ ionen. Voor alle andere ion types gaan we analoog te werk. Op deze manier kunnen veel onnodige knopen in de spectrumgraaf vermeden worden. De kans dat een piek uit rang één overeenkomt met een $b-H_2O$ ion is zeer klein. Het is dan ook niet nodig om de graaf te overladen met knopen die toch hoogstwaarschijnlijk niet de massa van een reële partiële peptide representeren. Deze knopen zullen toch niet in de graaf opgenomen worden of kunnen leiden tot valse knopen en/of bogen. Niet correcte gegevens opnemen in de graaf is uiteraard niet aangewezen en wordt op deze manier zo goed als het kan vermeden.
Langs de andere kant heeft deze werkwijze het nadeel dat, indien een bepaald fragmentatie-ion (bv. het b_2-ion) ondergerepresenteerd is in het spectrum, de overeenkomstige partiële peptide ook niet in de graaf aanwezig zal zijn. In het voorbeeld beperken we de b-ionen tot de rangen 1, 2 en 3. Het ondergerepresenteerde b_2-ion zal bijvoorbeeld enkel in rang 5 voorkomen en gaat op deze manier verloren.

3.3.4 Pseudo-code

Listing 3.2: Pseudo-code Sherenga : intensiteit tresholds

```
K = (parent massa)/100;
doorloop de pieken van spectrum S van grootste naar kleinste intensiteit
  verdeel de pieken in bins van grootte K volgens intensiteit;
voor alle rangen i
  bereken de offset frequentie functie in functie van de rangen;
bepaal hieruit de globale threshold;
voor ieder ion type do
  bepaal binnen deze globale threshold de intensiteit thresholds
  voor dit ion type;
```

3.4 C-terminale ionen

Tot nu toe hebben we steeds aangenomen dat we enkel werkten met N-terminale ionen. Een reëel spectrum bevat echter zowel N-terminale als C-terminale ionen. Eerder werd al aangehaald dat het omrekenen van de C-terminale ionen naar N-terminale ionen niet zo vanzelfsprekend is (sectie 3.1). Deze sectie zal de aanpassingen toelichten die de spectrumgraaf moet ondergaan.
We vergeten even de informatie die de offset frequentie functie en de intensiteit thresholds ons verschaffen en gaan er van uit dat het spectrum enkel b- en y-ionen bevat. We maken dus geen onderscheid tussen rangen, we zullen enkel een onderscheid maken tussen δ-waarden die resp. overeenstemmen met N-terminale ionen en met C-terminale ionen. Secties 3.4.2 en 3.4.3 zullen toelichten hoe de informatie van de offset frequentie functies en de intensiteit thresholds gecombineerd kan worden met C-terminale ionen.

3.4.1 Knopen voor C-terminale ionen

In de eerste plaats weten we niet welke pieken overeenstemmen met N-terminale en welke met C-terminale ionen. Dit is een probleem aangezien we door de aanname van enkel N-terminale ionen op

een eenvoudige manier een pad kunnen vinden doorheen de graaf. Daarom wordt iedere piek eerst beschouwd als N-terminale ion en vervolgens als de complementaire C-terminale ion. Zij s een piek uit het spectrum, en $m(P)$ de parent massa. We beschouwen de piek s eerst als b-ion en weten reeds dat dit zal leiden tot de vorming van een knoop met massawaarde $s + \delta_b = s - 1$, de residu-massa van het door s gerepresenteerde ion. Beschouwen we vervolgens de piek als y-ion, dan willen we dit ion omvormen tot een knoop die als waarde de residu-massa van de complementaire b-ion voorstelt. Hiertoe wordt er een knoop gevormd met massawaarde $m(P) - (s + \delta_y) = m(P) - (s - 19)$. Ieder aanwezig ion in het spectrum wordt hierdoor zowel door de juiste als door een 'verkeerde' partiële peptide toegevoegd in het spectrum. Deze verkeerde knoop wordt de *fake twin vertex* genoemd. Om dit te illustreren geven we een theoretisch voorbeeldje.

Zij $HA_1A_2A_3A_4A_5OH$ de voorstelling van de parent peptide, met A_i de aminozurenresidu's in de peptide. $m(P)$ stelt de som van de massa's van de aminozuren voor en is dus gelijk aan de massa van $A_1A_2A_3A_4A_5$. Er zijn twee mogelijkheden; de piek in kwestie stelt een b-ion voor, of een y-ion. We weten dat een H-molecule 1 Da weegt, en een O-molecule 16 Da.

- s=b-ion : HA_1A_2
  - beschouw s als een b-ion :

$$\begin{aligned} m(\text{knoop}) &= m(HA_1A_2) - 1 \\ &= m(A_1A_2) \\ &= N\text{-terminale peptide} \\ &= \textbf{juiste knoop} \end{aligned}$$

  - beschouw s als een y-ion :

$$\begin{aligned} \text{knoop} &= m(A_1A_2A_3A_4A_5) - (m(HA_1A_2) - 19) \\ &= m(A_1A_2A_3A_4A_5) - m(A_1A_2) - (m(H) - 19) \\ &= m(A_3A_4A_5) + 18 \\ &= \text{complementaire } C\text{-terminale peptide} + 18Da \\ &= \textbf{\textit{fake twin vertex}} \end{aligned}$$

- s=y-ion[5] : HHA_5OH
  - beschouw s als een b-ion :

$$\begin{aligned} \text{knoop} &= m(HHA_5OH) - 1 \\ &= m(A_5) + m(H_2O) \\ &= m(A_5) + 18Da \\ &= C\text{-terminale peptide} + 18Da \\ &= \textbf{\textit{fake twin vertex}} \end{aligned}$$

  - beschouw s als een y-ion :

$$\begin{aligned} \text{knoop} &= m(A_1A_2A_3A_4A_5) - (m(HHA_5OH) - 19) \\ &= m(A_1A_2A_3A_4A_5) - m(A_5) - (m(HH_2O) - 19) \\ &= m(A_1A_2A_3A_4) \\ &= \text{complementaire } N\text{-terminale peptide} \\ &= \textbf{juiste knoop} \end{aligned}$$

[5] Zie bijlage D voor meer uitleg over de vorming van y-ionen.

We zien dat er voor iedere knoop ook een *fake twin vertex* aangemaakt wordt. Helaas weten we niet welke van de twee knopen de juiste is, en welke de *fake twin vertex*. De lezer zal opgemerkt hebben dat de *fake twin vertices* niet de massa voorstellen van de som van de aminozuren, maar dat deze knoop 18 Da *te zwaar* is. Men zou denken dat deze knopen daarom toch niet in het pad zouden opgenomen worden. Helaas kunnen er heel wat aminozuren gevonden worden die in massa 18 Da van elkaar verschillen, zelfs met een tolerantie kleiner dan 0,5 Da. Een voorbeeld hiervan zijn Proline (P) en Aspartic Acid (D), die 18,3 Da in massa verschillen. Met de toegelaten tolerantie ϵ bij het berekenen van knopen en bogen, en de onnauwkeurigheden in de meetresultaten, zijn deze knopen niet langer een getal zonder betekenis, maar wel een aminozurensequentie die een *ander* aminozuur bevat dan in de oorspronkelijke sequentie. Zonder al te veel moeite kunnen er dus bogen getrokken worden tussen *fake twin vertices*, maar ook tussen juiste knopen en *fake twin vertices*. Een pad doorheen de graaf dat zowel de juiste als de verkeerde knoop van éénzelfde piek bevat kan uiteraard geen goede identificatie zijn voor de parent peptide. Tijdens het opstellen van de graaf wordt er daarom een lijst aangemaakt die deze ***verboden paren*** bevat. We moeten ons langste pad probleem nu herformuleren tot :

Het langste pad in een gerichte acyclische graaf met een set verboden paren.

Dit is echter een NP-hard probleem [24], wat inhoudt dat het niet in polynomiale tijd opgelost kan worden. Hoe men toch tot een haalbaar algoritme is gekomen wordt uitgebreid besproken in sectie 3.8 en hoofdstuk 4.

3.4.2 De offset frequentie functie voor C-terminale ionen

Indien we ook willen weten welke ion types van C-terminale partiële peptiden voorkomen dan wordt de offset frequentie functie ook berekend voor alle P_i^-. De intensiteit thresholds voor deze ionen kunnen op dezelfde manier bepaald worden als gebeurd is voor de N-terminale ionen.

3.4.3 Intensiteitsthresholds en C-terminale ionen

De literatuur over Sherenga vermeldt niets over de combinatie van C-terminale ionen en het gebruik van intensiteitsthresholds. We gaan er van uit dat er logischerwijs geen berekening wordt gemaakt voor een y-ion indien de piek in kwestie zich in een rang bevindt die wel met b- maar niet met y-ionen overeenstemt. In zo'n geval hebben we maar één i.p.v. twee knopen waardoor er geen verboden paar gevormd wordt. Complementaire ion types kunnen echter in dezelfde rangen voorkomen. Zo zullen zowel b- als y-ionen in hoge rangen voorkomen, en zijn verboden paren onvermijdelijk. Voor andere iontypes gebeurt dit op een analoge wijze.

3.5 Onnauwkeurigheden van meetresultaten en fragmentatie

We hadden onder meer aangenomen dat pieken die verschillende ion types van dezelfde partiële peptide representeren, in dezelfde knoop resulteren. Indien er d ion types ($d \leq k$) van een bepaalde partiële peptide aanwezig zijn in het spectrum krijgen we dus : $s_1 + \delta_1 = s_2 + \delta_2 = ... = s_d + \delta_d = m(P_i)$, wat er op neerkomt dat deze d knopen allen dezelfde waarde hebben, wat resulteert in één enkele knoop met waarde $m(P_i)$.
In werkelijkheid geldt dit echter niet, als gevolg van onnauwkeurigheden in de massametingen. Deze d pieken $s_1, ..., s_d$ zullen daardoor, na de optelling met hun δ, niet gelijk zijn aan elkaar. De verschillende pieken zullen dan ook leiden tot verschillende knopen $s_j + \delta_j, 1 \leq j \leq d$. Om deze knopen, die in wezen dezelfde partiële peptide voorstellen toch tot één enkele knoop samen te voegen wordt er een *merge algoritme* gebruikt.

3.5.1 Merge algoritme

Het merge algoritme beslist welke knopen in de spectrumgraaf gemerged kunnen worden tot één knoop. Dit moet op een zeer nauwkeurige manier gebeuren. Indien knopen die tot dezelfde partiële peptide behoren niet gemerged worden, dan lopen we de kans dat betekenisvolle spectrumpieken als foutieve knoop geïnterpreteerd worden waardoor ze niet in de spectrumgraaf opgenomen worden. Omgekeerd, indien er knopen bij elkaar gevoegd worden die niet tot dezelfde partiële peptide behoren, kan het zijn dat knopen die geen reële partiële peptide voorstellen geïnterpreteerd worden als zijnde betekenisvol.
Om de knopen te mergen wordt een *greedy* algoritme gebruikt; in iedere stap van dit algoritme wordt er gezocht naar de twee knopen u en v (resp. afkomstig van de pieken s en t) waarvan het verschil in waarde het kleinste is. Deze twee knopen worden gemerged tot één enkele knoop met een nieuwe waarde die het gewogen gemiddelde van de oorspronkelijke knopen bedraagt : $[i(s)u + i(t)v] / [i(s) + i(t)]$, waarbij *i(s)* de rang van piek s voorstelt. Dit algoritme wordt uitgevoerd totdat het verschil tussen iedere twee knopen minstens ϵ bedraagt, met ϵ een vooropgestelde nauwkeurigheid. Een mogelijke waarde voor ϵ is 0,5 [6].

Pseudo-code

Listing 3.3: Pseudo-code Sherenga : merge algoritme

```
do
  zoek de twee knopen u en v met het kleinste verschil;
  if (verschil < epsilon) do
    merge u en v;
    bereken waarde van nieuwe knoop
      waarde = [i(s)*u + i(t)*v]/[i(s) + i(t)];
until (verschil >= epsilon)
```

3.5.2 Bogen

Ook voor het verbinden van de knopen met bogen moet er rekening gehouden worden met de onnauwkeurigheden van de massametingen. Daarom wordt er niet gezocht naar exacte massa's van aminozuren, maar wordt er ook hier gebruik gemaakt van een vooropgestelde nauwkeurigheid ϵ, ook wel *error range* genoemd. Er wordt een boog getrokken van knoop u naar knoop v indien geldt : $-\epsilon \leq v - u - m(a) \leq \epsilon$, met $m(a)$ de massa van een aminozuur. Een mogelijke waarde voor deze ϵ is weer 0.5 [7].

Pseudo-code

$m(a)$ stelt de massa voor van eender welk aminozuur a.

Listing 3.4: Pseudo-code Sherenga : bogen bepalen

```
voor elk paar bogen do
  if (-epsilon <= v - u - m(a) <= epsilon) do
    trek een boog tussen de knopen v en u;
```

[6] Epsilon is een waarde die experimenteel bepaald wordt.
[7] Ook hier weer is ϵ experimenteel bepaald.

3.5.3 Gap en bridge edges

De fragmentatie van peptiden is een proces dat onderhevig is aan fouten die invloed hebben op de constructie van de spectrumgraaf. Een peptide kan een onvolledige fragmentatie ondergaan, waardoor niet alle mogelijke fragmentatie-ionen in het spectrum aanwezig zijn (of zo weinig voorkomen dat ze als ruis geïnterpreteerd worden). Dit leidt tot een niet compleet spectrum en kan leiden tot een gefragmenteerde spectrumgraaf, of tot een graaf die foutieve paden voorbrengt. Hiertoe wordt de spectrumgraaf aangepast door zogenaamde *gap edges* toe te voegen die di- en tripeptiden voorstellen. Zij overbruggen de kloven tussen de knopen waarbij het verschil in massa gelijk is aan twee, resp. drie aminozuren, oftewel di- resp. tripeptiden.

Door de toepassing van het merge algoritme op de spectrumgraaf kan het voorkomen dat, door de herberekening van de knoopmassa's, het verschil in massa tussen twee aangrenzende knopen groter wordt dan de massa van één aminozuur, rekening houdend met de toegelaten *error range* ϵ. Hierdoor zouden deze knopen door toedoen van het merge algoritme niet door een boog verbonden worden. Om dit te voorkomen worden er *bridge edges* ingevoerd. Om twee knopen u en v met een *bridge edge* te verbinden, moet er aan volgende voorwaarden voldaan zijn :

- er zijn pieken $s, t \in S$ en ion type $\delta \in \Delta$;
- hiervoor geldt :
  - $-\epsilon < |s - t| - m(a) < \epsilon$;
  - knoop $s + \delta$ is gemerged in knoop u;
  - knoop $t + \delta$ is gemerged in knoop v.

De *bridge edge* wordt net zoals gewone bogen gelabeld met het aminozuur a. Op die manier zullen knopen die vóór de toepassing van het merge algoritme verbonden zouden worden hun boog zeker niet verliezen.

Pseudo-code

Listing 3.5: Pseudo-code Sherenga : gap edges

```
voor elk paar knopen (u,v) dat nog niet verbonden is met een boog do
  if (-epsilon < |v-u| - m(dipeptide) < epsilon) do
    trek een boog tussen u en v;
    benoem de boog met de dipeptide;
  if (-epsilon < |v-u| - m(tripeptide) < epsilon) do
    trek een boog tussen u en v;
    benoem de boog met de tripeptide;
```

Listing 3.6: Pseudo-code Sherenga : bridge edges

```
voor elk paar knopen (u,v) dat nog niet verbonden is met een boog do
  if (-epsilon < |s-t| - m(a) < epsilon) and
     (knoop s+delta is gemerged tot knoop u) and
     (knoop t+delta is gemerged tot knoop v) do
    trek een boog tussen u en v;
    benoem de boog met aminozuur a;
```

3.6 Parent massa

Het is zeer belangrijk dat de meting van de parent massa/lading in de massaspectrometer zo nauwkeurig mogelijk gebeurt. Een error in deze meting leidt immers systematisch tot errors in de massa's van de knopen; we hebben in de sectie over C-terminale ionen gezien dat de parent massa gebruikt wordt om knopen te berekenen. In de praktijk is het verschil tussen de werkelijke parent massa en de experimenteel geobserveerde parent massa echter meestal zó groot, dat errors in de interpretatie van de sequentie onvermijdelijk zijn. Sherenga probeert dit op te lossen en heeft hiervoor een combinatorisch algoritme ontworpen om de parent massa te verbeteren. Dit algoritme tracht een nauwkeurigere bepaling te bekomen van de parent massa dan in het experiment is gebeurd. Hiertoe baseert Sherenga zich op de relatie die er bestaat tussen complementaire b- en y-ionen, en de gegeven parent massa.

3.6.1 Herberekening van de parent massa

We herhalen nog even de definities van partiële peptiden :

- N-terminale partiële peptide $P_i = p_1, ..., p_i$ met $i = 1, ..., n-1$;
- C-terminale partiële peptide $P_j^- = p_j, ...p_n$ met $j = 2, ..., n$.

Merk op dat de partiële peptiden P_i en P_{i+1}^- complementair zijn. Zij nu $S = \{s_1, ..., s_m\}$ het spectrum van een peptide P. De *reflectie* van S wordt gedefinieerd als een spectrum $\overline{S} = \{\overline{s}_1, ..., \overline{s}_m\}$ zo dat $\overline{s}_i = m(P) - s_i - d$, met $d = \delta_{b-ion} + \delta_{y-ion}$, de som van de offsets (δ's) van b- en y-ionen. Indien een spectrum S een piek s bevat die overeenkomt met een b-ion van een partiële peptide P_i, en een piek t die overeenkomt met een y-ion van de (complementaire) partiële peptide P_{i+1}^-. Dan geldt : $\overline{s} = t$. Een korte berekening toont dit aan :

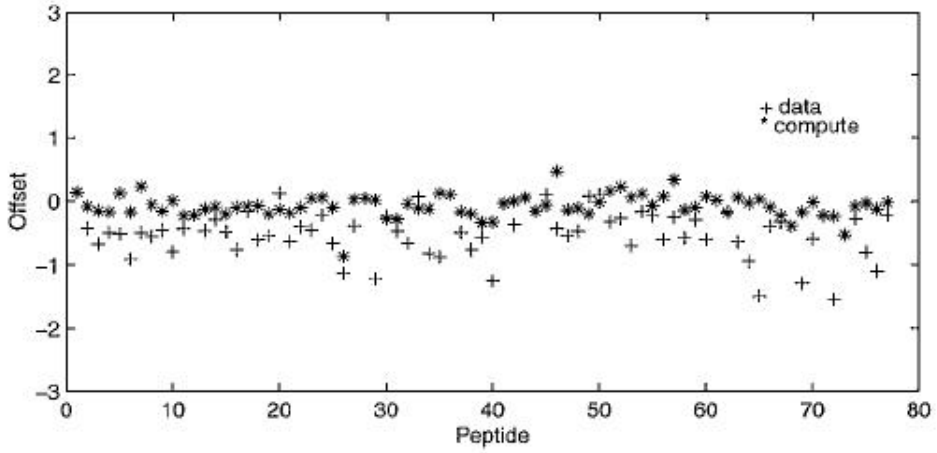
$$\begin{aligned}
\overline{s} &= m(P) - s - d \\
&= m(P) - (m(P_i) - \delta_b) - d \\
&= (m(P) - m(P_i)) + \delta_b - d \\
&= m(P_{i+1}) + \delta_b - (\delta_b + \delta_y) \\
&= m(P_{i+1}) - \delta_y \\
&= t
\end{aligned}$$

Met andere woorden, hebben we een piek $s \in S$ die een b-ion representeert, dan representeert de piek $\overline{s} \in \overline{S}$ niets anders dan het complementaire y-ion. In de berekening zijn we er van uitgegaan dat de parent massa juist is. In dat geval zou er dus een goede uitlijning moeten zijn tussen de pieken die overeenkomen met b-ionen in spectrum S, en die overeenkomen met y-ionen in spectrum $\overline{S}$. Waarom er gekozen werd voor b- en y-ionen is voor de hand liggend, deze komen het meeste voor waardoor het grootste deel van de pieken in een spectrum b- en y-ionen voorstellen. Hierdoor is er ook daadwerkelijk een goede uitlijning mogelijk. Op basis hiervan is het mogelijk om de parent massa te herberekenen.

3.6.2 Combinatorisch algoritme

Zij x de variabele die de berekende parent massa voorstelt. Voor een spectrum $S = \{s_1, ..., s_m\}$ definiëren we dan $\overline{S} = \{\overline{s}_1, ..., \overline{s}_m\}$ met $\overline{s}_i = x - s_i - d$. Het is mogelijk dat de spectra S en $\overline{S}$ gemeenschappelijke elementen hebben voor waarden van x die niet gelijk zijn aan (de theoretisch juiste) $m(P)$. Is echter x wel gelijk aan $m(P)$, dan zullen S en $\overline{S}$ veel meer gemeenschappelijke elementen hebben dankzij de uitlijning tussen b- en y-ionen.
Zij nu $c(S, \overline{S}(x))$ het aantal pieken, $s_i \in S$ en $\overline{s}_j \in \overline{S}$, zó dat $|s_i - \overline{s}_j| < \epsilon$, met ϵ een gegeven nauwkeurigheid. Dit is een meting van het aantal pieken dat goed uitgelijnd kan worden tussen

de twee spectra. Een goede keuze voor $m(P)$ zou dus de waarde voor x zijn die $c(S, \overline{S}(x))$ maximaliseert. In het geval dat er meerdere mogelijke waarden voor x zijn wordt díe x gekozen die de som van de afstanden $|s_i - \overline{s}_j|$ van de uitgelijnde pieken (dus met $|s_i - \overline{s}_j| < \epsilon$) minimaliseert. Figuur 3.7 geeft de offsets oftewel de afwijkingen ten opzichte van de theoretisch correcte parent massa weer van een tachtigtal parent peptiden waarop het algoritme is getest. De plus-tekens (+) geven de offsets weer op basis van de door de massaspectrometer gemeten parent massa. De ster-tekens (*) geven de offsets weer van diezelfde parent peptide na herberekening met bovenstaand algoritme. Op een paar uitzonderingen na hebben de herberekende parent massa's een veel kleinere offset. De juistheid van de parent massa wordt dus sterk verbeterd dankzij de toepassing van dit algoritme, waardoor er minder fouten zullen sluipen in de berekening van de knopen voor C-terminale ionen.



Figuur 3.7: Offsets voor (+) en na (*) de herberekening van de parent massa [20]

3.7 Scoring paths

Als er meerdere 'langste paden' gevonden zijn in de spectrumgraaf, dan moet er een manier zijn om uit te maken welk van deze paden, oftewel welk van deze *kandidaatpeptiden*, het experimentele spectrum het best 'verklaart'. Hiervoor wordt er een probabilistisch model opgesteld. Eerst wordt bepaald hoe er aan iedere knoop een *score* gegeven wordt. Vervolgens bekijken we hoe we een gevonden pad kunnen beoordelen aan de hand van deze scores.

3.7.1 Notaties en definities

Vooraleer we dieper ingaan op het probabilistisch model geven we eerst nog enkele notaties en definities die we zullen gebruiken.

- Zij P een peptide en S een spectrum. Dan definiëren we *p(P,S)* als de kans dat een peptide P het spectrum S heeft geproduceerd. We kunnen het *scoring paths* probleem nu als volgt formuleren :

  Vind een peptide P waarvoor p(P,S) maximaal is voor een gegeven spectrum S

  M.a.w., gegeven een aantal paden die gevonden zijn in de spectrumgraaf, vind hieruit het pad (dus de peptide) waarvoor *p(P,S)* maximaal is.

- We zijn ondertussen vertrouwd met het set ion types $\Delta = \{\delta_1, ..., \delta_k\}$ die bij een bepaalde massaspectrometer horen. Sommige van deze ion types zullen meer voorkomen dan andere en omgekeerd. Daarom noteren we de kans op een ion type δ_i als $p(\delta_i)$. Een δ_i-ion van een partiële peptide wordt, onafhankelijk van de probabiliteiten van de andere δ's, geproduceerd met kans $p(\delta_i)$.
- Tot slot moet ook de ruis die een massaspectrometer produceert vertegenwoordigd zijn. Op iedere positie in het spectrum kan de massaspectrometer een ruispiek genereren met kans q_R.

We kunnen nu zeggen dat, een piek die voorkomt op een plaats die overeenstemt met een δ_i-ion, gegenereert wordt met een kans q_i die gelijk is aan :

$$q_i = p(\delta_i) + (1 - p(\delta_i)) * q_R \tag{3.8}$$

In woorden : de kans dat er een piek voorkomt op een plaats in het spectrum die overeenkomt met een ion type is gelijk aan de kans dat dit ion type voorkomt in de massa spectrometer ($p(\delta_i)$) + de kans dat dit ion type niet voorkomt ($1 - p(\delta_i)$) vermenigvuldigd met de kans op ruis.
De waarde van $p(\delta_i)$ kan geschat worden van de geobserveerde empirische distributie van de *training set* (zie tabel 3.2).

Theoretisch gezien kan een partiële peptide tot k corresponderende pieken hebben in het spectrum; één piek voor ieder ion type. De kans dat voor een partiële peptide *al* deze k pieken (dus alle ion types uit de set Δ) ook effectief in het spectrum aanwezig zijn is gelijk aan :

$$\prod_{i=1}^{k} q_i \tag{3.9}$$

De kans dat een partiële peptide *geen enkele* piek in het spectrum heeft is gelijk aan :

$$\prod_{i=1}^{k} (1 - q_i) \tag{3.10}$$

3.7.2 Score voor de knopen

We hebben na het opstellen van de spectrumgraaf een gerichte, acyclische graaf bekomen. Enkel de knopen moeten nog een score krijgen. Het zoeken naar een langste pad in zo een graaf is één van de weinige graaf-problemen die in een lineaire tijd opgelost kunnen worden. We zullen echter zien (sectie 3.8) dat dit voor de spectrumgraaf niet zo vanzelfsprekend is.
Onderstel dat een spectrum de ionen $\delta_1, ..., \delta_l$ van een bepaalde partiële peptide P_i bevat ($l \leq k$), ook wel *present ions* genoemd, maar dat het spectrum *niet* de ionen $\delta_{l+1}, ..., \delta_k$ voor deze P_i bevat, ook wel *missing ions* genoemd. De l aanwezige ionen zullen resulteren in een knoop in de spectrumgraaf die overeenkomt met P_i. De vraag is nu welke score we aan deze knoop moeten geven.
Daarvoor maakt Sherenga gebruik van het *premium for present ions, penalty for missing ions* principe. De score van de knoop wordt dan gegeven door :

$$\frac{q_1}{q_R} ... \frac{q_l}{q_R} \frac{(1 - q_{l+1})}{(1 - q_R)} ... \frac{(1 - q_k)}{(1 - q_R)} \tag{3.11}$$

De noemers zijn toegevoegd om de waarden te normaliseren t.o.v. de aanwezige ruis. De gegevens uit de massaspectrometer zijn nu eenmaal onderhevig aan ruis. De waarden q_i/q_R kunnen hierdoor overgenomen worden uit de offset frequentie functie.
Op deze manier krijgen alle knopen van de graaf een score, die gebruikt worden tijdens het zoeken naar een pad in de graaf.

3.7.3 Het probabilistisch model

Het probabilistisch model definieert de kans *p(P,S)* dat een peptide P het spectrum S produceert. We gaan nu beschrijven hoe deze kans berekend wordt, en op welke manier we hieruit een score kunnen afleiden waardoor we de peptide kunnen vinden waarvoor *p(P,S)* maximaal is.
Voor de eenvoud wordt er verondersteld dat alle partiële peptiden gelijkwaardig zijn aan elkaar, en negeren we de intensiteiten van de pieken. We verdelen de ruimte van alle massa's (uit het spectrum S) in discrete intervallen van 0 tot $M = m(P)$ (massa van het parent ion).

Notaties :

- $T = \{0, ..., M\}$,
- we stellen het spectrum voor als een M-dimensionele vector $V = \{v_1, ..., v_M\}$,
- v_t is de indicator voor de aan- of afwezigheid van een piek op positie t, met $0 \leq t \leq M$. $v_t = 1$ als er een piek is op positie t, anders is $v_t = 0$,
- voor een gegeven peptide P en positie t is v_t een $0-1$ random variabele, wat wil zeggen dat v_t de waarden 0 (geen piek in het spectrum op positie t) of 1 (wel een piek op positie t) kan aannemen,
- de probabiliteitsdistributie van v_t wordt gegeven door $p(P, v_t)$ en geeft de kans aan dat er een piek in het spectrum van peptide P aanwezig is op positie t, m.a.w., de kans dat $v_t = 1$,
- zij $T_i = \{t_{i1}, ..., t_{il}\}$ $(l \leq k)$ de set posities die Δ-ionen van een partiële peptide P_i representeert (met $\Delta = \{\delta_1, ..., \delta_k\}$),
- $\cup_i T_i$ stelt dan de set posities voor die met een partiële peptide geassocieerd kunnen worden,
- zij $R = T \backslash \cup_i T_i$ de set posities die met *geen enkele* partiële peptide geassocieerd kan worden.

We kunnen de kansverdeling $p(P, v_t)$ nu als volgt definiëren :

De kans $p(P, v_t)$ voor een positie $t = t_{ij} \in T_i$ wordt gegeven door :

$$p(P, v_t) = \begin{cases} q_j & \text{if } v_t = 1 \ \ (\textit{er is een piek gegenereerd op positie } t) \\ 1 - q_j & \text{otherwise} \end{cases}$$

Analoog wordt $p(P, v_t)$ voor $t \in R$ gegeven door :

$$p_R(P, v_t) = \begin{cases} q_R & \text{if } v_t = 1 \ \ (\textit{er is een ruispiek op positie } t) \\ 1 - q_R & \text{otherwise} \end{cases}$$

De globale kans van pieken met ruis in het spectrum kan nu geschat worden als :

$$\prod_{t \in R} p_R(P, v_t) \tag{3.12}$$

De kans dat een partiële peptide P_i pieken in het spectrum voortbrengt op de posities van set T_i (alle andere posities worden genegeerd) wordt gegeven door :

$$p(P_i, S) = \prod_{t \in T_i} p(P, v_t) \tag{3.13}$$

Voor de eenvoud wordt er verondersteld dat iedere piek van het spectrum slechts tot één enkele set T_i behoort, wat impliceert dat het feit dat piek-waarden mogelijk verscheidene identificaties kunnen hebben genegeerd wordt.
Dan kunnen we de kans $p(P, S)$ dat een peptide P het spectrum S gegenereerd heeft als volgt definiëren :

$$p(P,S) = \prod_{t=1}^{M} p(P, v_t) = \left[\prod_{i=1}^{n} p(P_i, S)\right] \prod_{t \in R} p_R(P, v_t) \tag{3.14}$$

In woorden : de kans dat een peptide P een spectrum S gegenereerd heeft is gelijk aan het product over alle discrete posities in het spectrum van de kansen dat peptide P op de positie in kwestie een piek kan produceren. Ofwel : het product over alle partiële peptide P_i dat P_i pieken in het spectrum genereert op posities van de set T_i, vermenigvuldigd met de globale kans op ruispieken in het spectrum S.

We hebben gezien dat voor een gegeven spectrum S, de waarde van $\prod_{t \in T} p_R(P, v_t)$, afgekort tot $p_R(S)$, niet afhangt van P. De waarde van $p_R(P, v_t)$ hangt immers enkel af van het al dan niet aanwezig zijn van een ruispiek en de waarde van q_R.
De maximalisatie van $p(P, S)$ is daarom ook gelijk aan de maximalisatie van :

$$\frac{p(P,S)}{p_R(S)} = \frac{\prod_{i=1}^{n} \prod_{j=1}^{k} p(P, v_{t_{ij}}) \prod_{t \in R} p_R(P, v_t)}{\prod_{t \in T} p_R(P, v_t)} \tag{3.15}$$

$$= \prod_{i=1}^{n} \prod_{j=1}^{k} p(P, v_{t_{ij}}), \tag{3.16}$$

waardoor de rechterzijde van de vergelijking veel eenvoudiger wordt.

3.8 Paden in de graaf

Tot slot bespreken we nog op welke manier er naar paden wordt gezocht in de graaf. De spectrumgraaf is een gerichte acyclische graaf. Zodra de graaf opgesteld is kunnen we daarom het peptidesequentieprobleem als volgt uitdrukken :

Zoek het langste pad in een gerichte acyclische graaf

Het langste-pad-algoritme in een GAG is een zeer bekend probleem uit de theoretische informatica. Hiervan is bekend dat dit een snel, lineaire tijd, dynamisch algoritme is, wat een zeer groot voordeel is van de werkwijze met de spectrumgraaf [23]. Spijtig genoeg werkt dit algoritme niet in de praktijk, o.w.v. de *fake twin vertices*.

3.8.1 De *fake twin vertex*

In sectie 3.4.1 over C-terminale ionen hebben we het probleem met de *fake twin vertex* al even aangehaald. Doordat iedere piek in het spectrum S geïnterpreteerd wordt als een N-terminale ion maar ook als C-terminale ion heeft iedere 'echte' knoop (die overeenkomt met een massa m) een zogenaamde *fake twin vertex* met een massa $m(P) - m - offset$. Dit veroorzaakt het probleem dat, indien de echte knoop een hoge score behaalt, de valse knoop ook een hoge score zal behalen. Het langste-pad-algoritme zal hierdoor vaak zowel de echte als de valse knopen in het langste pad opnemen. Deze paden kunnen geen aannemelijke peptide sequenties representeren en kunnen dus best vermeden worden. We moeten het algoritme dus ietwat aanpassen.

3.8.2 Het antisymmetrisch langste-pad probleem

Zij G een graaf, en zij T een set *verboden paren* knopen, m.a.w. de *fake twin vertices*, uit G. Een pad in G heet *antisymmetrisch* indien het maximaal één knoop van ieder verboden paar bevat. Dit is net wat we willen bereiken. Het peptidesequentieprobleem klinkt nu als volgt :

Zoek het langste antisymmetrisch pad in G met een set verboden paren T

We hebben zonet een oplossing voor het *fake twin vertex* probleem gegeven, helaas stuiten we nu op een NP-hard probleem [24][25]. Tot op heden is er nog geen efficiënt algoritme gevonden om het antisymmetrisch langste pad probleem op te lossen. Toch is dit geen negatief resultaat. Dankzij de speciale structuur van de verboden paren is in dit specifieke geval wel een efficiënte oplossing mogelijk. Deze structuur is als volgt.
Twee verboden paren (x_1, y_1) en $(x_2, y_2), x_i < y_i$, worden *noninterleaving* genoemd indien de intervallen (x_1, y_1) en (x_2, y_2) *noninterleaving* zijn [20]. Dit wil zeggen dat het interval dat (x_1, y_1) beslaat, het interval (x_2, y_2) volledig omvat, of omgekeerd. Zouden we de knopen rankschikken volgens grootte dan moet dit er als volgt uitzien : $[x_1, x_2, y_2, y_1]$ of $[x_2, x_1, y_1, y_2]$. Omdat de verboden paren complementair zijn in massa (met betrekking tot de parent massa) geldt : $x_1 + y_1 = x_2 + y2$. Dankzij deze eigenschap zijn iedere twee paren complementaire knopen (oftewel verboden paren) *noninterleaving*.
We bewijzen dit eerst voor twee paren die voldoen aan bepaalde eigenschappen :

Bewijs :

- ***Gegeven*** :
  - twee complementaire paren (x_1, y_1) en (x_2, y_2),
  - $x_1 + y_1 = x_2 + y_2$.
- We gaan er van uit dat $x_1 < y_1$ en $x_2 < y_2$. Is dit niet zo, dan kunnen we de knopen hernoemen zodat aan deze eigenschap voldaan is.
- We geven dit bewijs voor het geval dat $x_1 < x_2$.
- ***Te bewijzen*** : de paren (x_1, y_1) en (x_2, y_2) zijn *noninterleaving*, ofwel : het is onmogelijk dat $x_2 < y_1 < y_2$ [26] indien $x_1 < x_2$.
- ***Bewijs*** : Onderstel dat $x_2 < y_1 < y_2$. Indien we de knopen rangschikken volgens grootte dan krijgen we : $[x_1, x_2, y_1, y_2]$. En ook : $[x_1, x_2, y_1, y_2, (x_2 + y_2) = m(P)]$. Vermits we verondersteld hebben dat $y_1 < y_2$ zou x_1 groter moeten zijn dan x_2 opdat $x_1 + y_1 = x_2 + y_2$, wat een **contradictie** oplevert met het *Gegeven*.

Voor het geval waarin $x_2 < x_1$ gaat het bewijs volledig annaloog met als *te bewijzen* : de paren (x_1, y_1) en (x_2, y_2) zijn *noninterleaving*, ofwel : het is onmogelijk dat $x_1 < y_2 < y_1$ [26] indien $x_2 < x_1$.
Iedere twee paren complementaire knopen van de spectrumgraaf zijn daarom *noninterleaving*.

Een graaf G met een set verboden paren T wordt geschikt genoemd indien elke twee verboden paren noninterleaving zijn [20].

Het antisymmetrisch langste pad probleem wordt nu teruggebracht tot het antisymmetrisch langste pad probleem in een *geschikte* graaf. In de literatuur over Sherenga [20] beweert men dat hiervoor een efficiënt algoritme bestaat. Het wordt echter niet door de auteurs beschreven. De algoritmen die anderen hiervoor beschreven hebben komen aan bod in het volgende hoofdstuk.

3.9 Pseudo-code

Listing 3.7: Pseudo-code van het algoritme Sherenga

```
bereken de set Delta d.m.v. de offset frequentie functie;
bereken de intensiteit thresholds;
herbereken de parent massa;
bereken de knopen;
voer het merge algoritme uit op de graaf;
trek de bogen;
trek gap edges;
trek bridge edges;

geef een score aan iedere knoop;
zoek de langste paden;
geef deze paden een score a.d.h.v. het probabilistisch model;
geef het pad met de hoogste score als output;
```

Hoofdstuk 4

Algoritmen voor het antisymmetrisch langste pad probleem in een geschikte graaf

In [27] wordt een algoritme besproken dat het langste pad zoekt doorheen een spectrumgraaf. Dit wordt ook wel het *optimale pad* of de *optimale oplossing* genoemd. Het is echter gebleken dat dit optimale pad niet altijd een aannemelijke identificatie is van de peptide in kwestie. Daarom zullen we ook een algoritme beschrijven dat net *niet* het optimale pad zoekt, maar wel paden die qua score dicht bij dit optimale pad liggen; de *suboptimale paden* of de *suboptimale oplossingen*.

4.1 Herdefinitie van de spectrumgraaf

De pad-algoritmen zijn niet door de auteurs van Sherenga geschreven waardoor de notaties en aannames lichtjes verschillen. We overlopen daarom eerst even voor de duidelijkheid alle gegevens waarmee de pad-algoritmen zullen werken. We bespreken ook de complexiteit van het opstellen van de spectrumgraaf.

4.1.1 De spectrumgraaf

Gegeven een spectrum S van een ongekende parent peptide P met parent massa $m(P)$, bestaande uit n fragmentatie-ionen[1]. Een spectrumgraaf $G = (V, E)$ bestaat uit een verzameling knopen V met $|V|$ knopen en een verzameling bogen E met $|E|$ bogen. Volledig analoog aan de graaf bij Sherenga, bevat G twee knopen $v_{initial}$ en v_{final} waarbij $v_{initial} = 0$ en $v_{final} = m(P)$.
Lichtjes anders dan bij Sherenga is dat deze spectrumgraaf enkel rekening houdt met b- en y-ionen, dit voor de eenvoud van het algoritme. De knopen voor de fragmentatie-ionen worden wel op dezelfde manier berekend. Iedere piek wordt omgezet in twee knopen waarbij de piek eerst als b-ion en vervolgens als y-ion beschouwd wordt. Ook hier zal telkens één van de twee knopen de *fake twin vertex* zijn.
Bogen worden op analoge manier als bij Sherenga bepaald, maar worden getrokken indien twee knopen verschillen in een waarde die overeenkomt met de massa van *een aantal* aminozuren. De beperking tot één of twee aminozuren zoals bij Sherenga valt hier dus weg. Indien er een boog bestaat tussen twee knopen v_i en v_j, dan is $E(v_i, v_j) = 1$.
We gaan er van uit dat $f()$ een gekende scorefunctie is voor de knopen en de bogen. Deze scorefunctie geeft hogere scores aan knopen die overeenkomen met pieken met een grote intensiteit, en aan bogen die gelabeld kunnen worden met één enkel aminozuur. Andere knopen en bogen krijgen

[1] Met n fragmentatie-ionen bedoelen we dat spectrum S pieken vertoont op n verschillende indexen op de massa-as (x-as).

kleinere scores. We gaan niet dieper in op de exacte berekening van $f()$. Wanneer we $f()$ aanhalen gaan we er dus van uit dat het resultaat gedefinieerd is.

4.1.2 Opstellen van de spectrumgraaf in polynomiale tijd

Rekening houdend met bovenstaande vereenvoudigingen kunnen we de spectrumgraaf opstellen in polynomiale tijd. Eerst stellen we een *mass array* A op [27] :

$$A[m] = \begin{cases} 1 & \text{if } (m = \text{de massa van één of meerdere aminozuren}) \\ 0 & \text{else} \end{cases} \tag{4.1}$$

Hierbij is $0 < m \leq h$ en h de grootste massa aanwezig in het spectrum.
Deze *mass array* kan opgesteld worden in $O(\frac{h}{\delta})$ tijd, met δ de nauwkeurigheid van de metingen [27]. Voor het bewijs hiervan verwijzen we naar [27].

Gegeven een spectrum S met n pieken, kan de spectrumgraaf opgesteld worden in $O(n^2)$ tijd :

Bewijs :

De knopen worden opgesteld in $O(2n + 2) = O(n)$ tijd : voor iedere piek worden er immers twee knopen aangemaakt, plus de begin- en de eindknoop.
Tussen iedere twee knopen v_i en v_j van G wordt een gerichte boog getrokken van v_i naar v_j als en slecht als aan de volgende twee voorwaarden voldaan is :

- $0 <$ knoopwaarde$(v_j) -$ knoopwaarde$(v_i) < h$, m.a.w., er kunnen enkel bogen getrokken worden indien het verschil tussen twee knopen positief is, aangezien we werken met massa's van aminozuren. Hierdoor is de graaf acyclisch.
- $A[\text{knoopwaarde}(v_j) - \text{knoopwaarde}(v_i)] = 1$, m.a.w., het verschil tussen de twee knopen is de massa van één of meerdere aminozuren.

Er zijn in totaal $|V| = (2n + 2)$ knopen. Aangezien er voor iedere knoop maximaal $(2n + 2 - 1)$ mogelijke *partners* bestaan zijn er nooit meer dan $(2n + 2) * (2n + 2 - 1)$ oftewel $(4n^2 + 6n + 2)$ mogelijke paren knopen. De bogen van de graaf kunnen dus in $O(n^2)$ tijd berekend worden.

Het opstellen van de graaf met al zijn knopen en bogen neemt $O(n) + O(n^2)$ tijd in beslag, wat overeenkomt met $O(n^2)$ en dus *polynomiale* tijd.

4.1.3 Complexiteit

Overzicht van de tijdscomplexiteit :

- berekening van de *mass array* $A[m]$: $O(\frac{h}{\delta})$
- opstellen van de spectrumgraaf :
  - berekening van de knopen : $O(n)$
  - berekening van de bogen : $O(n^2)$
  - totale complexiteit voor het opstellen van de spectrumgraaf : $O(n^2)$
- **Totale complexiteit :** $O(\frac{h}{\delta}) + O(n^2)$

4.1.4 Pseudo-code

Listing 4.1: Pseudo-code : opstellen van de spectrumgraaf

```
voor ieder fragmentatie-ion aanwezig in het spectrum do
  beschouw het ion als een b-ion;
  bereken de knoop;
  beschouw het ion als een y-ion;
  bereken de knoop;
stel de mass array A op;
voor ieder paar knopen in de graaf do
  if (A[verschil tussen de twee knoopwaarden]=1)
    trek een gerichte boog;
```

4.2 De optimale oplossing voor het ideale peptidesequentieprobleem

We zullen het algoritme dat de optimale oplossing berekent stap voor stap opbouwen, door eerst uit te gaan van een aantal vereenvoudigingen die we later zullen laten vallen [27].
We hernoemen vanaf nu de knopen van de graaf als volgt : $x_0, x_1, ..., x_n, y_n, ..., y_1, y_0$, met x_0 de beginknoop met waarde 0 en y_0 de eindknoop met waarde $m(P)$. Alle andere knopen hiertussen zijn van klein naar groot gerangschikt. Dankzij de *non-interleaving*-eigenschap (sectie 3.8.2) geldt dan ook dat alle paren knopen x_i en y_i, $1 \leq i \leq n$ overeen komen met de b-ion- en de y-ion-interpretatie[2] van eenzelfde piek. Een pad van x_0 naar y_0 zal dus eerst een aantal x-knopen doorlopen, en vervolgens een aantal y-knopen. Het pad zal daarom altijd precies één boog van een x-knoop naar een y-knoop bevatten.

4.2.1 Het *ideale* peptidesequentieprobleem

In het ideale peptidesequentieprobleem bestaat een massaspectrum enkel uit b- en y-ionen, komen alle pieken met dezelfde intensiteit voor, en is er geen ruis aanwezig. We definiëren dit als volgt :

Het ideale peptidesequentieprobleem is equivalent aan het probleem dat, gegeven matrix M_I, vraagt naar een gericht pad van x_0 naar y_0 dat voor iedere $0 < i \leq n$ *precies* één knoop van het knopenpaar (x_i, y_i) bevat [27].

4.2.2 De matrix M_i

De matrix M_I uit bovenstaande definitie wordt gedefinieerd als een twee-dimensionale matrix, opgebouwd uit de gegevens van de spectrumgraaf $G(V, E)$.
$M_I(i, j)$, $0 \leq i, j \leq k$:

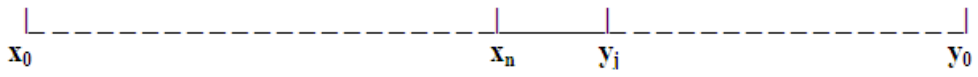
$$M_I(i,j) = \begin{cases} 1 & \text{als en slechts als aan de volgende voorwaarden voldaan is :} \\ & \text{1. er is een pad } L \text{ in } G \text{ van } x_0 \text{ naar } x_i, \text{ en} \\ & \text{2. er is een pad } R \text{ in } G \text{ van } y_j \text{ naar } y_0, \\ & \textit{zó dat } L \cup R \text{ precies één enkele knoop van } x_p \text{ of } y_p \text{ bevat, } p \in [1,i] \cup [1,j] \\ 0 & \textit{else} \end{cases} \tag{4.2}$$

De paden L en R worden de LR-paden voor $M_I(i, j) = 1$ genoemd.
De matrix M_I moet opgesteld worden vooraleer het algoritme op zoek kan gaan naar paden. Deze matrix kan opgesteld worden in $O(|V^2|)$ tijd. Hiervoor verwijzen we naar [27].

[2] Of omgekeerd.

4.2.3 De optimale oplossing voor G

We gaan er van uit dat de matrix M_I reeds berekend is, en weten dat dit $O(|V|^2)$ tijd in beslag neemt. Verder gaan we er van uit dat de optimale oplossing voor het *ideale* peptidesequentieprobleem een pad Q is.
We nemen aan dat deze oplossing Q de knoop x_n bevat. Vermits een mogelijke oplossing, volgens de definitie van het ideale peptidesequentieprobleem, voor iedere $0 < i \leq n$ ofwel de knoop x_i ofwel de knoop y_i bevat, is deze aanname geen verlies van algemeenheid. Het pad bevat daarom in het bijzonder ofwel de knoop x_n ofwel de knoop y_n. Het bewijs kan volledig analoog gevoerd worden indien we aannemen dat y_n deel uitmaakt van het pad. We gaan er van uit dat er een methode bestaat om uit te maken welke van deze twee paden het juiste is. Hier gaan we niet verder op in.
Nemen we aan dat het pad Q de knoop x_n bevat, dan moet Q een boog (x_n, y_j) bevatten, $j < n$, en is $M_I(n, j) = 1$. Om deze knoop y_j te vinden wordt de laatste rij uit matrix M_I doorzocht, de rij die overeenstemt met rij-index n. We gaan hierin op zoek naar de grootste j waarvoor $M_I(n, j) = 1$ en $E(x_n, y_j) = 1$, dus de grootste j waarvoor er twee paden $x_0, ..., x_n$ en $y_j, ..., y_0$ bestaan, die bovendien verbonden worden door de boog (x_n, y_j). Merk op dat deze extra vraag naar de *grootste* j geen extra kost met zich meebrengt, we kunnen de rij gewoon van achter naar voor doorlopen. Dit kost $O(|V|)$ tijd. Op dit moment hebben we dus een klein stukje van het pad Q reeds geïdentificeerd. We illustreren dit met een schematische voorstelling in figuur 4.1. De volle lijn stelt het geïdentificeerde deel van Q voor, de stippellijn het niet-geïdentificeerde deel.

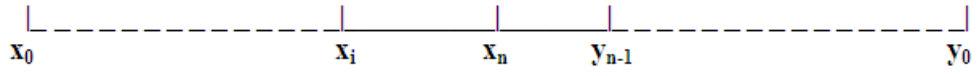


Figuur 4.1: Schematische voorstelling : $(x_n, y_j) \in Q$

We kunnen nu stap voor stap de overige knopen en bogen van Q identificeren. Aangezien $j < n$ kunnen we j opdelen in twee gevallen :

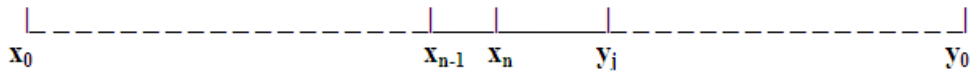
- $j = n-1$: we hebben dus zonet de boog (x_n, y_{n-1}) geïdentificeerd en zoeken vanaf $i = n-2$ tot $i = 0$ totdat we een knoop x_i gevonden hebben waarvoor geldt :
  - $E(x_i, x_n) = 1$ en
  - $M_I(i, j) = M_I(i, n-1) = 1$

  Het resultaat is te zien in figuur 4.2.



Figuur 4.2: Schematische voorstelling : $j = n - 1$

- $j < n-1$: dit impliceert dat de knoop x_{n-1} deel moet uitmaken van het pad Q. Wegens de definitie van het ideale peptidesequentieprobleem zou anders knoop y_{n-1} in Q moeten zitten, wat niet zo is aangezien $j < n - 1$.
  Dan is $E(x_{n-1}, x_n) = 1$ en ook $M_I(n-1, j) = 1$ (zie figuur 4.3).

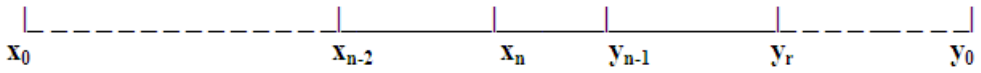


Figuur 4.3: Schematische voorstelling : $j < n - 1$

We hebben nu reeds drie (verbonden) knopen van pad Q geïdentificeerd. Om een volgende knoop te identificeren gaan we op precies dezelfde wijze te werk. In de vorige stap was $j = n - 1$ of $j < n - 1$. We bespreken het geval voor $j = n - 1$. Indien $j < n - 1$, gaat het algoritme volledig analoog aan voorgaande stap. Toen zijn we immers vertrokken met de kennis dat $j < n$.
Het geval $j = n - 1$ leidt tot het geïdentificeerde stuk pad aangegeven in figuur 4.2. Vermits y_{n-1} tot pad Q behoort moet $i < n - 1$, ofwel $i \leq n - 2$. We hebben weer twee mogelijke gevallen, volledig analoog aan de voorgaande stap :

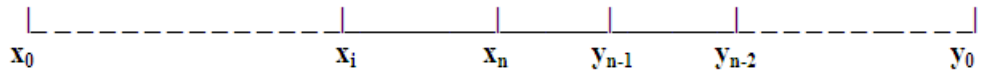
- $i = n - 2$: we zoeken vanaf $r = n - 3$ tot $r = 0$ totdat we een knoop y_r gevonden hebben waarvoor geldt
  - $E(y_{n-1}, y_r) = 1$ en
  - $M_I(i, r) = M_I(n - 2, r) = 1$

  Figuur 4.4 geeft het resultaat weer.



Figuur 4.4: Schematische voorstelling : $j = 1$ en $i = n - 2$

- $i < n - 2$: dit impliceert dat de knoop y_{n-2} deel moet uitmaken van het pad Q. Dan is $E(y_{n-1}, y_{n-2}) = 1$ en ook $M_I(i, n - 2) = 1$. Figuur 4.5 geeft het resultaat weer.



Figuur 4.5: Schematische voorstelling : $j = 1$ en $i < n - 2$

Eén van deze twee gevallen zal weer geldig zijn en we werken daar mee verder voor de volgende stap, enz. Op deze manier wordt het pad Q volledig geïdentificeerd. Op analoge wijze kunnen we ook het optimale pad berekenen indien y_n deel uitmaakt van Q. Omdat we iedere knoop maximaal één keer passeren, van x_n naar x_0 en van y_n naar y_0, kan dit algoritme werken in $O(|V|)$ tijd. We zijn er wel van uitgegaan dat de matrix M_I reeds berekend was en weten dat dit $O(|V^2|)$ tijd in beslag neemt. De totale kost van dit algoritme om een pad te vinden doorheen de graaf is daarom $O(|V|^2 + |V|)$.

4.2.4 Optimalisatie voor de matrix M_I

De matrix M_I kan geïmplementeerd worden als twee lineaire arrays in plaats van als een matrix. Hierdoor kan M_I geconstrueerd worden in $O(|V| + |E|)$ tijd. Voor het bewijs hiervan verwijzen we naar de literatuur [27].

4.2.5 Complexiteit

Overzicht van de tijdscomplexiteit voor het vinden van een optimaal pad in de spectrumgraaf voor het ideale peptidesequentieprobleem :

- opstellen van de matrix M_I : $O(|V| + |E|)$
- het vinden van het optimale pad : $O(|V|)$

Totale complexiteit : $O(|V| + |E|)$.

4.2.6 Pseudo-code

Listing 4.2: Pseudo-code : het optimaal algoritme voor het ideale peptidesequentieprobleem

```
construeer de matrix M_I;
zoek het optimale pad dat de knoop x_n bevat;
zoek het optimale pad dat de knoop y_n bevat;
geef het meest optimale pad van deze twee als output;
```

4.3 De optimale oplossing voor het peptidesequentieprobleem

Bij het *reële* peptidesequentieprobleem bevat een massaspectrum ruis en onnauwkeurige meetresultaten, en zijn de voorkomende iontypes niet beperkt tot b- en y-ionen. Het zonet besproken algoritme moet dan ook lichtjes gewijzigd worden wil men het uitvoeren op een reëel massaspectrum. Eerst wordt de spectrumgraaf opgesteld. We gaan er weer van uit dat de scorefunctie $f()$ gekend is.

4.3.1 De matrix M_R

Net zoals matrix M_I bij het ideale probleem gaan we ook nu een tweedimensionale matrix M_R opstellen. $M_r(i,j) > 1$ met $0 \leq i,j \leq n$, als en slechts als er in de graaf G een pad L bestaat van x_0 naar x_i, en een pad R van y_j naar y_0, zó dat voor iedere $p \in [1,i] \cup [1,j]$ *maximum één* van de twee knopen x_p en y_p deel uitmaakt van $L \cup R$. Anders is $M_R(i,j) = 0$.
De waarde van $M_R(i,j)$, indien $M_R(i,j) > 0$, bedraagt $\max_{L,R}\{s(L) + s(R)\}$, de maximum score die in de graaf aanwezig is van alle L-R paren.
Deze matrix M_R kan berekend worden in $O(|V||E|)$ tijd [27].

4.3.2 Een optimale oplossing voor G

Voor iedere i en j, m.a.w. voor ieder mogelijk paar knopen, wordt de scoresom

$$M_R(i,j) + f(x_i, y_j)$$

berekend, op voorwaarde dat voldaan is aan volgende twee voorwaarden :

- $M_R(i,j) > 0$, m.a.w. er bestaat een pad L van x_0 naar x_i en een pad R van y_j naar y_0, en
- $E(x_i, y_j) = 1$, m.a.w. er bestaat een boog van x_i naar y_j in graaf G.

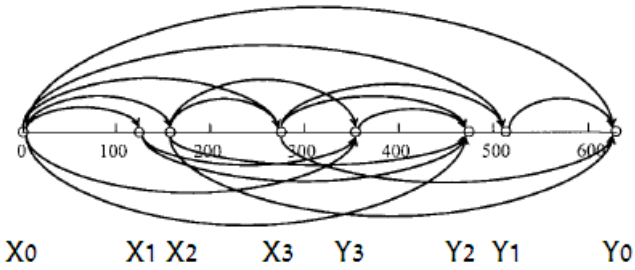
Merk op dat deze boog (x_i, y_j) de twee paden L en R met elkaar verbindt. De berekening van de som wordt dus enkel en alleen gemaakt indien het knopenpaar in kwestie een boog uit een mogelijke oplossing is.
Zijn nu $M_R(p,q) + f(x_p, y_q)$ de maximum waarde uit de berekende sommen. Door de constructie van M_R houdt dit ook in dat de boog (x_p, y_p) onderdeel is van het pad met de hoogste score. Om het hele pad te identificeren kunnen we, net zoals in het ideale peptidesequentieprobleem $M_R(p,q)$ *backtracken* om het hele pad te identificeren.
Omdat deze berekeningen kunnen gebeuren *tijdens* de berekening van de M_R-waarden moet de matrix niet opnieuw doorlopen worden. Het hele algoritme neemt daarom slechts $O(|V||E|)$ tijd in beslag [27].

4.3.3 Complexiteit

Overzicht van de tijdscomplexiteit voor het vinden van een optimaal pad in de spectrumgraaf :

- opstellen van de matrix M_R : $O(|V||E|)$
- het vinden van het optimale pad : geen extra tijd voor nodig

Totale complexiteit : $O(|V||E|)$.



Figuur 4.6: Voorbeeld-spectrumgraaf [29]

4.3.4 Pseudo-code

Listing 4.3: Pseudo-code : het optimale pad voor het peptidesequentieprobleem

```
bereken de matrix M_R;
bereken voor ieder paar knopen de scoresom;
identificeer het pad met de hoogste score;
```

4.4 Suboptimale oplossingen

Het is gebleken dat de optimale oplossing die het hierboven beschreven algoritme voortbrengt, niet altijd de juiste oplossing is. Door ruis en onnauwkeurigheden in massaspectra worden de parent peptiden vaak niet correct geïdentificeerd. Het zou daarom nuttig zijn om niet enkel naar de meest optimale oplossing te kijken, maar ook naar mogelijke identificaties die op het eerste zicht net iets minder optimaal lijken, maar die uiteindelijk wel de correcte oplossing bevatten. Een *suboptimaal algoritme* voor het *ideale peptidesequentieprobleem* werd door Lu en Chen [28] [29] beschreven. Dit algoritme zet de spectrumgraaf G om in een *matrix-spectrumgraaf* G_m, een matrix-representatie van de spectrumgraaf. De gedetailleerde uitleg hierover volgt in sectie 4.4.3.

4.4.1 Voorbeeld ter illustratie

Omdat het algoritme voor het vinden van suboptimale algoritmen zeer technisch overkomt zullen we het algoritme illustreren met een voorbeeld. Zij de graaf in figuur 4.6 de spectrumgraaf van een spectrum met pieken voor massawaarden $[157.10, 285.16, 357.22, 510.30]$ en een parent massa van 622.38 Da. Aan de hand van deze graaf zullen we de werking van het suboptimaal algoritme uitleggen.

4.4.2 Suboptimale oplossingen : de definitie

Onderstel dat P het langste pad is voor matrix-spectrumgraaf G_m, dus het pad met de hoogste score. Zij $f_{max} = f(P)$, met f de scorefunctie. Gegeven een ratio α, $0 < \alpha \leq 1$.

*Als een pad Q voldoet aan $f(Q) \geq \alpha * f_{max}$, dan is Q een α-**suboptimaal pad** voor G_m. Ofwel : Q is een **suboptimale oplossing*** [29].

Het suboptimale *de novo* peptidesequentieprobleem wordt nu gegeven door :

Gegeven een matrix-spectrumgraaf G_m, vind alle α-suboptimale paden [29].

4.4.3 De matrix-spectrumgraaf

De *matrix-spectrumgraaf* G_m is de matrix-voorstelling van de graaf[3]. We maken nog steeds gebruik van de reeks knopen van de graaf : $x_0, x_1, ..., x_n, y_n, ..., y_1, y_0$, met x_0 de beginknoop met waarde 0 en y_0 de eindknoop met waarde $m(P)$. Alle andere paren knopen x_i en y_i, $1 \leq i \leq n$ komen overeen met de b-ion en de y-ion interpretatie van eenzelfde piek. Zij $f()$ de scorefunctie en E de verzameling van bogen van de graaf G.

Knopen en bogen van de matrix-spectrumgraaf

Zij $X = \{x_0, x_1, ..., x_n\}$ en $Y = \{y_0, y_1, ..., y_n\}$, respectievelijk de verzameling x-knopen en de verzameling y-knopen. De matrix-spectrumgraaf wordt vervolgens gedefinieerd als $G_m = (V_m, E_m)$, met $V_m \subseteq X \times Y$ en $E_m \subseteq V_m \times V_m$:

- een *matrix-knoop* $\langle x_i, y_j \rangle \in V_m, i \neq j$, wordt gedefinieerd als v_{ij}. Voor iedere $i > 0$ geldt dat $v_{ii} \notin V_m$. In de matrix betekent dit dat deze elementen geen deel uitmaken van de verzameling matrix-knopen;
- er bestaan twee soorten *bogen* :
  1. $(v_{ij}, v_{im}) \in E_m$ als $m > i$ en $m > j$, en $(y_m, y_j) \in E$. De scorefunctie $f_m(v_{ij}, v_{im}) = f(y_m, y_j)$.
     Deze bogen representeren bogen tussen y-knopen en benoemen we ook met de term *horizontale bogen*. We kunnen (v_{ij}, v_{im}) uitschrijven als $(\langle x_i, y_j \rangle, \langle x_i, y_m \rangle)$. Zulk een boog representeert een boog van y_m naar y_j in de oorspronkelijke graaf. In de oorspronkelijke graaf lopen de bogen tussen y-knopen van een grotere naar een kleinere index. In de matrix-spectrumgraaf worden deze bogen gerepresenteerd door een *omgekeerde* boog, van de kleine naar de grote index. Hierdoor moet $m > j$.
     De andere beperking, $m > i$, komt zodadelijk aan bod.
  2. $(v_{ij}, v_{mj}) \in E_m$ als $m > i$ en $m > j$, en $(x_i, x_m) \in E$. De scorefunctie $f_m(v_{ij}, v_{mj}) = f(x_i, x_m)$.
     Deze bogen representeren bogen tussen x-knopen en benoemen we ook met de term *vertikale bogen*. We kunnen (v_{ij}, v_{mj}) uitschrijven als $(\langle x_i, y_j \rangle, \langle x_m, y_j \rangle)$. Zulk een boog representeert een boog van x_i naar x_m in de oorspronkelijke graaf. In de oorspronkelijke graaf lopen de bogen tussen x-knopen van een kleinere naar een grotere index. In de matrix-spectrumgraaf lopen de bogen op dezelfde manier. Hierdoor moet $m > i$.
     De tweede beperking, $m > j$, zorgt er samen met de beperking uit het vorige punt $m > i$ voor dat éénzelfde pad niet meerdere keren gevonden wordt. Dit zal geïllustreerd worden aan de hand van het voorbeeld.
- G_m heeft $|V_m| = O(n^2)$ knopen;
- G_m heeft $|E_m| = O(n^3)$ bogen. Uit iedere knoop in de matrix-spectrumgraaf kunnen maximaal $O(n)$ bogen vertrekken. Voor de $O(n^2)$ knopen geeft dit : $O(n^2 * n) = O(n^3)$.

Merk op dat de bogen enkel getrokken worden tussen knopen uit ofwel de X-reeks ofwel de Y-reeks. Daarenboven worden de bogen binnen de Y-reeks *omgekeerd* getrokken in de matrix-spectrumgraaf. Dit zal verduidelijkt worden aan de hand van het voorbeeld.

Terminale knopen

Een knoop v_{ij} die voldoet aan volgende voorwaarden wordt een *terminale knoop* genoemd :

- $(x_i, y_j) \in E$, in woorden : er is in de originele graaf een boog van knoop x_i naar y_i,
- v_{ij} heeft geen uitgaande bogen, en

[3] Merk op dat de matrix-spectrumgraaf niet hetzelfde is als de matrices die geconstrueerd werden voor het zoeken naar de optimale oplossing in secties 4.2 en 4.3.

- $i, j > 0$.

De verzameling terminale knopen wordt T benoemd. De terminale knopen representeren de bogen van een x-knoop naar een y-knoop uit de originele graaf die aan de twee laatste voorwaarden voldoen.

Paden

Een mogelijk pad in de matrix-spectrumgraaf is een pad dat begint in knoop v_{00} en eindigt in een terminale knoop.

4.4.4 Voorbeeld

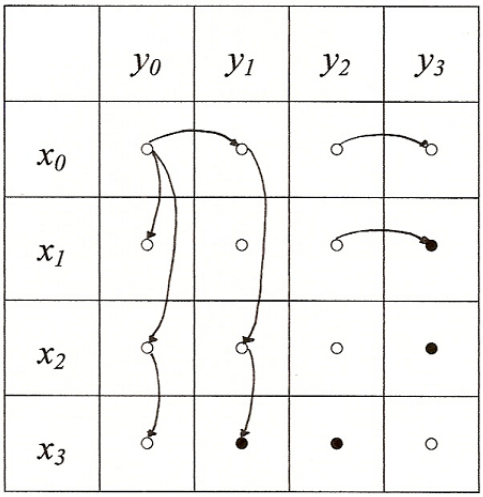
We gaan nu de matrix-spectrumgraaf opstellen voor de graaf uit ons voorbeeld. We hernemen hierbij stapsgewijs dezelfde indeling als in de vorige sectie, nu toegepast op het voorbeeld.

Knopen en bogen van de matrix-spectrumgraaf

Vooreerst hebben we de knopen $x_0, x_1, x_2, x_3, y_3, y_2, y_1, y_0$ van de originele graaf. Deze zijn ook zo benoemd op figuur 4.6. Volgende opsomming geeft een overzicht van de bogen uit de originele graaf en hun bijhorende matrix-bogen indien van toepassing :

- bogen tussen x-knopen :
  - $(x_0, x_1) \Rightarrow (v_{0j}, v_{1j})$, $j < 1$
  - $(x_0, x_2) \Rightarrow (v_{0j}, v_{2j})$, $j < 2$
  - $(x_0, x_3) \Rightarrow (v_{0j}, v_{3j})$, $j < 3$
  - $(x_2, x_3) \Rightarrow (v_{2j}, v_{3j})$, $j < 3$
- bogen tussen y-knopen :
  - $(y_3, y_2) \Rightarrow (v_{i2}, v_{i3})$, $i < 3$
  - $(y_1, y_0) \Rightarrow (v_{i0}, v_{i1})$, $i < 1$
- bogen van een x-knoop naar een y-knoop (kandidaat-terminale knopen), rekening houdend met de beperking die index 0 uitsluit :
  - $(x_1, y_3) \Rightarrow v_{13}$
  - $(x_1, y_2) \Rightarrow v_{12}$
  - $(x_2, y_3) \Rightarrow v_{23}$
  - $(x_3, y_2) \Rightarrow v_{32}$
  - $(x_3, y_1) \Rightarrow v_{31}$

De matrix-spectrumgraaf die hieruit opgesteld is, is te zien in figuur 4.7. We houden hierbij in het achterhoofd dat knopen v_{ii}, met $i > 0$, eigenlijk *geen* deel uitmaken van de matrix-spectrumgraaf. Bogen van of naar zulk een knoop behoren dan ook niet tot de verzameling matrix-bogen.
De gewone matrix-knopen worden weergegeven door een open bolletje, de terminale knopen door een gevuld bolletje. We zien dat van de vijf kandidaat-terminale knopen er nog vier overblijven die aan de voorwaarden voldoen.
De bogen tussen de x-knopen zijn *vertikale* bogen. Bogen tussen y-knopen zijn *horizontale* bogen die omgekeerde zijn aan de bogen in de originele graaf. De boog van v_{21} naar v_{23} bijvoorbeeld representeert de boog (y_3, y_1) uit de originele graaf. Door het omkeren van de y-bogen lopen alle bogen in de matrix-spectrumgraaf van links naar rechts of van boven naar onder.
Nog één laatste detail is het ontbreken van de boog (v_{00}, v_{30}). De boog (x_0, x_3) is immers in de graaf aanwezig. Omdat x_3 echter vanuit x_0 bereikbaar is via de knoop x_2, wordt de boog (x_0, x_3) niet weergegeven in de matrix-spectrumgraaf. Het passeren van de knoop x_2 impliceert immers dat er meer data uit het spectrum gebruikt wordt om uiteindelijk toch in dezelfde knoop x_3 te belanden.



Figuur 4.7: Matrix-spectrumgraaf van de graaf uit het voorbeeld [29]

Terminale knopen

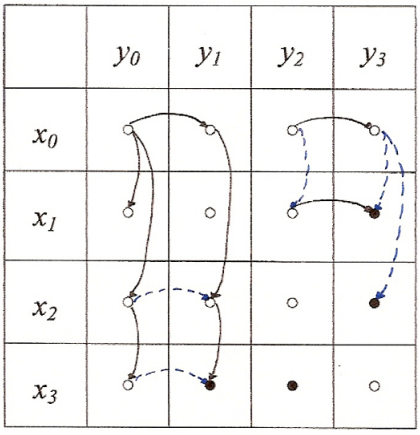
Alle bogen tussen een x-knoop en een y-knoop zijn kandidaat-terminale knopen. De echte terminale knopen zijn die knopen die aan de twee voorwaarden uit de definitie voldoen :

- v_{ij} heeft geen uitgaande bogen, en
- $i, j > 0$.

De bedoeling van de eerste voorwaarde is het zo lang mogelijk maken van de paden, in aantal gepasseerde knopen. Onderstel even dat er in het voorbeeld een boog zou zijn van x_2 naar y_1. Onderstel ook dat de voorwaarde er niet is, dan zou v_{21} een terminale knoop zijn. We vinden dan het pad v_{00}, v_{01}, v_{21} dat in de originele graaf overeenkomt met het pad x_0, x_2, y_1, y_0. Vervolgens vinden we ook een pad $v_{00}, v_{01}, v_{21}, v_{31}$ dat overeenkomst met x_0, x_2, x_3, y_1, y_0. Behouden we de voorwaarde, dan zullen we enkel het laatste pad vinden. En dat volstaat ook, want het laatste pad is, op één enkele knoop na, identiek aan het eerste. Het laatste pad passeert een extra knoop en gebruikt op die manier meer informatie uit het originele spectrum. Daardoor is het een betere oplossing dan het eerste pad.
De tweede voorwaarde houdt in dat de indices van de matrix-knopen groter moeten zijn nul. Vermits een pad vertrekt in v_{00} en eindigt in een terminale knoop moeten de x-rij en de y-kolom met index nul *verlaten* worden. Er moet dus minstens één boog naar rechts een één boog naar onder gevolgd worden. De reden om deze beperking op te leggen is de volgende. In het voorbeeld zien we dat er een boog (x_3, y_0) bestaat. Zonder deze beperkende voorwaarde zou matrix-knoop v_{30} ook een terminale knoop zijn. Dan kunnen we een *vertikaal pad* terugvinden in de matrix-spectrumgraaf : v_{00}, v_{20}, v_{30}. In de originele graaf is dit het pad x_0, x_2, x_3, y_0. We zien dat we in het eerste gedeelte van de peptide knopen hebben op x_2 en x_3, die aangeven dat er voor die massawaarden pieken in het spectrum aanwezig zijn, en dus fragmentatie-ionen. Daarna echter *springt* het pad als het ware naar y_0 zonder ook maar enig bewijs te leveren van fragmentatie-ionen in de regio tussen x_3 en y_0. Door de voorwaarde te hanteren wordt zulk een pad niet als juist aanvaard en zullen de gevonden paden allemaal minstens het bewijs leveren van één fragmentatie-ion zowel in de x-reeks als één in de y-reeks.

Paden

Een mogelijk pad in de matrix-spectrumgraaf is ook een mogelijk pad in de originele graaf. Zulk een pad begint met de knoop v_{00} en eindigt in een terminale knoop, zeg v_{kl}. Dit houdt in dat er in de originele graaf een boog is tussen de knopen x_k en y_l. Om in de terminale knoop te geraken



Figuur 4.8: Matrix-spectrumgraaf van de graaf uit het voorbeeld zonder de extra beperkende voorwaarde op de matrix-bogen [29]

heeft het pad minstens één horizontale en minstens één vertikale boog doorlopen, in willekeurige volgorde. De bogen die het pad vormen zijn niets meer dan de knopen uit de originele graaf die in dit pad voorkomen. De horizontale bogen vormen het omgekeerde y-pad van y_0 naar y_l. De vertikale bogen vormen het x-pad van x_0 naar x_k. De terminale knoop representeert de boog (x_k, y_l) en vormt zo de verbinding tussen beide deelpaden.

We illustreren hier ook de beperkingen uit de definitie van de bogen van de matrix-spectrumgraaf (sectie 4.4.3). Figuur 4.8 geeft de matrix-spectrumgraaf weer indien we deze beperkingen zouden laten vallen. De bijkomende bogen zijn in blauwe stippellijn weergegeven. We zien op de figuur dat het pad x_0, x_2, x_3, y_1, y_0 op drie verschillende manieren kan gevonden worden :

$$[v_{00}, v_{01}, v_{21}, v_{31}]$$
$$[v_{00}, v_{20}, v_{21}, v_{31}]$$
$$[v_{00}, v_{20}, v_{30}, v_{31}]$$

Om dit te vermijden werd de extra beperkende voorwaarde ingevoerd zodat éénzelfde pad slechts eenmaal gevonden wordt.

Fake twin vertex

Door de specifieke constructie van de matrix-xpectrumgraaf[4] is het onmogelijk geworden om een pad te vinden doorheen de matrix-spectrumgraaf dat, omgezet naar zijn werkelijke pad in de spectrum graaf, zowel de knoop x_i als de knoop y_i bevat. We hebben het probleem nu dus herleid naar een *normale* graaf en kunnen hiermee verder werken om op zoek te gaan naar paden.

4.4.5 Het langste pad P

Om het langste pad in deze gerichte acyclische graaf te berekenen maken we gebruik van het gekende *DFS-algoritme*[5]. Dit langste pad P kan, na constructie van de matrix-spectrumgraaf, gevonden worden in $O(|V| + |E|)$ tijd [29] [30].
Merk op dat het eerder besproken algoritme dat op zoek gaat naar een optimale oplossing voor het ideale peptidesequentieprobleem ook het langste pad vindt in $O(|V| + |E|)$ tijd, rekening houdend met de matrix M_I die hiervoor moet opgesteld worden. Omdat het suboptimale algoritme geen gebruik zal maken van de matrix M_I maar wel van de matrix-spectrumgraaf, hoeven we de kost

[4] De beperking dat $v_{ii} \notin V_m, \forall i > 0$, en de beperkingen op de bogen.
[5] Dept-first-search algoritme.

voor het opstellen van de matrix-spectrumgraaf niet expliciet in rekening te brengen voor het bepalen van het langste pad. De matrix-spectrumgraaf zal nodig zijn om de suboptimale paden te vinden en moet dus zowieso opgesteld worden. Beide algoritmen zorgen dus voor een extra kost van $O(|V| + |E|)$ voor het vinden van P.

4.4.6 Constructie van $l()$ en $r()$

Definitie

- zij $l(v_{ij})$ de maximum padscore van alle paden van v_{00} naar v_{ij};
- zij $r(v_{ij})$ de maximum padscore van alle paden van v_{ij} naar de terminale knopen;
- indien er geen pad bestaat dan worden de scores van l en r ingesteld op $-\infty$;
- zij O_{ij} de verzameling uitgaande bogen van v_{ij} [29].

Constructie

Om $l()$ en $r()$ te construeren worden eerst alle knopen topologisch geordend. Een topologische ordening van een gerichte acyclische graaf is een lineaire ordening van alle knopen op een welbepaalde manier. Zo moet, indien de graaf een boog (u, v) bevat, de knoop u vóór de knoop v staan in de geordende reeks. Als we de knopen op een horizontale as zouden rangschikken, dan moeten alle bogen van links naar rechts lopen. Het topologisch ordenen van een graaf kost $O(|V_m| + |E_m|)$ tijd [30].
Dit doen we nu met de matrix-spectrumgraaf. Vermits er uit v_{00} enkel matrix-bogen vertrekken en niet toekomen is v_{00} de eerste matrix-knoop in rij.
Zij u en w voorstellingen van matrix-knopen. Voor iedere boog $(v_{00}, u) \in E_m$ berekenen we $l(u) = f_m(v_{00}, u)$. Voor iedere matrix-knoop die met één boog bereikbaar is vanuit v_{00} krijgt $l()$ dus de waarde van de scorefunctie f_m voor deze boog.
Daarna wordt voor iedere matrix-knoop $u \in V_m$, ook weer in de topologische volgorde, de waarde van $l(w)$ aangepast voor alle matrix-knopenparen $(u, w) \in E_m$. De nieuwe waarde voor $l(w)$ is gelijk aan $max\{l(w),\ l(u) + f_m(u, w)\}$. Dus voor iedere matrix-knoop u die met één boog bereikbaar is vanuit v_{00}, wordt de verzameling O_u doorlopen, met O_u de verzameling van de uitgaande bogen van matrix-knoop u. Een boog van u naar w impliceert dat matrix-knoop w ook bereikbaar is vanuit v_{00}, zij het via de matrix-knoop u. Dit pad heeft de score $l(u)+f_m(u, w)$. De mogelijkheid blijft ook bestaan dat matrix-knoop w rechtstreeks bereikbaar is vanuit v_{00}. Dit pad heeft dan de waarde $l(w)$. De waarde van $l(w)$ wordt dan ingesteld op de hoogste padscore van deze twee mogelijke paden.
Door het doorlopen van de matrix-knopen in de welbepaalde volgorde worden paden als het waren opgebouwd vanuit v_{00}, en worden ook alle paden vanuit v_{00} gevonden. Het pad met de hoogste $l()$-score heeft daarom ook een score die gelijk is aan f_{max}, de maximum padscore voor G_m.
Het berekenen van de $r()$-waarden verloopt analoog, maar hierbij worden de scores berekend door de matrix-knopen in de omgekeerde topologische volgorde te doorlopen.
De constructie van $l()$ en $r()$ gebeurt op een gelijkaardige manier als het vinden van *single-source shortest paths*[6] in een gerichte acyclische graaf. Matrix-knoop v_{00} is hierbij de *source*-knoop voor het berekenen van $l()$, en voor het berekenen van $r()$ zijn de terminale matrix-knopen de *source*-knopen. Voorts wordt niet het kortste pad maar wel het langste pad bewaard. De constructie van $l()$ en $r()$ neemt, dankzij het gebruik van dit algoritme, $O(n^3)$ tijd in beslag [29] [30].

4.4.7 Het suboptimale algoritme

We herhalen even dat een suboptimaal pad Q voor de matrix-spectrumgraaf moet voldoen aan $f_m(Q) \geq \alpha * f_m^{max}$, voor een zekere α, $0 < \alpha \leq 1$. Merk op dat $f_m(Q)$ niets anders is dan $l(u)$,

[6] Het *single-source shortest path* algoritme gaat in een gerichte acyclische graaf op zoek naar het kortste pad tussen één bepaalde *source*-knoop en alle andere knopen.

met u de laatste knoop uit het pad Q. Wanneer we dus de uitdrukking $f_m(Q)$ in de tekst zullen gebruiken weet de lezer dat de waarde hiervan reeds berekend is bij de constructie van $l()$.
De matrix-knopen zijn nog steeds topologisch geordend. We doorlopen de geordende knopenlijst met het *DFS-**algoritme*** [30], en trachten dus steeds eerst het huidige pad te verlengen, eer we op zoek gaan naar andere paden.
Onderstel dat de huidige sequentie Q gelijk is aan $v_{00}, ..., u$, en dat de score van dit pad gelijk is aan $f(Q)$. Matrix-knoop u is de laatst bijgevoegde knoop in de iteratie van ons algoritme. Nu stelt $f_m(Q) + r(u)$ de score voor van een pad doorheen de matrix-spectrumgraaf van beginknoop v_{00}, via een matrix-knoop u, naar een terminale knoop. Dit pad bestaat uit een gekend *prefix-pad* Q en een (nog) ongekend *suffix-pad*, en is dus een mogelijke oplossing. Er zijn dan twee mogelijke operaties die uitgevoerd kunnen worden om het ongekende *suffix-pad* te identificeren :

- ***Backtracking*** : indien $f_m(Q) + r(u) < \alpha * f_m^{max}$
  Er is niet voldaan aan de voorwaarde voor een suboptimale oplossing. De matrix-knoop u kan dus geen deel uitmaken van een mogelijke suboptimale oplossing voor de matrix-spectrumgraaf en wordt daarom uit het pad Q verwijderd. Vermits $r(u)$ de maximale score is van alle paden van u naar een terminale knoop, zal geen enkel pad dat Q als *prefix-pad* heeft een waarde hebben die groter is dan $f_m(Q) + r(u)$.
  Stel dat de matrix-knoop s de voorlaatste knoop is uit het pad Q, dan geldt uiteraard dat $(s, u) \in E_m$. Dan wordt de eerstvolgende matrix-knoop t uit de topologisch gerangschikte reeks aan Q toegevoegd op voorwaarde dat $(s, t) \in E_m$.
  Het algoritme itereert verder met sequentie $Q = v_{00}, ..., s, t$.

- ***Exploring*** : indien $f_m(Q) + r(u) \geq \alpha * f_m^{max}$
  Er is aan de voorwaarde voldaan voor een suboptimale oplossing. Er bestaat dus een oplossing met het pad Q als *prefix-pad*. We moeten niet backtracken en voegen de eerstvolgende matrix-knoop w uit de geordende knopenlijst toe aan Q waarvoor geldt dat $(u, w) \in E_m$. We zijn dus zeker dat er een suboptimale oplossing bestaat die als *prefix-pad* het pad Q bevat. We zoeken dus verder naar nieuwe matrix-knopen die het pad verlengen om de identificatie te bekomen van de hele suboptimale oplossing.
  Het algoritme itereert verder met sequentie $Q = v_{00}, ..., u, w$.

In iedere iteratie wordt ook gecontroleerd of voldaan is aan volgende twee voorwaarden :

- $f_m(Q) \geq \alpha * f_m^{max}$, en
- u is een terminale knoop.

In dat geval wordt het pad Q bewaard als zijnde een suboptimale oplossing. Het DFS-algoritme werkt daarna gewoon verder en blijft itereren totdat alle mogelijke suboptimale paden gevonden zijn. In dat geval heeft ook de beginknoop v_{00} de operatie *backtracking* ondergaan, waardoor sequentie Q leeg is.

Complexiteit

Iedere boog ondergaat in het DFS-algoritme maximaal één keer de operatie *backtracking* en één keer de operatie *exploring*. De kost voor het vinden van alle suboptimale paden is dan ook $O(|E_m|)$ tijd.

4.4.8 Rangschikking van de suboptimale oplossingen

Alle gevonden suboptimale oplossingen worden nu gerangschikt om uit te kunnen maken welke kandidaatoplossing het meeste aansluit met het experimentele spectrum :

- voor iedere kandidaatoplossing worden de mogelijke peptidesequenties opgesteld. Deze zijn af te leiden uit de sequentie van knoopwaarden die telkens verschillen in de massa van een aminozuur;

- voor iedere mogelijke peptidesequentie wordt een *hypothetisch spectrum* opgesteld. Het verschil met een *theoretisch spectrum* is het toekennen van intensiteiten. De aanwezige ionmassa's krijgen een bepaalde intensiteit toegewezen afhankelijk van het iontype dat ze representeren[7];
- ieder hypothetisch spectrum wordt vergeleken met het experimenteel spectrum. Zij S_1 de som van alle intensiteiten die in het hypothetisch spectrum voorkomen. Zijn S_2 de som van alle intensiteiten in het hypothetisch spectrum waarvan de ionmassa overeenkomt met aanwezige ionmassa's in het experimenteel spectrum. De score S_2/S_1 geeft dan weer in welke mate het hypothetisch spectrum overeenkomt met het experimenteel spectrum;
- alle kandidaatpeptiden worden gerangschikt volgens de score die ze gekregen hebben.

4.4.9 Complexiteit

Overzicht van de tijdscomplexiteit voor het vinden van de suboptimale paden in de spectrumgraaf :

- het langste pad P zoeken : $O(|V| + |E|) = O(n + n^2) = O(n^2)$
- opstellen van de matrix-spectrumgraaf : $O(n^3)$
- topologisch ordenen van de knopen : $O(|V_m| + |E_m|) = O(n^2 + n^3) = O(n^3)$
- constructie van $l()$ en $r()$: $O(n^3)$
- suboptimale paden vinden : $O(|E_m|) = O(n^3)$

Totale complexiteit : $O(n^3)$.

4.4.10 Pseudo-code

Listing 4.4: Pseudo-code : suboptimale paden in een graaf

```
stel de spectrumgraaf op;
zoek het langste pad P en bewaar de score;
stel de matrix-spectrumgraaf op;
orden de knopen in topologische volgorde;
bereken l();
bereken r();
doorloop de knopenlijst met het DFS-algoritme
  if (suboptimaal-pad-conditie voldaan)
    explore;
  else
    backtrack;
  if (pad gevonden)
    geef pad als output;
voor iedere gevonden suboptimale oplossing do
  stel alle mogelijke peptidesequenties op voor dit pad;
  voor iedere mogelijke sequentie do
    stel een hypothetisch spectrum op;
    vergelijk het hypothetisch spectrum met het experimenteel spectrum;
    bereken de score;
rangschik de peptidesequenties volgens de score;
```

[7] De gebruikte intensiteiten in dit algoritme werden bepaald door de auteurs zelf [29].

Hoofdstuk 5

Het Hidden Markov Model : NovoHMM

NovoHMM [19] maakt voor de identificatie gebruik van het *Hidden Markov Model* [31]. Het Hidden Markov Model is een statistisch model dat veel gebruikt wordt in de computationele biologie. Om deze structuur aan te wenden voor het peptidesequentieprobleem wordt er gebruik gemaakt van het feit dat een peptide een sequentie is van aminozuren. We weten dat er 20 aminozuren voorkomen in de natuur die de bouwstenen vormen van proteïnen. We kunnen zo'n peptide dus schrijven als een sequentie van symbolen van een vast alfabet Σ van 20 aminozuren.
Vooraleer we tot de beschrijving van het eigenlijke algoritme overgaan bespreken we in een aantal stappen hoe men geëvolueerd is van een algemeen HMM naar het specifiek HMM dat gebruikt kan worden voor de identificatie van peptiden.

5.1 Algemeen model

Het HMM kan gezien worden als een abstracte machine die in staat is een bepaalde output te produceren. Deze output bestaat uit een opeenvolging van symbolen van een bepaald alfabet Σ, en heeft ook een bepaalde probabiliteitswaarde. Het genereren van de output gebeurt in een aantal discrete stappen, waarbij de machine zich aan het begin van iedere stap in een bepaalde *staat* bevindt. Er zijn een vast aantal, zeg n, staten. Gedurende iedere stap maakt het HMM twee beslissingen :

1. Naar welke staat ga ik verplaatsen?
2. Welk symbool uit het alfabet Σ geef ik hierbij als output?

Als antwoord op deze twee vragen kiest het HMM zowel tussen de n staten als tussen de symbolen van Σ. Deze keuzes worden bepaald aan de hand van probabiliteitsdistributies die aangeven naar welke staat er vanuit een bepaalde staat moet bewogen worden, en welk symbool hierbij als output moet gegeven worden. Deze distributies zijn verschillend van staat tot staat; als er n staten zijn, dan zijn er n verschillende *volgende-staat*-distributies en n verschillende *symbool-output*-distributies. Deze worden respectievelijk *transitieprobabiliteiten* en *outputprobabiliteiten* genoemd.
Een HMM wordt eerst opgesteld uit gekende gegevens. Onderstel dat we zo een HMM hebben. Zij een geobserveerde sequentie een sequentie bestaande uit symbolen van het alfabet Σ, geobserveerd door de gebruiker. Het *staten-pad* dat deze sequentie zou kunnen genereren is *verborgen* voor de gebruiker, vandaar de naam *Hidden* MM. Gegeven zo'n geobserveerde sequentie, is het nu de bedoeling om de meest waarschijnlijke statensequentie te bepalen die het HMM zou kunnen doorlopen hebben om deze sequentie als output te genereren. We gaan m.a.w. op zoek naar het verborgen staten-pad voor de geobserveerde sequentie, en wel dat met de hoogste probabiliteit. Dit staten-pad wordt een (verborgen) Markov-ketting genoemd, waarmee men wil aangeven dat

iedere volgende staat in dit pad enkel en alleen bepaald wordt door de huidige staat, en afhangt van diens transitieprobabiliteit [32]. De probabiliteit van zo een pad is het product van alle transitieprobabiliteiten en alle outputprobabiliteiten die voorkomen.
De volgende sectie beschrijft de algemene definitie en notaties die gebruikt worden voor het HMM. Vanaf sectie 5.3 wordt er dieper ingegaan op het gebruik van het HMM voor de identificatie van peptiden.

5.2 Definitie en notaties

Een HMM H wordt gedefinieerd door :

- Σ, een alfabet van symbolen;
- Q, een set staten, waarbij elk van deze staten een symbool van het alfabet Σ als output zal geven;
- $A = (a_k l)$, een $|Q|x|Q|$ matrix die de probabiliteit beschrijft om naar staat l te bewegen gedurende een stap waarin het HMM zich in staat k bevindt. Sommatie van deze probabiliteiten over alle staten l van het HMM (incusief de staat zelf waarin het HMM zich bevindt), voor een zekere staat k : $\sum_l a_k l = 1$; en
- $E = (e_k(b))$, een $|Q|x|A|$ matrix die de probabiliteit beschrijft waarmee symbool b als output gegeven wordt gedurende een stap waarbij het HMM in staat k is. Sommatie van deze probabiliteiten over alle symbolen b van het alfabet Σ, voor een zekere staat k : $\sum_b e_k(b) = 1$.

5.3 HMM voor peptidesequentie indentificatie

De hierboven beschreven algemene definitie van het HMM kan nu aangepast worden zodat ze betekenis krijgt voor het identificeren van peptidesequenties [31].
Aminozuren worden aangeduid met het symbool α, $\alpha \in \Sigma$, met Σ het alfabet van de 20 aminozuren. De massa van een aminozuur, afgerond tot op één Dalton nauwkeurig, wordt voorgesteld door $m(\alpha)$. Het experimenteel spectrum wordt gediscretiseerd in stappen van één Dalton, en kan op die manier aanzien worden als een sequentie van intensiteitswaarden. Het spectrum is dan de zogenaamde geobserveerde sequentie.
Zij S een experimenteel bekomen spectrum, dan wordt de intensiteit (piekwaarde) van een bepaalde massa in dit spectrum voorgesteld door $x(M)$, met M de massa in kwestie. Omdat de intensiteit niets anders is dan het aantal ionen met massa M dat in het spectrum voorkomt, wordt $x(M)$ ook wel *ionentelling* genoemd.
Zij $P = (\alpha_1, ..., \alpha_n)$ een peptide met parent massa $m(P)$:

$$m(P) = \sum_{i=1}^{n} m(\alpha_i) \tag{5.1}$$

We gaan er nog even van uit dat de parent massa die we zullen gebruiken correct is. Bij Sherenga zagen we reeds dat deze sterk kan afwijken van de theoretisch correcte waarde omwille van onnauwkeurigheden in de massameting, en dat men tracht deze parent massa nauwkeuriger te bepalen. Ook het algoritme NovoHMM herberekent de parent massa in een poging deze waarde nauwkeuriger te bepalen. Hier komen we later op terug.

5.3.1 Aangepaste definitie

De aangepaste definitie is dan als volgt :

- Σ, een alfabet van 20 symbolen die de 20 aminozuren voorstellen;

- Q, een set staten bestaande uit :
  - een beginstaat s_0 waaruit er vertrokken wordt : $\{s_0\}$;
  - twee eindstaten s_+ en s_- die respectievelijk aangeven of het resultaat positief of negatief is (meer hierover volgt verderop in de tekst) : $\{s_+, s_-\}$;
  - voor ieder aminozuur $\alpha \in \Sigma$ wordt er een *statenlijst* van $m(\alpha)$ staten $s_1^\alpha, ..., s_{m(\alpha)}^\alpha$ opgesteld (dus evenveel staten als het aantal Dalton[1] dat het aminozuur weegt) : $\left\{s_j^\alpha | \alpha \in \Sigma, 1 \leq j \leq m(\alpha)\right\}$.

  De volledige set staten S kan dus gegeven worden door :

$$S = \{s_0\} \cup \left\{s_j^\alpha | \alpha \in \Sigma, 1 \leq j \leq m(\alpha)\right\} \cup \{s_+, s_-\}\,; \tag{5.2}$$

- voor iedere staat de transitieprobabiliteiten. Deze worden uitgebreid besproken in sectie 5.3.2;
- voor iedere staat de outputprobabiliteiten. Deze worden uitgebreid besproken in sectie 5.3.4.

Beide probabiliteitswaarden worden als output gegeven bij het doorlopen van het HMM. In tegenstelling tot het algemene HMM worden de symbolen van het alfabet Σ *niet* als output gegeven.
Op dit moment moeten we echter eerst nog bepalen welke precies de probabiliteitswaarden zijn. Om het HMM op te stellen maken we, net zoals bij Sherenga, gebruik van een *training set*. Pas wanneer het HMM volledig is opgesteld kan het aangewend worden voor de identificatie van peptiden.

5.3.2 Transitieprobabiliteiten

De transitieprobabiliteit van een staat s om naar een staat t te gaan wordt aangegeven door *a(s,t)*. Zij Y_i de random variabele voor de statensequentie. De transitieprobabiliteiten worden dan gegeven door :

$$a(s,t) = P\left\{Y_{i+1} = t | Y_i = s\right\} = \begin{cases} 1 & \forall \alpha \in \Sigma,\ 1 \leq i < m(\alpha) : s = s_i^\alpha \wedge t = s_{i+1}^\alpha, \\ r_\alpha & \forall \alpha \in \Sigma,\ \beta \in \Sigma : s = s_{m(\beta)}^\beta \wedge t = s_1^\alpha, \\ 0 & \text{else.} \end{cases} \tag{5.3}$$

In woorden beschrijft de eerste regel het geval wanneer het HMM zich in een niet-eindstaat van de statenlijst van een aminozuur α bevindt. De staat waarnaar bewogen wordt *moet* dan de volgende staat uit deze statenlijst zijn en wordt dan ook aanvaard met probabiliteit 1. Een aminozuur wordt dus in stappen van één Dalton doorlopen totdat zijn massa (en dus zijn eindstaat) bereikt is. De tweede regel verwijst naar het geval dat het HMM zich net wel in zo'n eindstaat bevindt. De volgende staat kan dan de beginstaat zijn van eender welk aminozuur. Deze beginstaat wordt dan gekozen met een probabiliteit r_α, die aangeeft hoe groot de kans is dat aminozuur α in een sequentie volgt op het huidige aminozuur β. Vanzelfsprekend zijn de waarden van de verschillende r_α van groot belang om een goed resultaat te bekomen. Hierop komen we later terug (sectie 5.3.3). Het is ondertussen duidelijk geworden dat we bij het doorlopen van het HMM eigenlijk een aminozurensequentie vormen, en dus een peptide.
Tot slot moet voor de volledigheid vermeld worden dat in alle andere gevallen de probabiliteit 0 is. Dit is logisch aangezien er geen andere gevallen dan bovenvermelde mogelijk zijn. Het HMM bevindt zich immers altijd in een (al dan niet eind-) staat die behoort tot de statenlijst van een aminozuur. De speciale staten, beginstaat en eindstaten, hebben hun eigen probabiliteitsfuncties die hieronder beschreven worden.

[1] De massa's van de aminozuren worden hiervoor afgerond tot op één Dalton nauwkeurig.

Zoals gezegd zijn er één begin- en twee eindstaten toegevoegd aan het geheel van staten. Ook hiervoor zijn er probabiliteiten opgesteld. Deze worden weergegeven door formule 5.4. Het is duidelijk dat het HMM vanuit beginstaat s_0 naar een beginstaat van een aminozuur moet bewegen :

$$a(s_0, t) = \begin{cases} r_\alpha & \forall \alpha \in \Sigma : t = s_1^\alpha \\ 0 & \text{else} \end{cases} \tag{5.4}$$

Deze eerste stap naar de eerste staat van een aminozuur α gebeurt met een probabiliteit r_α, die hier in het bijzonder aangeeft hoe groot de kans is dat een peptide met het aminozuur α begint. Merk op dat dit eigenlijk bedoelt hoe groot de kans is dat het aminozuur α volgt op de aminozuren K of R[2], vermits één van deze twee aminozuren altijd het laatste aminozuur van de voorgaande peptide vormt[3]. Hebben we te doen met de allereerste peptide van de oorspronkelijke proteïne, dan weten we (hoofdstuk 1) dat de peptide met aminozuur M[4] zal beginnen. Het zijn zulke wetenswaardigheden die gebruikt worden om de exacte waarden van r_α te bepalen.
Bewegen naar een andere staat dan een eerste is niet toegelaten, dus alle andere gevallen hebben probabiliteit 0.

Tijdens het doorlopen van de staten maakt het HMM telkens stappen van 1 Dalton. Zodra de in beschouwing genomen parent massa $m(P)$ bereikt is, dus na $m(P)$ stappen, gelden de volgende probabiliteiten :

$$a'(s, t) = \begin{cases} 1 & \forall \alpha \in \Sigma : s = s_{m(\alpha)}^\alpha,\ t = s_+ \\ 1 & \forall \alpha \in \Sigma,\ 1 \leq i < m(\alpha) : s = s_i^\alpha,\ t = s_- \\ 0 & \text{else} \end{cases} \tag{5.5}$$

Is het HMM op dat moment in een eindstaat $s_{m(\alpha)}^\alpha$ van een aminozuur-statenlijst, dan wordt er bewogen naar de positieve eindstaat s_+ met probabiliteit 1. De parent massa is bereikt en kan worden samengesteld uit de aminozuren die doorlopen zijn. Is het HMM in een niet-eindstaat geëindigd, dan wordt er naar de negatieve eindstaat s_- bewogen. We hebben in dat geval een aantal aminozuren doorlopen en een *stuk* van het laatste aminozuur, wat leidt tot een onmogelijk bestaande sequentie. Het gevolgde pad heeft in dit geval niet geleid tot een mogelijke oplossing voor de gegeven parent massa. Dankzij de indeling van ieder aminozuur in stappen van één Dalton kan op een zeer eenvoudige manier bepaald worden of er al dan niet een geldige aminozurensequentie gevolgd is om de parent massa te bekomen.

Zoals reeds aangehaald wordt niet een waarde uit het gebruikte alfabet Σ als output gegeven, maar wel de probabiliteitswaarden. Bij iedere verplaatsing naar een volgende staat wordt er een transitieprobabiliteit als output gegeven. Bovendien wordt er in iedere staat nog een tweede probabiliteitswaarde als output gegeven, de outputprobabiliteit. Deze wordt besproken in sectie 5.3.4.

5.3.3 Bepaling van r_α

Het belangrijkste aspect uit de transitieprobabiliteiten is r_α. Stel dat het HMM zich in een zekere staat s bevindt, die ofwel de beginstaat s_0 is ofwel de eindstaat van één van de aminozuur-statenlijsten. Voor ieder aminozuur stelt r_α de probabiliteit voor waarmee er naar de beginstaat van aminozuur α bewogen kan worden vanuit de staat in kwestie. Met andere woorden, r_α is de kans dat een bepaald aminozuur α volgt op het aminozuur van de huidige staat.
Om een goed model tot stand te brengen zijn de waarden van deze r_α's uiteraard van zeer groot

[2] Door de knipeigenschap van het enzyme trypsine.
[3] Aan de *C-terminal*.
[4] Aminozuur Methionine.

belang. Om de r_α-waarden te kunnen bepalen wordt er gebruik gemaakt van gekende sequentiedata of *training set*, en de *maximum likelihood functie*. Door simpelweg te *kijken* naar gekende sequenties kan men tellen hoe vaak een aminozuur α_i volgt op een aminozuur α_j. We illustreren dit met een eenvoudig voorbeeld.
Onderstel dat er slechts twee aminozuren bestaan, A en B. De gekende sequenties waar we mee werken zijn : $\{ABBA, BBBB, AABB, BAAB\}$. Dan observeren we het volgende :

- aminozuren die volgen op A :
  - A : 2 keer
  - B : 3 keer
- aminozuren die volgen op B :
  - A : 2 keer
  - B : 5 keer

Uit de gegeven sequenties hebben we vijf gevallen waarin aminozuur A voorafgaat aan een bepaald aminozuur. Aminozuur A wordt twee keer gevolgd door A zelf, en drie keer door B. Willen we nu bijvoorbeeld de kans berekenen dat aminozuur A gevolgd wordt door aminozuur B, dan gaan we als volgt te werk. Zij X het *opvolgende* aminozuur, in dit geval dus B. Parameter p bepaalt de kans en is gelegen tussen 0 en 1. Dan wordt de *likelihood functie* als volgt gedefinieerd :

$$L(p) = P(X = 3) = \left\langle \begin{matrix} 5 \\ 3 \end{matrix} \right\rangle p^3(1-p)^{5-3} \tag{5.6}$$

Voor een bepaalde waarde van p geeft de likelihood functie aan hoe *aannemelijk* deze parameterwaarde is bij de gevonden steekproefuitkomst[5]. Vervolgens wordt er nagegaan voor welke waarde van p de likelihood functie *maximaal* is. Deze waarde is de meest aannemelijk schatting voor p. Na wat rekenwerk wordt het maximum van L gevonden voor $p = 0,6$. In woorden geldt dan : de kans p dat aminozuur A gevolgd wordt door aminozuur B is gelijk aan $0,6$.

Deze werkwijze wordt toegepast op de gekende sequentiedata, rekening houdend met de twintig aminozuren. Nu we voor ieder aminozuur de kans berekend hebben dat eender welk aminozuur erop volgt, kunnen we de transitieprobabiliteiten vervolledigen met de waarden van r_α die instaan voor de transitie naar de beginstaten van aminozuur-statenlijsten.

5.3.4 Outputprobabiliteiten

In iedere staat waar het HMM aankomt wordt een *outputprobabiliteit* als output gegeven. Het HMM geeft dus een sequentie van probabiliteitswaarden als output, met afwisselend een transitie- en een outputprobabiliteit in de sequentie.

Counter staten

De meest voorkomende fragmentatie is diegene die leidt tot de vorming van b- en/of y-ionen, waarbij de fragmentatie op een peptidebinding gebeurt. We vereenvoudigen het probleem weer en gaan er van uit dat de massapieken in het spectrum enkel afkomstig zijn van N-terminale ionen. In sectie 5.7 bespreken we de aanpassingen die het HMM moet ondergaan om in de praktijk, met reële data, te kunnen werken.
Beschouwen we een $b - H_2O$-ion, dan kunnen we berekenen dat de massa van een $b - H_20$-ion van een partiële peptide 18 Da minder weegt dan het b-ion van dezelfde partiële peptide[6]. Een $b - H_2O$-ion heeft dus een verschuiving van -18 Da t.o.v. het b-ion. Het b-ion heeft in het bijzonder dezelfde massa als de som van de aminozuren waaruit hij bestaat, plus 1 Da voor de H-molecule

[5] Hier is de steekproefuitkomst : *A wordt drie keer gevolgd door B.*
[6] Water of H_2O weegt immers 18 Da.

aan de *N-terminal*. We houden in dit algoritme geen rekening met de *terminals* en mogen dus deze ene Dalton laten vallen. Hierdoor is de massa van het b-ion gelijk aan de som van zijn aminozuren. Om geen verwarring te veroorzaken zullen we een N-terminaal ion zonder de H-molecule een *aminozuurion* noemen. Dit impliceert ook dat we een piek uit een spectrum voor massa m zullen aanzien als een piek voor massa $m - 1$.
De staten van het HMM krijgen elk een specifieke functie toegewezen. De staten binnen één aminozuur worden ook de *counter* staten genoemd omdat ze een *counter* in de massa van een aminozuur representeren. Het b-ion komt overeen met *geen counter* en wordt daarom gerepresenteerd door de staten $s_{m(\alpha)}$, de laatste staten van de aminozuren. Omdat de *counter* staten telkens een stap van 1 Da voorstellen wordt het $b - H_2O$-ion gerepresenteerd door de staat $s_{m(\alpha)-18}$. Analoog worden andere N-terminale ionen gerepresenteerd door andere staten uit de statenlijst. Staten die niet gerelateerd kunnen worden aan een ion type kunnen ruis voorstellen.

Outputprobabiliteiten van de staten

Telkens het HMM in een staat passeert wordt er een outputprobabiliteit als output gegeven. Om de waarde van deze probabiliteiten te bepalen maken we weer gebruik van onze *training set*.
Een beetje analoog aan de *intensiteitsthresholds* bij Sherenga gaat men ook hier aan de hand van een *training set* bepalen welke ion types in welke intensiteitsrangen voorkomen. De pieken[7] uit de *training set* worden hiervoor in gelijke *bins* verdeeld. Nummeren we deze *intensiteitsbins* van 1 tot en met k, met 1 de bin met de hoogste intensiteit, dan zullen de pieken van veel voorkomende ionen voorkomen in intensiteitsbins met een hoge rang, en de pieken van ionen met een lagere intensiteit of ruis zullen in bins met een lagere rang voorkomen. Voor alle N-terminale ion types wordt er gekeken in welke bins ze voorkomen, en met welke probabiliteit dit gebeurt[8]. Net zoals bij het berekenen van de transitieprobabiliteiten wordt er gewoon *gekeken* naar de spectra uit de *training set* om deze probabiliteitswaarden vast te stellen.
Alle staten krijgen nu, voor alle k bins, een probabiliteitswaarde. Voor iedere intensiteitsbin stellen deze waarden de kans voor dat een piek met zulk een intensiteit voorkomt op die plaats binnen een aminozuur-statenlijst. De eindstaten die de b-ionen voorstellen zullen dus voor de intensiteitsbins met een hoge rang een grote probabiliteit hebben, en een kleinere voor intensiteitsbins met een lagere rang. Het b-ion is immers een veel voorkomend iontype en *moet* bij wijze van spreken in een hoge intensiteit in het het spectrum aanwezig zijn. Een *counter* staat die geen functie heeft zal dan weer een grote probabiliteitswaarde krijgen voor intensiteitsbins met een lage rang, die ruis kunnen voorstellen. Sectie 5.3.6 zal dit verduidelijken.

5.3.5 Pseudo-code voor het opstellen van het HMM

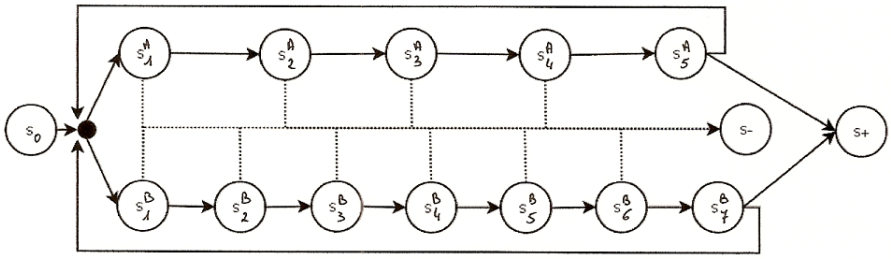
Tot hier toe hebben we *voorbereidend* werk gedaan. Zodra het HMM opgesteld is hebben we een werkende basis voor de identificatie van peptiden. In de volgende secties komt het gebruik van dit HMM aan bod. Het HMM op zich kan echter onveranderd blijven. Daarom geven we hier eerst de pseudo-code voor het opstellen van het HMM.

Listing 5.1: Pseuco-code : Opstellen van het HMM

```
voor alle aminozuren do
  maak statenlijst van het aminozuur;
maak beginstaat s0;
maak eindstaten s+ en s-;

stel transitieprobabiliteiten op;
stel outputprobabiliteiten op;
```

[7] Intensiteiten.
[8] Voor b-ionen bijvoorbeeld worden alle pieken uit alle spectra van de *training set* die b-ionen voorstellen in acht genomen.



Figuur 5.1: HMM voor het fictieve voorbeeld

5.3.6 Voorbeeld : het doorlopen van het HMM

Om de functie van de counter staten wat te verduidelijken geven we hier een fictief voorbeeld. Omdat het alfabet van twintig aminozuren veel te uitgebreid is zullen we ons houden aan het beperkte alfabet uit het vorige voorbeeld; het alfabet met twee aminozuren A en B. We geven het aminozuur A een massa van 5 Da en aminozuur B een massa van 7 Da. We gaan er van uit dat enkel het ion type b-ion en het fictieve ion type $b - H_2$-ion voorkomen, beide N-terminale ion types, waarbij het $b - H_2$-ion twee H-moleculen verloren heeft en 2 Da minder weegt dan een b-ion. Dit impliceert dat voor het aminozuur A de staat s_5^A de functie van b-ion zal hebben, en de staat s_3^A de functie van $b - H_2$-ion. Voor aminozuur B zijn dit respectievelijk de staten s_7^B en s_5^B.
Het HMM wordt weergegeven in figuur 5.1. Binnen een aminozuur-statenlijst en naar de eindstaten toe kennen we reeds de transitieprobabiliteiten. Tabel 5.1 geeft een overzicht van de transitieprobabiliteiten voor de overige gevallen. Tabel 5.2 tenslotte geeft een overzicht van de outputprobabiliteiten. We veronderstellen dat de intenstiteiten in de spectra die we willen identificeren met dit HMM onderverdeeld zijn in drie bins :

- bin 1 : intensiteiten strikt groter dan 10;
- bin 2 : intensiteiten strikt groter dan 5 en kleiner dan 10;
- bin 3 : intensiteiten kleiner of gelijk aan 5.

Al deze waarden zijn fictief en enkel van toepassing op dit voorbeeld.

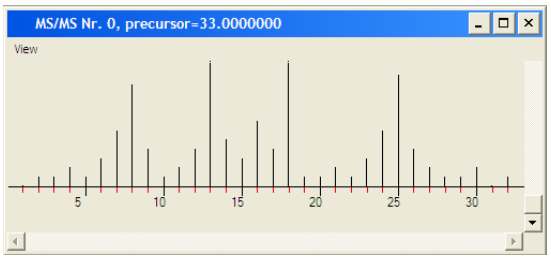
	s_1^A	s_1^B
s_0	0.4	0.6
s_5^A	0.5	0.5
s_7^B	0.9	0.1

Tabel 5.1: Transitieprobabiliteiten voor het fictieve voorbeeld

Het spectrum dat we willen identificeren wordt weergegeven in figuur 5.2 en is een spectrum van een parent peptide met $m(P) = 29$ Da. We behouden de *N-terminal* van 1 Da maar maken gebruik van een fictieve *C-terminal* van 2 Da. De parent peptide is bovendien éénwaardig geladen. De gemeten massa in het spectrum is daarom $(29 + 1 + 2 + 1)Da = 33Da$.
Rekening houdend met de *N-terminal* H die niet in rekening gebracht wordt kunnen we de geobserveerde sequentie uitschrijven zoals weergegeven in tabel 5.3. De bovenste regel geeft telkens de massa aan, de onderste regel de intensiteitswaarde van de piek voor die massa in het spectrum. Deze tabel is dus de geobserveerde sequentie.
We nemen aan dat er een algoritme is dat het meest waarschijnlijke staten-pad kan vinden en doorlopen het HMM alsof we weten hoe dit pad zal lopen. De functie van de *counter* staten zal hierdoor duidelijker worden. Voor de transitieprobabiliteiten gebruiken we tabel 5.1. Voor de outputprobabiliteiten gebruiken we de tabellen 5.3 en 5.2.

	bin 1	bin 2	bin 3
s_1^A	0.1	0.2	0.7
s_2^A	0.1	0.3	0.6
s_3^A	0.2	0.6	0.2
s_4^A	0.1	0.4	0.5
s_5^A	0.8	0.1	0.1
s_1^B	0.1	0.2	0.7
s_2^B	0.1	0.2	0.7
s_3^B	0.1	0.2	0.7
s_4^B	0.1	0.3	0.6
s_5^B	0.2	0.6	0.2
s_6^B	0.1	0.4	0.5
s_7^B	0.8	0.1	0.1

Tabel 5.2: Outputprobabiliteiten voor het fictieve voorbeeld



Figuur 5.2: Voorbeeldspectrum, gegenereerd met NovoHMM [19]

- stap 1 : $s_0 \rightarrow s_1^B$
  - transitieprobablititeit : 0.6
  - outputprobabiliteit : voor 1 Da hebben we een piek met intensiteit 1, wat overeenkomt met bin 3. In de staat s_1^B komt een piek in bin 3 voor met probabiliteit 0.7
- stap 2 : $s_1^B \rightarrow s_2^B$
  - transitieprobablititeit : 1.0
  - outputprobabiliteit : voor 2 Da hebben we een piek met intensiteit 1, wat overeenkomt met bin 3. In de staat s_2^B komt een piek in bin 3 voor met probabiliteit 0.7
- stappen 3 t.e.m. 5 : analoog met telkens volgende probabiliteiten :
  - transitieprobablititeit : 1.0
  - outputprobabiliteit : 0.7

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

Tabel 5.3: Geobserveerde sequentie voor het fictieve voorbeeld

- stap 6 : $s_5^B \rightarrow s_6^B$
  - transitieprobablititeit : 1.0
  - outputprobabiliteit : voor 6 Da hebben we een piek met intensiteit 6, wat overeenkomt met bin 2. In de staat s_6^B komt een piek in bin 2 voor met probabiliteit 0.4
- stap 7 : $s_1^B \rightarrow s_2^B$
  - transitieprobablititeit : 1.0
  - outputprobabiliteit : voor 7 Da hebben we een piek met intensiteit 11, wat overeenkomt met bin 1. In de staat s_7^B komt een piek in bin 1 voor met probabiliteit 0.8
- stap 8 : $s_7^B \rightarrow s_1^A$
  - transitieprobablititeit : 0.9
  - outputprobabiliteit : voor 8 Da hebben we een piek met intensiteit 4, wat overeenkomt met bin 3. In de staat s_1^A komt een piek in bin 3 voor met probabiliteit 0.7.
- stappen 9 t.e.m. 23 : analoog
- stap 29 : $s_4^A \rightarrow s_5^A$
  - transitieprobablititeit : 1.0
  - outputprobabiliteit : voor 29 Da hebben we een piek met intensiteit 1, wat overeenkomt met bin 3. In de staat s_5^A komt een piek in bin 3 voor met probabiliteit 0.1
  - **de parent massa is bereikt en we zitten in een eindstaat van een aminozuur**
- stap 30 : $s_5^A \rightarrow s_+$ met transitieprobabiliteit 1.0

De volledige output van het HMM wordt hieronder gegeven. Voor de leesbaarheid worden de probabiliteitswaarden twee aan twee van elkaar gescheiden met een vertikale lijn :

0.6 0.7 | 1.0 0.7 | 1.0 0.7 | 1.0 0.7 | 1.0 0.7 | 1.0 0.4 | 1.0 0.8 | 0.9 0.7 | 1.0 0.6 | 1.0 0.2 |
1.0 0.5 | 1.0 0.8 | 0.5 0.7 | 1.0 0.6 | 1.0 0.6 | 1.0 0.5 | 1.0 0.8 | 0.5 0.7 | 1.0 0.7 | 1.0 0.7 |
1.0 0.6 | 1.0 0.2 | 1.0 0.4 | 1.0 0.8 | 0.9 0.7 | 1.0 0.6 | 1.0 0.2 | 1.0 0.5 | 1.0 0.1 | 1.0

Output van het HMM

Hoe men, afgaande op de probabiliteitswaarden die als output gegeven worden, de aminozurensequentie reconstrueert, wordt niet in de literatuur aangehaald [19]. We zullen echter wel een eenvoudige methode beschrijven die de aminozurensequentie opnieuw kan samenstellen.
Gegeven de outputsequentie uit ons voorbeeld. We kunnen deze sequentie nu onderverdelen in aminozuren aan de hand van het aantal transitieprobabiliteiten die gelijk zijn aan 1.0 :

⟨0.60.7|1.00.7|1.00.7|1.00.7|1.00.7|1.00.4|1.00.8⟩ |
⟨0.90.7|1.00.6|1.00.2|1.00.5|1.00.8⟩ |
⟨0.50.7|1.00.6|1.00.6|1.00.5|1.00.8⟩ |
⟨0.50.7|1.00.7|1.00.7|1.00.6|1.00.2|1.00.4|1.00.8⟩ |
⟨0.90.7|1.00.6|1.00.2|1.00.5|1.00.1⟩ |
1.0

Aan de hand van deze indeling kunnen we al vaststellen dat de sequentie bestaat uit vijf aminozuren, respectievelijk met een massa van 7 Da, 5 Da, 5 Da, 7 Da en 5 Da. Omdat we slechts twee aminozuren in ons voorbeeldalfabet hebben is de oplossing snel gevonden : we hebben het spectrum geïdentificeerd en de oplossing is de peptide $BAABA$.
Bij het HMM voor het reële peptidesequentieprobleem kan men analoog te werk gaan. Enkel voor de aminozuren die dezelfde massa hebben zijn er meerdere mogelijke oplossingen. Bijvoorbeeld

aan de hand van de transitieprobabiliteiten kan men dan bepalen naar welk aminozuur het HMM bewogen heeft. Uiteraard kan de identificatie van de parent peptide niet overeenkomen met de werkelijke identiteit van deze parent peptide. Door onnauwkeurigheden in de metingen kan het zijn dat het meest waarschijnlijke staten-pad niet de juiste oplossing is. NovoHMM zoekt echter enkel het meest waarschijnlijke pad en maakt hiervoor gebruik van het *Viterbi* algoritme [31] [19].

5.3.7 Pseudo-code voor het doorlopen van het HMM

Ook hier geven we een stukje pseudo-code dat overeenkomt met het hierboven besproken gedeelte van NovoHMM. Deze pseudo-code geeft een overzicht van wat er binnen het HMM gebeurt indien het meest waarschijnlijke staten-pad zonder meer gevolgd kan worden. Uiteraard *weet* het HMM op voorhand niet welk het meest waarschijnlijke staten-pad is voor een bepaald spectrum. Hiervoor wordt het Viterbi algoritme gebruikt. Deze pseudo-code geeft eigenlijk een overzicht van wat het uiteindelijke resultaat van dit algoritme is, zonder de omwegen die het moet maken om het juiste pad te vinden.

Listing 5.2: Pseudo-code : Het doorlopen van het HMM

```
do
  ga naar een volgende staat;
  voeg de transitieprobabiliteit toe aan de outputsequentie;
  massa_count++;
  voeg de outputprobabiliteit toe aan de outputsequentie;
until (parent massa is bereikt);

if (huidige staat is een eindstaat van een aminozuur statenlijst)
  ga naar eindstaat s+;
  bewaar outputsequentie;
else
  ga naar eindstaat s-;
```

5.4 Bepaling van de meest waarschijnlijke sequentie

Gegeven het experimentele spectrum en het HMM, moeten we nog een methode hebben om de meest waarschijnlijke sequentie te vinden doorheen het model. Hiervoor wordt er gebruik gemaakt van het *Viterbi* algoritme. Het Viterbi algoritme gaat aan de hand van deze geobserveerde sequentie op zoek naar de meest waarschijnlijke opeenvolging van staten die, omgezet naar de overeenkomstige aminozurensequentie, deze geobserveerde sequentie als resultaat zou kunnen hebben in een spectrum. Dit staten-pad wordt ook wel het *Viterbi-pad* genoemd. Het Viterbi algoritme is een bekend algoritme. We zullen het hier daarom slechts zeer kort beschrijven. Kort samengevat onderzoekt het algoritme alle mogelijke paden doorheen het HMM, in dit specifieke geval alle mogelijke paden van s_0 naar s_+. Enkel het meest waarschijnlijke pad wordt bewaard. In iedere stap zal het algoritme naar de beste volgende staat zoeken van de mogelijke staten waarnaar er bewogen kan worden. Merk op dat in het HMM voor het peptidesequentieprobleem er in alle *counter* staten slechts één mogelijke volgende staat is.

5.4.1 Pseudo-code

Listing 5.3: Pseudo-code : Het Viterbi algoritme voor het peptidesequentieprobleem

```
begin in staat s_0;
repeat
  do
    beweeg naar de meest waarschijnlijke volgende staat;
    count++;
  until (count=m(P))
  if (eindstaat = s_+)
    bewaar outputsequentie;
until (alle mogelijke statensequenties zijn gevonden en beoordeeld)
bewaar enkel de outputsequentie met de hoogste probabiliteit;
```

5.5 Herberekening van de parent massa

Net zoals bij Sherenga wordt ook in dit algoritme de parent massa herberekend. Hiervoor wordt er gebruik gemaakt van het reeds opgestelde HMM. Zij M de parent massa, gekend uit het spectrum. Dan wordt de herberekende parent massa gegeven door :

$$\begin{aligned} \hat{M} &= \text{argmax}(M)\ P\{x|s_+, M\} \\ &= \text{argmax}(M)\ \Sigma_s P\{x, s|s_+, M\}, \end{aligned}$$

waarmee naar de waarde voor M gezocht wordt waarvoor de uitdrukking $\Sigma_s P\{x, s|s_+, M\}$ maximaal is. In woorden wordt er gezocht naar de waarde van M waarvoor de som van de probabiliteiten van alle mogelijke staten-sequenties voor die bepaalde parent massa het hoogst is. Deze som kan efficiënt berekend worden door het *Forward*- of het *Backward* algoritme [19]. Voor M worden meestal de waarden $m(P)$, $m(P)-1$ en $m(P)+1$ aangenomen, omdat proefondervindelijk is gebleken dat de gemeten parent massa meestal maximaal ongeveer 1 Da afwijkt van de werkelijke parent massa. Het beste resultaat wordt dan gebruikt als parent massa om de meest waarschijnlijke sequentie te bepalen.

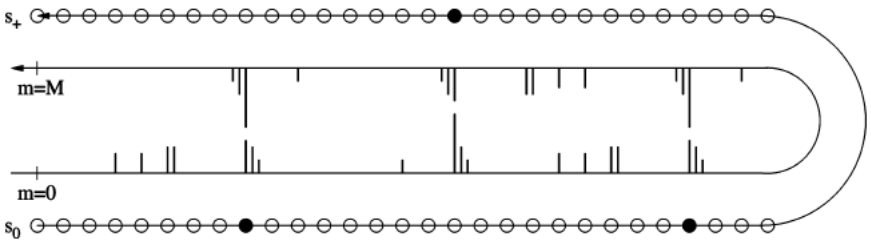
5.6 Pseudo-code

Listing 5.4: Pseuco-code : HMM

```
stel het HMM op met behulp van de training set;
herbereken de parent massa;
zoek het Viterbi-pad voor de geobserveerde sequentie;
```

5.7 Het HMM in de praktijk

De grootste vereenvoudiging die we hebben gemaakt is de aanname dat een spectrum enkel N-terminale ionen bevat. Bij het algoritme Sherenga zagen we reeds dat het wegvallen van deze vereenvoudiging heel wat moeilijkheden met zich mee kan brengen. In deze sectie beschrijven we hoe NovoHMM dit probleem oplost.
Voor C-terminale ionen kan er een analoog model opgesteld worden. Deze twee modellen moeten nu gecombineerd worden. Het grote, steeds terugkerende probleem is dat men niet weet welke pieken uit het spectrum N-terminale ionen voorstellen en welke C-terminale ionen. Bovendien vormt het *fake-twin-vertex*-probleem uit Sherenga ook hier een soortgelijk probleem. Indien er zich een piek in het spectrum bevindt voor een kleine massawaarde, dan is de kans immers groot



Figuur 5.3: Voorbeeld : een dubbelgevouwen spectrum citenovoHMM

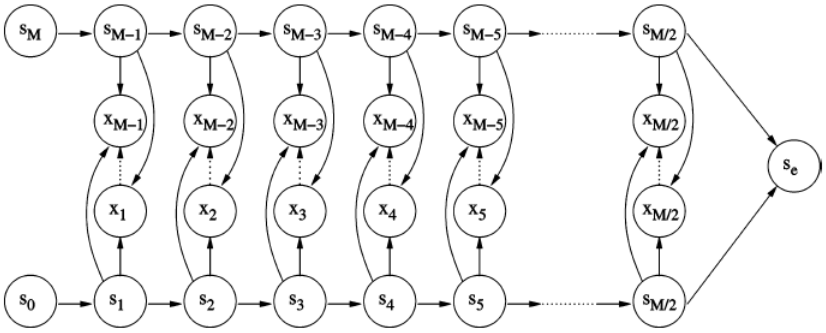
dat er ook een corresponderende piek voor een grote massawaarde in het spectrum aanwezig is. Dit zijn de corresponderende N- en C-terminale ionen voor bepaalde fragmentatieplaatsen. Om dit op te lossen worden er niet één maar twee Markov-kettingen gemaakt.

5.7.1 Twee Markov-kettingen

We herinneren de lezer eraan dat een Markov-ketting gewoon een staten-pad is waarin iedere volgende staat enkel en alleen bepaald wordt door de huidige staat waarin het HMM zich bevindt. Met andere woorden, een staten-pad zoals in het HMM voor het peptidesequentieprobleem.
Zij $m(P) = M$ de parent massa, dan passeert een Markov-ketting M staten. Deze Markov-ketting wordt nu in twee gedeeld, zodat er twee Markov-subkettingen ontstaan, elk met lengte $M/2$. De eerste subketting genereert een sequentie tot aan de helft van de parent massa. De tweede ketting begint aan de parent massa en genereert een *omgekeerd* sequentie, ook weer tot aan de helft van de parent massa. We kunnen nu het spectrum als het ware in twee vouwen. Figuur 5.3 geeft zo een dubbelgevouwen spectrum weer. Het spectrum in de figuur is afkomstig van een peptide bestaande uit vier aminozuren. De zwart gekleurde bolletjes geven de aminozuurgrenzen aan, en bepalen dus de omgeving waar mogelijke fragmentatieplaatsen zich bevinden. Dit is uiteraard informatie die niet gekend is bij de identificatie van de peptide. We merken wel op in de figuur dat de grotere pieken in de twee helften met elkaar overeenkomen. Deze grotere pieken stellen de b- en de y-ionen voor. Als voorbeeld bekijken we even de piek boven het eerste zwarte bolletje. Deze aminozuur-grens genereert duidelijk een piek voor een kleinere massa (N-terminaal ion) en een overeenkomstige piek (C-terminaal ion) voor een grotere, complementaire massa. Dat beide pieken complementair zijn in massa, ten opzichte van de parent massa, blijkt duidelijk uit de figuur, en is logisch aangezien b- en y-ionen complementair zijn. Bij het derde ingekleurde bolletje, bovenaan, krijgen we een N-terminaal ion voor een grotere massa, en een overeenkomstig C-terminaal ion voor een kleinere massa. Merk op dat het benoemen van deze pieken met N- en C-terminaal ion enkel mogelijk is omdat we weten waar de aminozuur-grenzen liggen. We herhalen nogmaals dat dit geen gekend gegeven is bij de identificatie, maar dat we dit hier enkel in het voorbeeld gebruiken om de werking van de twee Markov-kettingen te kunnen verduidelijken.

5.7.2 Afhankelijkheidsstructuur

De twee Markov-kettingen zijn afhankelijk van elkaar omdat ze voor een groot deel complementaire ionen bevatten. De outputprobabiliteiten worden daarom ook geherdefinieerd en worden afhankelijk van twee staten. Dit is mogelijk omdat beide kettingen tegelijkertijd gegenereerd worden. Figuur 5.4 is een schematische voorstelling van de afhankelijkheidsstructuur van de twee Markov-kettingen. De staat s_e staat hier voor de eindstaat. De twee pieken voor x_1 en $x_{M-1} = x_{m(P)-1}$ worden tegelijkertijd in beschouwing genomen. Op basis van de outputprobabiliteiten van beide staten s_1 en s_{M-1} worden vervolgens de outputprobabiliteiten bepaald. In iedere stap worden er dus twee transitieprobabiliteiten en twee outputprobabiliteiten gegeven. De outputprobabiliteiten



Figuur 5.4: Voorbeeld : afhankelijkheidsstructuur voor het HMM [19]

worden dan gedefinieerd door :

$$\begin{aligned} e_{s_m,s_{M-m}}(x_m) &= P(x_m|s_m, s_{M-m}) \\ e_{s_{M-m},s_m}(x_{M-m}) &= P(x_{M-m}|s_{M-m}, s_m) \end{aligned}$$

Op de figuur zijn de overeenkomstige pieken x_i en x_{M-i} met een stippelijn-pijl verbonden. Door de symmetrie van de b- en de y-ionen in het dubbelgevouwen spectrum zijn de massawaarden waarop de pieken zich bevinden complementair. De stippelijn-pijlen duiden dit verband tussen de pieken x_i en x_{M-i} aan.

5.7.3 Benadering van het model

Omdat dit model computationeel veel tijd in beslag neemt tracht men het te benaderen in plaats van exact te genereren. Er wordt hiervoor gebruik gemaakt van zogenaamde *N/C-bits*. Daarmee tracht men te voorspellen of een piek een N-terminaal ion of een C-terminaal ion voorstelt. Er zijn recent nog andere methoden uitgewerkt om te achterhalen welke ionen uit een spectrum N-terminaal en welke C-terminaal zijn. Het zou ons echter te ver leiden om hier verder op in te gaan. De geïnteresseerde lezer kan terugvallen op de referenties [19] en [33].

Hoofdstuk 6

Lutefisk

Lutefisk [14] is een algoritme dat een *de novo* algoritme combineert met het gebruik van een database. De eerste fase van het algoritme is het *de novo* gedeelte en is, net als Sherenga, gebaseerd op de grafentheorie. De output hiervan zijn aminozurensequenties. Deze gevonden sequenties worden dan gebruikt om in een database op zoek te gaan naar de *proteïnen* waaruit de sequenties afkomstig kunnen zijn. Hiervoor wordt er gebruik gemaakt van een database zoekprogramma, *CIDentify* gedoopt. Deze database-stap is, in tegenstelling tot de eerder besproken algoritmen, ook in Lutefisk geïntegreerd.
Belangrijk om weten is dat Lutefisk gemaakt is voor spectrumdata die bekomen is door CID, een low-energy ionisatie methode. Deze methode brengt specifieke iontypes voort waar Lutefisk rekening mee houdt. De iontypes die in dit hoofdstuk voorkomen zullen ondermeer enkel eenwaardig geladen zijn. Verder zijn het de meest voorkomende ion types zoals b- en y-ionen die door CID voortgebracht worden. Hiervoor moeten we dus geen extra kennis opdoen.
Het grote verschil met de eerder besproken *de novo* algoritmen is dat Lutefisk niet enkel op zoek gaat naar volledig geïdentificeerde sequenties[1], maar ook naar peptide sequenties die, in tegenstelling tot *normale* sequenties die een opeenvolging van aminozuren zijn, ook niet-geïdentificeerde delen kunnen bevatten die de massa van een di-peptide voorstellen[2]. Een sequentie kan er dan als volgt uitzien :

HLI[248]SR

Hierbij stelt [248] een dipeptide voor met een massa van 248 Da.
We houden in het achterhoofd dat de sequenties waarover we spreken mogelijk niet volledig geïdentificeerd zijn.

6.1 Waarom een combinatie algoritme?

De reeds besproken algoritmen vertrekken van een experimenteel massaspectrum en trachten de parent peptide te identificeren. De identificatie van peptiden is echter vaak maar een 'tussenstation', daar het uiteindelijk doel is de proteïne te identificeren waarvan deze peptide afkomstig is. Het zijn immers de proteïnen waarvan bekend is welke eigenschappen en functies ze hebben.
Indien men te maken heeft met peptiden afkomstig van nieuwe of ongekende proteïnen, dan is het onmogelijk om de identificatie van de proteïnen uit een database te halen. In dat geval zijn de *de novo* methoden een goede manier om de aminozurensequenties van de peptiden van deze proteïne te reconstrueren.
Een geïdentificeerde peptide die afkomstig is van een gekende proteïne[3] kan wél gebruikt worden als zoekstring om in een database te achterhalen van welke proteïne ze afkomstig is. De peptide

[1] Een sequentie waarvan alle aminozuren geïdentificeerd zijn.
[2] Merk op dat bij Sherenga deze di- (en ook tri-) peptiden geïdentificeerd werden door de gelabelde bogen.
[3] De gegevens van gekende proteïnen zijn opgeslagen in de proteïne databases.

identificatie is dan slechts een tussenstap van het onderzoek, en wordt aangevuld met een database onderzoek. Omdat in vele onderzoeken juist deze proteïne identificatie van groot belang is, heeft men in Lutefisk een database zoekalgoritme geïntegreerd om, zonder naar een ander algoritme te moeten grijpen, deze proteïne identificatie te kunnen bekomen. Zoals in het inleidende deel van deze tekst aangehaald werd, is het database onderzoek geen onderdeel van deze thesistekst. Omdat Lutefisk dit echter geïntegreerd heeft in zijn algoritme zal ook de database methode kort besproken worden. Een massaspectrum als input voor Lutefisk levert dus niet enkel mogelijke identificaties op van de parent peptiden in kwestie, maar ook mogelijke proteïnen waaruit de verschillende peptiden afkomstig kunnen zijn.

6.2 Het *de novo* algoritme

Het *de novo* algoritme dat in Lutefisk gebruikt wordt bestaat uit vijf stappen, vertrekkende van een experimenteel massaspectrum. De *file* met de gegevens van het massaspectrum wordt de *inputfile* genoemd.

6.2.1 Stap 1 : Identificatie van significante ionen

In de eerste stap wordt de inputfile gereduceerd om enkel de significante fragmentatie-ionen over te houden. Het bepalen van het al dan niet significant zijn van de ionen wordt volledig door de gebruiker bepaald.
Eerst en vooral bepaalt de gebruiker een *m/z-venster*[4] en een *threshold*[5]. Het m/z-venster verdeelt het spectrum in stappen ter grootte van het venster. Voor ieder venster worden nu de volgende regels toegepast om de spectrumdata binnenin het venster te reduceren :

- enkel lokale maxima boven de threshold zullen als data (en niet als ruis) beschouwd worden,
- er wordt slechts een maximum *max* aantal ionen toegelaten binnen ieder venster. Indien er na het toepassen van de threshold nog meer dan *max* ionen in het venster aanwezig zijn, worden enkel de *max* grootste pieken behouden. Ruis maar ook lage-intensiteit ionen worden op deze manier verwijderd.

Tot slot wordt er een globale limiet toegepast op het aantal ionen dat in de gereduceerde lijst overblijft. Deze limiet heeft uiteraard enkel nut indien de gebruiker deze limiet een kleinere waarde geeft dan het aantal m/z-vensters vermenigvuldigd met *max*.
Binnen ieder m/z-venster worden de resterende pieken nu uitgemiddeld zodat ze allemaal dezelfde m/z waarde krijgen. Er wordt dus van uit gegaan dat deze pieken allen hetzelfde fragmentatie-ion voorstellen en dus samengevoegd kunnen worden tot één enkele piek.

Het resultaat van stap 1 is een gereduceerde lijst spectrumdata, beperkt tot significante ionen.

6.2.2 Stap 2a : Conversie naar overeenkomstig b-ion

In het vervolg van deze stap zullen alle ionen uit het gereduceerde spectrum mathematisch omgezet worden in hun corresponderende b-ion massa. De herberekening gebeurt aan de hand van tabel 6.1. Deze tabel bevat mathematische conversieformules van N-terminale en C-terminale ionen naar b-ionen. Lutefisk gaat er hierbij van uit dat N-terminale iontypes b, $b - NH_3$, $b - H_2O$, $a-$, $a - NH_3$ en $a - H_2O$ in het spectrum voorkomen (CID). Voor de C-terminale ionen worden ion types y, $y - NH_3$ en $y - H_2O$ in acht genomen. De C-terminale ionen worden omgerekend met behulp van de parent massa $m(ParentIon)$. Merk op dat we hier de totale massa van het parent ion bedoelen, en niet enkel de som van de aminozuren[6] waaruit het bestaat. Ook de massa's van de N- en de *C-terminal* worden in rekening gebracht.

[4] Het m/z-venster wordt bepaald afhankelijk van de resolutie van de gebruikte ionisatiemethode in de massaspectrometer.
[5] De threshold is een fractie van de gemiddelde intensiteit in het spectrum.
[6] De som van de aminozuren wordt voorgesteld door $m(P)$.

N-terminale ionen	Conversieformule
b	m(ion)
$b - NH_3$	m(ion) + 17
$b - H_2O$	m(ion) + 18
a	m(ion) + 28
$a - NH_3$	m(ion) + 45
$a - H_2O$	m(ion) + 46
C-terminale ionen	Conversieformule
y	(m(ParentIon) - m(ion)) + 2
$y - NH_3$	(m(ParentIon) - m(ion)) - 15
$y - H_2O$	(m(ParentIon) - m(ion)) - 16

Tabel 6.1: Conversieformules voor N- en C-terminale ionen [14]

We verduidelijken even de correctheid van deze formules.

N-terminale conversieformules

Een b-ion moet uiteraard niet omgezet worden en behoudt dus zijn originele massawaarde. De b-ionen met het verlies van moleculen kunnen eenvoudigweg teruggebracht worden tot gewone b-ionen door het verlies er terug bij te tellen; NH_3 weegt 17 Da en H_2O weegt 18 Da.
Om de formule voor een a-ion te controleren maken we gebruik van de formules in tabel 2.1. De massa van een b-ion is gelijk aan $S+1$, die van een a-ion is gelijk aan $S-27$. Een korte berekening geeft het volgende :

$$\begin{aligned} m(a\text{-ion}) + 28 &= (S - 27) + 28 \\ &= S + 1 \\ &= m(b\text{-ion}) \end{aligned}$$

Door 28 Da op te tellen bij de massa van een a-ion bekomen we dus de massa van het overeenkomstige b-ion. a-ionen met verlies van moleculen kunnen eerst in gewone a-ionen omgezet worden door het verlies in massa terug op te tellen. Met bovenstaande berekening bekomen we dan de overeenkomstge b-ionen. Voor een $a - H_2O$-ion geeft dit :

$$(m(a - H_2O\text{-ion}) + 18) + 28 = m(a - H_2O\text{-ion}) + 46$$

Voor alle andere N-terminale ionen kunnen de formules op analoge wijze afgeleid worden.

C-terminale conversieformules

Bij C-terminale ionen is de berekening iets minder voor de hand liggend. De massa van een y-ion is volgens de formule uit tabel 2.1 gelijk aan $S+19$. Beschouwen we nu even de C-terminale partiële peptide P_C (dus niet geïoniseerd) die overeenkomt met een bepaald y-ion. De massa van P_C is dan gelijk aan $S + m(OH)$, met OH de C-terminal. De moleculen OH wegen samen 17 Da. De bijdrage die de ionisatie aan deze partiële peptide levert is dus gelijk aan (19 Da - 17 Da) = 2 Da. De N-terminale partiële peptide, P_N, die complementair is aan P_C, heeft een massa gelijk aan $m(ParentIon) - P_C$. De massa van P_N is gelijk aan $S + m(H)$, met H de N-terminal. H weegt 1 Da. De massa van een b-ion is gelijk aan $S + 1$ (tabel 2.1). De ionisatie van een b-ion levert dus

geen bijdrage in massa (voor details : zie bijlage D). Hiermee rekening houdend bekomen we het volgende :

$$\begin{aligned} m(ParentIon) - m(y\text{-ion}) + 2 &= m(ParentIon) - (S + 19) + 2 \\ &= m(ParentIon) - S - 17 - 2 + 2 \\ &= m(ParentIon) - (S + 17) \\ &= m(ParentIon) - P_C \\ &= m(b\text{-ion}) \end{aligned}$$

Voor het omzetten van een y-ion met verlies van moleculen naar een b-ion gaan we weer analoog te werk zoals voordien.

6.2.3 Stap 2b : Bepaling van de N- en C-terminale *evidence*-lijsten

Nu de conversieformules gekend zijn kunnen we de *evidence*-lijsten opstellen. Deze lijsten weerspiegelen de waarschijnlijkheid dat bepaalde ion types in het spectrum voorkomen.

N-terminale *evidence*-lijst

Lutefisk beschouwt eerst alle ionen uit het gereduceerde spectrum als b-ionen. Alle indexposities in de lijst die overeenkomen met de, tot op één Dalton nauwkeurig, afgeronde ionmassa's krijgen nu een waarde. Deze waarde is gelijk aan de probabiliteitswaarden volgens *Bartels* [34] en heeft betrekking tot het belang of de significantie van het ion type in kwestie (hier het b-ion).
Ieder ion uit het spectrum wordt vervolgens beschouwd als een $b - NH_3$-ion. Alle ion massa's worden met behulp van de conversieformule omgerekend naar het overeenkomstige b-ion. Indien de berekende waarde in het spectrum en dus in de lijst voorkomt, dan wordt de waarde op deze indexpositie in de lijst verhoogd met de probabiliteitswaarde van een $b - NH_3$-ion. Deze werkwijze steunt op het *proefondervindelijk* feit dat de kans op eender welk ander N-terminaal ion type dan een b-ion van een bepaalde partiële peptide kleiner is dan de kans op een b-ion van deze partiële peptide. De aanwezigheid van niet-b-ionen van een partiële peptide zal daarom altijd gepaard gaan met de sterkere aanwezigheid van b-ionen van deze partiële peptide.
Volledig analoog worden de ionen uit het spectrum beschouwd als $b - H_2O$-, a-, $a - NH_3$- en $a - H_2O$-ionen.

C-terminale *evidence*-lijst

Om de C-terminale *evidence*-lijst op te stellen worden eerst alle ionen beschouwd als y-ionen. Alle waarden worden dan omgezet naar hun overeenkomstige b-ion-waarde. De indexplaatsen uit de C-terminale *evidence*-lijst die overeenkomen met de bekomen waarden, krijgen nu de probabiliteitswaarde volgens *Bartels* voor y-ionen. Merk op dat deze probabiliteitswaarde voor een y-ion wordt toegekend aan indexposities die overeenkomen met de overeenkomstige b-ionwaarde van deze y-ionen. Vervolgens worden alle ionen respectievelijk beschouwd als $y - NH_3$- en $y - H_2O$-ionen.
Volledig analoog aan het opstellen van de N-terminale *evidence*-lijst wordt de C-terminale *evidence*-lijst aangevuld. Ook hier gaat men er van uit dat een C-terminaal, niet-y-ion enkel voorkomt in het spectrum indien het y-ion zelf van deze partiële peptide ook in het spectrum aanwezig is. Na het omrekenen naar het overeenkomstige y-ion wordt het bekomen resultaat nog eens extra omgezet naar het corresponderende b-ion. Zowel de N-terminale als de C-terminale *evidence*-lijst bevatten dus enkel mogelijke ionmassa's van b-ionen.

Voorbeeld

Het opstellen van de *evidence*-lijsten illustreren we met een eenvoudig voorbeeld [14]. Onderstel dat een spectrum van een parent ion, met een massa van 999.5 Da, pieken bevat voor volgende massawaarden :

$$[155.0, 173.1, 184.1, 201.1, 783.3, 800.4]$$

Ieder van deze ionen wordt eerst beschouwd als zijnde een b-ion. De N-terminale *evidence*-lijst krijgt waarden op de indexposities $[155, 173, 184, 201, 783, 800]$ die overeenstemmen met de *waarschijnlijkheidswaarden* voor het iontype, beschreven door *Bartels* [34], en afhankelijk van het belang of van de significantie van het iontype in kwestie, hier het b-ion.
Vervolgens beschouwen we alle ionen als zijnde een $b - H_2O$ ion en berekenen we het overeenkomstig b-ion volgens tabel 6.1. Voor conversie van een $b - H_2O$ ion naar zijn overeenkomstig b-ion moeten we de formule m(ion) + 18 gebruiken. We krijgen dan volgende reeks :

$$[155.0 + 18, 173.1 + 18, 184.1 + 18, 201.1 + 18, 783.3 + 18, 800.4 + 18]$$
$$=$$
$$[173, 191, 202, 219, 801, 818]$$

We zien dat de waarde 155 voor een $b - H_2O$-ion overeenkomt met de waarde 173 voor een b-ion. Er is dus een partiële peptide die zowel onder de vorm van een b-ion als onder de vorm van een $b - H_2O$-ion voorkomt. De indexpositie 173 in de N-terminale *evidence*-lijst wordt verhoogd met de waarschijnlijkheidswaarde van een $b - H_2O$-ion.
Doen we dit analoog voor $b - NH_3$-ionen, dan zullen we merken dat massawaarden 201 en 800 overeenstemmen met het b-ion van de $b - NH_3$-ionen met massawaarden 184 en 783. De indexposities op 184 en 783 worden dus verhoogd met de waarschijnlijkheidswaarde van een $b - NH_3$-ion.
Voor andere N-terminale iontypes gaan we op dezelfde manier te werk. Voor C-terminale ionen beschouwen we eerst alle ionen als een y-ion en converteren ze naar hun overeenkomstig b-ion. Voor de C-terminale *evidence*-lijst krijgen volgende indexposities de waarschijnlijkheidswaarde van een y-ion :

$$[201, 218, 800, 817, 828, 846]$$

Vervolgens worden alle pieken beschouwd als een ander C-terminale iontype en worden de overeenkomstige y-ionen berekend. Voor de ionen die voorkomen in het spectrum, wordt de waarde van de overeenkomstige b-ion-indexpositie in de C-terminale *evidence*-lijst verhoogd met de waarschijnlijkheidswaarde van het iontype in kwestie.
Merk op dat *alle* pieken uit het spectrum als b- en als y-ion beschouwd worden, om vervolgens te zoeken naar andere resp. N- en C-terminale iontypes. Hierdoor krijgt iedere piek de kans om eender welk iontype voor te stellen.

6.2.4 Stap 3 : Bepaling van het sequentie spectrum

De N- en C-terminale *evidence*-lijsten worden samengevoegd en leiden tot het zogenaamde *sequentie spectrum*. De waarden op iedere indexpositie worden hierbij gesommeerd. De b-ionmassa's die zowel in de N-terminale als in de C-terminale *evidence*-lijst voorkomen worden dus opgeteld. Dit impliceert dat er voor het b-ion in kwestie ook een complementair y-ion in het spectrum aanwezig is, wat een extra bewijs levert voor een fragmentatie van de parent peptide op die plaats.
De N- en *C-terminal* van de parent peptide komen niet voor in het spectrum maar zijn wel onderdeel van resp. de N-terminale en de C-terminale ionen. De indexposities die overeenstemmen met deze *terminals* krijgen daarom een willekeurige waarde toegekend in het sequentie spectrum[7].
Om het sequentie spectrum te vervolledigen kent Lutefisk een *bonus probabiliteitsscore* toe in bepaalde gevallen. Als we in het achterhoofd houden dat de indexposities in het sequentie spectrum eigenlijk de massa's van mogelijke b-ionen voorstellen, dan kunnen we zoeken naar indices die 'verlengd' kunnen worden tot aan de *C-terminal* door de massa's van aminozuren bij te tellen. De waarden van deze indices worden verhoogd met een bonus. Er wordt enkel gekeken *of* het mogelijk is de sequentie te verlengen naar de *C-terminal*, niet *hoe* dat mogelijk is. Het algoritme zal dus nagaan of het verschil in massa tussen de *C-terminal*-representatie en de index in kwestie kan samengesteld worden uit de som van aminozuren. Voor indices, en dus massawaarden, die hieraan voldoen, wordt er ook gekeken of ze, door telkens de massa van een aminozuur af te trekken, ook

[7] Indexpositie 1 voor de *N-terminal* en *(m(ParentIon)−17)* voor de *C-terminal*.

verbonden kunnen worden met de *N-terminal*. Gaat dit niet, dan wordt de piek uit het sequentie spectrum in kwestie opgeslagen als een zogenaamde *one-degree-node*. Deze *one-degree-nodes* worden in de volgende stap gebruikt wanneer men een sprong wilt maken. Merk op dat we hier niet noodzakelijk bedoelen dat de knoop in kwestie niet bereikbaar is vanuit de *N-terminal*, maar wel dat dit niet lukt zonder het gebruik van di- of tripeptiden.
We hebben reeds aangehaald dat dit algoritme net zoals Sherenga gebaseerd is op de grafentheorie. De indexen uit het sequentie spectrum kunnen beschouwd worden als de knopen van de graaf. Deze knopen hebben als massawaarde de index uit het sequentie spectrum. Bovendien krijgen ze een score die overeenkomt met de waarde van die index in het sequentie spectrum. In tegenstelling tot Sherenga worden er *geen* bogen getrokken, waardoor er ook geen nood is aan een padalgoritme. De volgende sectie licht het zoeken naar sequenties toe. Door het gebrek aan bogen kunnen we ook niet echt spreken van een graaf. In plaats van bogen te trekken zullen we in de volgende sectie zien dat de paden opgebouwd worden uit sequenties van de knopen. Omdat deze aanpak gebaseerd is op de grafentheorie zullen we de benaming *knopen* blijven gebruiken.

6.2.5 Stap 4 : Generatie van sequenties

Vertrekkende van de *N-terminal* worden de sequenties nu gezocht doorheen het sequentie spectrum. Eerst worden *alle b*-ion waarden (knopen uit de graaf) gezocht die gelijk zijn aan de massa van één aminozuur. Omdat het vaak voorkomt dat er geen N-terminale ionen bestaande uit slechts één aminozuur in het spectrum voorkomen, wordt er ook meteen gezocht naar knopen die een waarde hebben gelijk aan de som van twee (willekeurige) aminozuren. Het aantal toegelaten sprongen van bi-peptiden ná dit punt is een parameter die door de gebruiker vastgelegd wordt. Meestal bedraagt deze limiet slecht één enkele extra bi-peptide-sprong per gevonden sequentie.
Iedere knoop uit deze gevonden *initiële lijst* stelt een kleine sequentie voor van één of twee aminozuren, ook wel *subsequentie* genoemd. Deze bevat ofwel het gevonden aminozuur in kwestie, ofwel de massa van de di-peptide. De reden voor het opslaan van de massa en niet de mogelijke identificaties voor de di-peptide is o.a. het besparen van computergeheugen. We zullen in sectie 6.3 zien dat dit geen probleem vormt om in de database-stap op zoek te gaan naar proteïnen, maar dat deze werkwijze nog andere voordelen heeft.
Iedere subsequentie wordt nu stap voor stap uitgebreid door telkens te zoeken naar een knoop die verschilt in de massa van één aminozuur (méér), rekening houdend met het aantal toegelaten di-peptiden dat nog mag voorkomen. Op die manier wordt een sequentie opgebouwd vanuit de *N-terminal*. Door de conversies naar overeenkomstige *b*-ionen hebben we hier niet het probleem van Sherenga dat de knopen zowel N-terminale als C-terminale ionen kunnen voorstellen. Het maximum aantal subsequenties is ook weer een *user-defined* parameter en is afhankelijk van het computer geheugen. Alle subsequenties worden daarom gesorteerd op basis van hun *score*. Deze score is niets meer dan de som van de ion probabiliteitswaarden van de knopen in de subsequentie. Overtollige subsequenties worden verwijderd uit de lijst op basis van deze score.
Een laatste *user-defined* parameter is het aantal verschillende aminozuren die één bepaalde subsequentie kunnen verlengen. Indien er meerdere mogelijkheden zijn om de subsequentie te verlengen dan worden enkel díe knopen behouden met de hoogste probabiliteiten.
De toegelaten bi-peptiden mogen eender waar in de subsequentie voorkomen, maar het gebruik ervan wordt wel 'bestraft' door de score van de subsequentie te verlagen, tenzij deze bi-peptide gebruikt wordt om de subsequentie te verlengen naar een *one-degree-node*. Het zijn juist deze *one-degree-nodes* die niet naar de *N-terminal* verlengd konden worden met sprongen van één enkel aminozuur. Een di-peptide-sprong naar deze *nodes* wordt daarom als *normaal* beschouwd en dus niet bestraft in de score.

Zodra de subsequentie de parent massa bereikt, wordt de sequentie opgeslagen voor de finale toekenning van een score en de ranking van alle gevonden sequenties.

6.2.6 Stap 5 : Scoring en ranking van de sequenties

De gevonden sequenties krijgen tot slot van het *de novo* algoritme een score. Deze score is gebaseerd op het aantal fragmentatie-ionen uit het spectrum dat gelinkt kan worden aan gekende iontypes, m.a.w., het aantal geïdentificeerde knopen dat voorkomt in de sequenties. Hierbij wordt er ook rekening gehouden met het feit dat een knoop zowel van een N-terminaal als van een C-terminaal ion afkomstig kan zijn. De sequentie van aminozuren representeert een opeenvolging van knopen uit de graaf, oftewel indices uit het sequentie spectrum. Deze indices zijn ofwel afkomstig uit de N-terminale ofwel uit de C-terminale *evidence-lijst*. De aanwezige fragmentatie-ionen in het spectrum van een parent peptide zijn in de praktijk meestal een opeenvolging van N-terminale ionen tot ongeveer de helft van het parent ion, en van C-terminale ionen voor de andere helft[8]. De corresponderende fragmentatie-ionen kunnen uiteraard ook voorkomen in het spectrum, maar het omgekeerde geval komt niet zomaar voor in de praktijk.
We geven een klein theoretisch voorbeeldje. Stel dat we de parent peptide HLITFSR hebben. Dan is het hoogst waarschijnlijk dat van volgende partiële peptiden er een ion in het spectrum aanwezig is :

- N-terminale ionen :
  - H
  - HL
  - HLI
- C-terminale ionen :
  - R
  - SR
  - FSR

Als we het parent ion van links naar rechts doorlopen en telkens de fragmentatieplaats tussen twee aminozuren beschouwen, dan zien we hier dat we een reeks ionen krijgen die als volgt ingedeeld kunnen worden : N-terminaal ion, N-terminaal ion, N-terminaal ion, C-terminaal ion, C-terminaal ion, C-terminaal ion.
Een spectrum met bijvoorbeeld ionen van volgende partiële peptiden is veel minder voor de hand liggend :

- N-terminale ionen :
  - HL
  - HLIT
  - HLITFS
- C-terminale ionen :
  - LITFSR
  - TFSR
  - SR

Doorlopen we het parent ion weer op dezelfde manier, dan zien we hier dat we een reeks ionen krijgen die als volgt ingedeeld kunnen worden : C-terminaal ion (LITFSR), N-terminaal ion (HL), C-terminaal ion (TFSR), N-terminaal ion (HLIT), enz. Het *terminal-type* occileert als het ware tussen N-terminaal en C-terminaal.
De gevonden sequentie in kwestie zal ook een bepaald patroon vertonen. Een patroon dat aanleunt bij het eerste voorbeeld zal een hoge score krijgen. Een patroon dat aanleunt bij het laatste

[8] Merk op dat dit, net zoals zoveel andere aspecten die we reeds zijn tegengekomen, weer gebaseerd is op proefondervindelijke resultaten.

voorbeeld komt overeen met een onwaarschijnlijke opeenvolging van ionen en zal een lage score krijgen.
De sequenties worden nu een eerste keer gerangschikt. De best scorende sequenties ondergaan een bijkomende rankingprocedure door middel van een *cross-correlation* analyse, analoog aan de methode beschreven in [16]. De literatuur over Lutefisk beschrijft niet waarmee de sequenties vergeleken worden. Aangezien de methode beschreven in [16] betrekking heeft op databasegegevens gaan we er van uit dat ook voor deze tweede scoreprocedure beroep gedaan wordt op databasegegevens, maar hierover is geen zekerheid. We gaan hier dan ook niet verder op in.
Beide scores worden gecombineerd om de uiteindelijke score en ranking van de sequenties te bepalen. Deze sequenties worden gebruikt om in de database-stap op zoek te gaan naar de proteïnen waaruit de sequenties afkomstig zijn.

6.2.7 Pseudo-code

Listing 6.1: Pseudo-code *de novo* algoritme van Lutefisk

```
reduceer de gegevens uit het MS/MS spectrum;

converteer alle ionen naar hun overeenkomstig b-ion;
stel de N-terminale evidence-lijst op;
stel de C-terminale evidence-lijst op;

bereken het sequentie spectrum;
ken bonus probabiliteitsscores toe;

stel de initiële lijst met subsequenties op;
kort de lijst in indien nodig;
do
  for all subsequenties do
    verleng de subsequentie;
  verwijder overtollige subsequenties indien nodig;
until (alle subsequenties hebben de parent massa bereikt)

geef de gevonden sequenties een score;
plaats de gerangschikte sequenties in een file;
```

6.3 CIDentify

CIDentify [14] is gebaseerd op FASTA [17], een homologie-gebaseerde database zoekmethode. FASTA zelf is dan weer een uitbreiding van FASTP [35] [36], een database zoekmethode voor *protein sequence similarity searching*. FASTP werd uitgebreid om ook met andere alfabetten zoals het DNA alfabet te kunnen werken. FASTA staat dan ook voor "FAST-All" (uitpraak : 'FAST-Aye'). Omdat het de *de novo* algoritmen zijn die het onderwerp zijn van deze thesistekst zullen we FASTA en CIDentify slechts kort en in grote lijnen beschrijven. Voor meer details verwijzen we naar de literatuur [17] [14].

6.3.1 FASTA

Kort gezegd neemt FASTA een sequentie als input, in dit geval een peptidesequentie, en doorloopt hiermee een database. Deze sequentie wordt als *tag* gebruikt om te zoeken in welke proteïnen deze *tag* voorkomt. We zullen de peptide sequentie daarom vanaf nu als *tag* benoemen. Voor

iedere database sequentie wordt er een score berekend om de *match* tussen de *tag* en de database sequentie aan te geven[9]. We beschrijven kort de werkwijze van FASTA :

- input = *tag*;
- vergelijk deze *tag* met alle database sequenties;
- geef iedere database sequentie een score op basis van de vergelijking;
- rangschik de gevonden database sequenties volgens hun gekregen score.

De snelheid van FASTA zit in de wijze waarop de vergelijking tussen de *tag* en de database sequenties gebeurt. Een gedetailleerde beschrijving is terug te vinden in [17].
Het voordeel van het opslaan van de massa van een di-peptide en niet alle mogelijke identificaties komt ook hier tot uiting. Alle mogelijke identificaties worden gerepresenteerd door één enkele *tag*, waardoor er ook slechts één *tag* aangeboden moet worden aan *CIDentify* in plaats van alle mogelijke. Met deze ene *tag* kan *CIDentify* namelijk gedurende één database-search alle mogelijke proteïnen vinden die deze *tag* bevatten en op de plaats van de massaweergave een di-peptide hebben die met deze massa overeenkomt. Zouden deze di-peptiden telkens geïdentificeerd worden, dan krijgen we telkens meerdere *tags* en dus evenveel database-searches. Voor meer details verwijzen we weer naar de literatuur.

6.3.2 Van FASTA naar CIDentify

We zullen nu de belangrijkste aanpassingen van FASTA beschrijven die hebben geleid tot CIDentify. In tegenstelling tot FASTA werkt CIDentify niet met één sequentie met neemt alle gevonden sequenties die bij één parent peptide horen als input. De code van het FASTA programma werd daarom aangepast. Om alle sequenties te kunnen doorlopen werd er een extra lus ingebouwd. Iedere sequentie uit de database wordt vergeleken met alle gevonden *tags*. De score voor de database sequentie wordt dan de som van de scores die deze database sequentie krijgt voor de afzonderlijke *tags*.
Zo krijgt iedere database sequentie een score voor de parent peptide. We weten dat de output van een massaspectrometer de gegevens van meerdere parent peptiden bevat. We hebben aanvankelijk een aantal verschillende massaspectra die allemaal deel uitmaken van één *identificatieproces*. Al deze gegevens worden parent peptide na parent peptide of massaspectrum na massaspectrum als input aan Lutefisk aangeboden. Voor iedere parent peptide krijgen de database sequenties dus een aparte score. Indien de parent peptiden afkomstig zijn van een peptidemengsel bekomen door chromatografie, dan kunnen de parent peptiden van verschillende proteïnen afkomstig zijn, in willekeurige volgorde (hoofdstuk 1). Zijn de parent peptiden echter afkomstig van een peptidemengsel dat bekomen werd na gelelectroforese, dan is de kans zeer groot dat ze afkomstig zijn van één bepaalde proteïne. De score van de database sequentie zou voor de proteïne in kwestie dus hoog moeten zijn voor alle aangeboden *tags*, wat een extra gegeven is dat aangewend kan worden voor de identificatie.
In het andere geval kunnen de afzonderlijke resultaten van de parent peptiden ook naast elkaar gelegd worden, maar door het ontbreken van het verband tussen verschillende parent peptiden zal de identificatie toch moeilijker verlopen.

[9] Merk op dat de sequenties in een database volledige *proteïne sequenties* zijn, of op zijn minst *polypeptide sequenties*.

6.3.3 Pseudo-code

Listing 6.2: Pseudo-code CIDentify

```
neem een reeks tags als input;
for all database sequenties do
  for all tags do
    vergelijk de tag met de database sequentie;
    geef de database sequentie een score;
    tel de score op bij de huidige score van de database sequentie;
geef als output de database sequenties gerangschikt volgens score;
```

Hoofdstuk 7

Conclusie

De algoritmen voor de identificatie van peptiden staan nog lang niet op punt. Doorheen de tekst hebben we bijna uitsluitend rekening gehouden met éénwaardig geladen fragmentatie-ionen. Er zou recent een ionisatiemethode ontwikkeld zijn die enkel éénwaardig geladen parent peptiden voortbrengt, wat impliceert dat er ook enkel éénwaardig geladen fragmentatie-ionen kunnen voorkomen. Onze aangenomen beperking zal in de toekomst daarom geen beperking meer zijn maar een standaard worden.
Maar zelfs met éénwaardig geladen ionen blijft het moeilijk om een peptide juist te identificeren. De grote boosdoener is het niet-kennen van de aard van het ion, met name of het om een N-terminaal of een C-terminaal ion gaat. Het algoritme Sherenga lost dit probleem misschien op de minst elegante manier op met het *fake twin vertex* probleem tot gevolg, waardoor er nood is aan ingewikkelde padalgoritmen. Opmerkelijk hierbij is dat deze algoritmen door andere onderzoekers ontwikkeld moesten worden. Dit benadrukt nogmaals de complexiteit van het hele identificatieproces. Bovendien zijn het veelal wetenschappers uit de biologische vakgebieden die bekend zijn in de wereld van de peptiden. De combinatie met niet voor de hand liggende algoritmen loopt ook daarom meestal niet van een leien dakje.
Het algoritme Lutefisk dat ook gebaseerd is op de grafentheorie pakt dit probleem op een heel andere manier aan. Het probleem kan aan de kant geschoven worden door de oorzaak van het probleem aan te pakken in plaats van de gevolgen. Dankzij de conversieformules van Lutefisk kan men immers een graaf-achtige structuur opstellen die niet te kampen heeft met het *fake twin vertex* probleem. Lutefisk heeft dan weer de beperking dat het algoritme, in tegenstelling tot Sherenga dat machine-onafhankelijk tracht te zijn, enkel toegepast kan worden op data bekomen door CID. Deze ionisatie methode wordt veel gebruikt, maar er bestaan nog tal van andere methoden die veelvuldig gebruikt worden. Slechts een beperkt percentage van de beschikbare data kan hierdoor aan Lutefisk aangeboden worden.
NovoHMM benadert de identificatie op een totaal andere manier en baseert zich op de automatentheorie. Bovenstaand probleem tracht men op te lossen door het gebruik van twee Markovkettingen. Het werkende model dat men hiervoor heeft ontwikkeld houdt echter enkel rekening met b- en y-ionen, wat toch een behoorlijke beperking is.
Het tweede grote obstakel waar de algoritmen mee te kampen hebben is de onnauwkeurigheid van de gegevens. Hierin spelen de massaspectrometers een grote rol. De evolutie van deze toestellen zal in grote mate gaan bepalen of de algoritmen met betere resultaten voor de dag kunnen komen. Hoe kleiner de onnauwkeurigheden in de metingen, hoe minder aanpassingen en herberekeningen een algoritme moet uitvoeren, en dus hoe sneller en nauwkeuriger de identificaties zullen verlopen.
Van de drie besproken algoritmen lijkt Lutefisk mij de beste *de novo* methode. De integratie van CIDentify speelt hierbij zeker in het voordeel van het algoritme. Per slot van rekening is het uiteindelijke doel van de peptide-identificatie bijna altijd de identificatie van de proteïnen waaruit de peptide afkomstig is. Bovendien is Lutefisk het enige algoritme van de drie dat een eenvoudige oplossing biedt aan het N-C-terminaal probleem. Het enige nadeel aan de software van Lutefisk is de onmogelijkheid om meerdere files, oftewel meerdere spectra, tegelijk aan te bieden. Hierdoor

is het programma niet echt gebruiksvriendelijk. Het Biomedisch Onderzoekscentrum bijvoorbeeld gebruikt om die reden Lutefisk enkel om resultaten van database zoekprogramma's zoals Mascot [37] en Sequest [16] te verifiëren indien er getwijfeld wordt aan de correctheid ervan. Een automatisatie voor het aanbieden van meerder files na elkaar zou dit euvel van de baan helpen. Lutefisk is gratis ter beschikking op het internet en *open source*.

Het belangrijkste wat we moeten onthouden uit deze studie is dat er nog veel werk is op het gebied van identificatie algoritmen. De grote afhankelijkheid van de kwaliteit van de massaspectrometers, en daardoor ook de meetresultaten, is hierbij jammer genoeg een te bepalende en vooral te beperkende factor. Er is bovendien te weinig controle over de gevormde fragmentatie-ionen. Hierdoor is men al te vaak genoodzaakt uit te gaan van onrealistische vereenvoudigingen. Het is duidelijk dat er nog veel werk te verrichten is in het domein van de bioinformatica, alleen al voor de identificatie van proteïnen.

Deel III

Bijlagen

Bibliografie

[1] João Carlos Setubal and João Meidanis. *Introduction to Computational Molecular Biology.* PWS Publishing Company, 20 Park Plaza Boston, MA 02116, 1997.

[2] New Century College, George Mason University, Fairfax, Virginia: http://www.ncc.gmu.edu/dna/peptid6.gif.

[3] Estrella Moutain Community College, Avondale, Arizona: http://www.emc.maricopa.edu/faculty/farabee/BIOBK/BP2.gif.

[4] Consument, Biotechnologie en Life Sciences: http://www.genomics.nu.

[5] Jean-Paul Noben (Biomedisch onderzoeksinstituut BIOMED). Electroforese : http://users.skynet.be/starr/BiomedElect.pps.

[6] Jean-Paul Noben (Biomedisch onderzoeksinstituut BIOMED). Chromatografie : http://users.skynet.be/starr/BiomedChrom.pps.

[7] Jean-Paul Noben (Biomedisch onderzoeksinstituut BIOMED). Inleidende slides : http://users.skynet.be/starr/Biomed.pps.

[8] Bernd Fischer and Volker Roth and Franz Roos and Jonas Grossmann and Sacha Baginsky and Peter Widmayer and Wilhelm Gruissem and Joachim M. Buhmann. NovoHMM: A Hidden Markov Model for de Novo Peptide Sequencing. *Analytical Chemistry*, 77(22):7265–7273, 15 November 2005.

[9] Vlado Dancik and Theresa A. Addona and Karl R. Clauser and James E. Vath and Pavel A. Pevzner. De Novo Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, 6(3/4):327–342, 1999.

[10] Bingwen Lu and Ting Chen. A Suboptimal Algorithm for *De Novo* Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, 10(1):1–12, 2003.

[11] Nick Wedd: http://www.weddslist.com/ms/tandem.html.

[12] J. Alex Taylor and Richard S. Johnson. Sequence Database Searches via de Novo Peptide Sequencing by Tandem Mass Spectrometry. *Rapid Communications in Mass Spectrometry*, 11:1067–1075, 1997.

[13] Larry Gonick and Mark Wheelis. *The Cartoon Guide to Genetics.* HarperCollins, Updated 1991 edition, 1983.

[14] Paul R. Graves and Timothy A. J. Haystead. Molecular Biologist's Guide tot Proteomics. *Microbiology and Molecular Biology Reviews*, 66(1):39–63, March 2002.

[15] Jean-Paul Noben. Biomedisch Onderzoekscentrum Diepenbeek.

[16] Thermo Electron Corporation: http://www.thermo.com.

[17] Swiss-Prot: http://au.expasy.org/sprot/.

[18] Vineet Bafna and Nathan Edwards. On *de novo* interpretation of tandem mass spectra for peptide identification. In *RECOMB: Research in Computational Molecular Biology*, Berlin, Germany, April 2003.

[19] Rovshan G. Sadygov and Daniel Cociorva and John R. Yates III. Large-scale database searching using tandem mass spectra: Looking up the answer in the back of the book. *Nature Methods*, 1(3):195–202, December 2004.

[20] Ashley L. McCormack and John R. Yates. An approach to correlate tandem mass spectral data of peptides. *Journal of the American Society for mass spectrometry*, 5(11):976–989, 1994.

[21] University of California, San Francisco, Department of Cellular and Molecular Pharmacology: http://www.cmpharm.ucsf.edu/ jmc/pred2ary/formats.html.

[22] Pavel A. Pevzner. *Computational Molecular Biology : An Algorithmic Approach.* The MIT Press, August 2000.

[23] Neil C. Jones and Pavel A. Pevzner. *An Introduction to Bioinformatics Algorithms.* The MIT Press, August 2004.

[24] Steven S. Skiena. *The Algorithm Design Manual.* Springer-Verlag, 1997.

[25] Michael R. Garey and David S. Johnson. *Computers and intractability : A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 41 Madison Avenue, New York, N.Y. 10010 and 20 Beaumont Street, Oxford, England OX1 2NQ, Twenty-fourth Printing 2003 edition, October 1978. Both authors are members of the technical staff of the Mathematics Research Center at Bell Laboratories, Murray Hill, New Jersey.

[26] Harold N. Gabow and Sachindra N. Maheshwari and Leon J. Osterweil. On Two Problems in the Generation of Program Test Paths. *IEEE Trans. Software Engrg*, SE-2:227–231, 1976.

[27] Dan Burns. Algorithms for *De Novo* Peptide Sequencing, Bioinformatics 800, March 11, 2004 : http://www.bioinformatics.med.umich.edu/courses/800/ClassNotes/BI800DeNovo04.ppt.

[28] Ting Chen and Ming-Yang Kao and Matthew Tepel and John Rush and George M. Church. A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry. *Journal of Computational Biology*, 8(3):325–337, 2001.

[29] Bingwen Lu and Ting Chen. Algorithms for de novo peptide sequencing using tandem mass spectrometry. *DDT: BIOSILICO*, 2(2):85–90, March 2004.

[30] Thomas H. Cormen and Charles E. Leierson and Ronald L. Rivest. *Introduction to algorithms.* The MIT Press, Twenty-third printing edition, 1999.

[31] Bernd Fischer and Volker Roth and Joachim M. Buhmann and Jonas Grossmann and Sacha Baginsky and Wilhelm Gruissem and Franz Roos and Peter Widmayer. A Hidden Markov Model for de Novo Peptide Sequencing, 2004.

[32] Charles M. Grinstead and J. Laurie Snell. *Introduction to Probability.* American Mathematical Society, 1997.

[33] M. Bern and D. Goldberg. EigenMS: De Novo Analysis of Peptide Tandem Mass Spectra by Spectral Graph Partitioning. *LNCS*, 3500, 2005.

[34] Christian Bartels. Fast Algorithm for Peptide Sequencing by Mass Spectroscopy. *Biomedical and Environmental Mass Spectrometry*, 19:363–368, 1990.

[35] David J. Lipman and William R. Pearson. Rapid and Sensitive Protein Similarity Searches. *Science*, 227:1435–1441, 22 March 1985.

[36] Eveline Hoekx. Databases in de bioinformatica: een overzicht. Master's thesis, transnationale Universiteit Limburg, 2003.

[37] David N. Perkins and Darryl J.C. Pappin and David M. Creasy and John S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20:3551–3567, 1999.

[38] Vineet Bafna and Nathan Edwards. SCOPE: a probabilistic model for scoring tandem mass spectra against a peptide database. *Bioinformatics*, 17(Suppl. 1):S13–S21, 2001.

[39] Sacha Baginsky and Mark Cieliebak and Wilhelm Gruissem and Torsten Kleffmann and Zsuzsanna Lipták and Matthias Müller and Paolo Penna. AuDeNS: A Tool for Automatic De Novo Peptide Sequencing. Technical Report Technical Report no. 383, ETH Zurich, Department of Computer Science, October 2002.

[40] Klaus Biemann. Appendix 6. Mass Values for Amino Acid Residues in Peptides. *Methods in Enzymology*, 193:888, 1990.

[41] Andreas M. Boehm and Florian Grosse-Coosmann and Albert Sickmann. Command line tool for calculating theoretical ms spectra for given sequences. *Bioinformatics*, 20(16):2889–2891, 1 November 2004.

[42] Tomasz Burzykowski (Center for Statistics Limburgs Universitair Centrum). Hidden Markov Models.

[43] Robertson Craig and Ronald C. Beavis. TANDEM: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, 2004.

[44] Sean R. Eddy. What is a hidden Markov model? *Nature Biotechnology*, 22(10):1315–1316, October 2004.

[45] Lewis Y. Geer and Sanford P. Markey and Jeffrey A. Kowalak and Lukas Wagner and Ming Xu and Dawn M. Maynard and Xiaoyu Yang and Wenyao Shi and Stephen H. Bryant. Open Mass Spectrometry Search Algorithm. *Journal of Proteome Research*, 3:958–964, 2004.

[46] Anders Krogh. *Computational Methods in Molecular Biology (Chapter 4: An Introduction to Hidden Markov Models for Biological Sequences)*. Elsevier, 1998.

[47] Bin Ma and Kaizhong Zhang and Christopher Hendrie and Chengzhi Liang and Ming Li and Amanda Doherty-Kirby and Gilles Lajoie. PEAKS: powerful software for peptide *de novo* sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry*, 17:2337–2342, 2003.

[48] Bin Ma and Kaizhong Zhang and Chengzhi Liang. An Effective Algortihm for the Peptide De Novo Sequencing from MS/MS Spectrum.

[49] Matthias Mann and Ronald C. Hendrickson and Akhilesh Pandey. Analysis of Proteins and Proteomes by Mass Spectrometry. *Annual Review of Biochemistry*, 70:437–473, 2001.

[50] William R. Pearson and David J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 85:2444–2448, April 1988.

[51] J. Alex Taylor and Richard S. Johnson. Implementation and Uses of Automated de Novo Peptide Sequencing by Tandem Mass Spectrometry. *Analytical Chemistry*, 73(11):2594–2604, 1 June 2001.

[52] Shalini Venkataraman and Vidhya Gunaseelan. Hidden Markov Models in Computational Biology.

[53] Yunhu Wan and Austin Yang and Ting Chen. PepHMM: A Hidden Markov Model Based Scoring Function for Mass Spectrometry Database Search. In *RECOMB: Research in Computational Molecular Biology*, pages 342–356, Cambridge, Massachusetts, May 2005.

[54] Michael P. Washburn and Dirk Wolters and John R. Yates III. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nature Biotechnology*, 19:242–247, March 2001.

[55] Bo Yan and Chongle Pan and Victor N. Olman and Robert L. Hettich and Ying Xu. A graph-theoretic approach for the separation of b and y ions in tandem mass spectra. *Bioinformatics*, 21(5):563–574, 2005.

[56] Mascot : http://www.matrixscience.com/.

[57] Sequest : http://fields.scripps.edu/sequest/index.html.

[58] Wikipedia, the free encyclopedia: http://www.wikipedia.org.

Bijlage A

Glossarium

- **adenine (A) :** één van de basen in *DNA*. Samen met *thymine*, *cytosine* en *guanine* wordt deze base in een DNA keten gebonden. Adenine kan *waterstofbruggen* vormen met thymine. In *RNA* wordt adenine gebonden aan *uracil*. Adenine en guanine worden *purinen* genoemd.
- **alpha carbon :** zie C_α.
- **alternatieve splicing :** verandering van genetische informatie na de transcriptie (onderdeel van de *proteïnesynthese*), waardoor er andere proteïnen gevormd worden.
- **aminozuur :** aminozuren zijn de bestanddelen van *proteïnen* of eiwitten. Een aminozuur is een chemische verbinding die zowel een $COOH$ koolstofgroep als een aminogroep NH_2 bezit. Er zijn twintig aminozuren die de bouwstenen van de proteïnen vormen. De aminogroep en de koolstofgroep zitten vast aan hetzelfde koolstofatoom, C_α, zodat de aminozuren allemaal met de algemene formule $R-CH(NH_2)-COOH$ voorgesteld kunnen worden (zie ook figuur 1.3 bovenaan). Zij verschillen onderling in de groep R die de onderscheidende **side chain** voorstelt.
- **antigen :** specifieke structuur van ziekteverwekkers waarmee deze door het afweersysteem herkend kunnen worden.
- **atoom :** een atoom is van alle (scheikundige) elementen de kleinste, nog als zodanig herkenbare, bouwsteen. Ze zijn samengesteld uit drie soorten subatomaire deeltjes: electronen, die een negatieve lading hebben; protonen, die een positieve lading hebben; en neutronen, die geen lading hebben.
- **bp :** base pairs (basis paren). Eenheid van lengte om de lengte van *DNA* molecules te beschrijven.
- **chromosoom :** een lange streng *DNA* waarop vele *genen* een plaats vinden. Een chromosoom bestaat uit twee strengen van DNA moleculen.
- **cytosine (C) :** één van de basen in *DNA*. Samen met *adenine*, *thymine* en *guanine* wordt deze base in een DNA keten gebonden. Cytosine kan *waterstofbruggen* vormen met guanine. Cytosine en guanine worden pyrimidinen genoemd.
- **codon :** opeenvolging van drie *nucleotiden* die één *aminozuur* specificeert.
- C_α**:** centraal koolstof*atoom* in een *aminozuur*.
- **Da :** Dalton, atomaire massa-eenheid, genoemd naar de scheikundige John Dalton, en gelijk aan $1/12^e$ van de massa van één koolstof-12 atoom. Dit is ongeveer $1,6605402 * 10^{-27}$ kilogram.

- **DNA *(deoxyribonucleic acid)*** : desoxyribonucleïnezuur, een voor het leven zeer belangrijke chemische verbinding. Het DNA bevat namelijk de complete erfelijke informatie van het organisme waar het de eigenschappen van omschrijft, zoals van een mens. Het DNA bevindt zich in de kern van iedere lichaamscel, in een aantal afzonderlijke strengen, die *chromosomen* heten.
  Een DNA molecule bestaat uit twee strengen, elk bestaande uit herhalingen van een basiseenheid. Deze basiseenheden verschillen onderling in de base die er aan vasthangt. Dit kunnen de basen *adenine*, *guanine*, *cytosine* of *thymine* zijn. De twee strengen van een DNA molecule houden samen dankzij specifieke bindingsregels (via *waterstofbruggen*) tussen de basen.

- **enzyme :** een *proteïne* dat optreedt als *katalysator*. Een enzyme maakt een biologische reactie in een cel mogelijk of versnelt deze zonder daarbij zelf verbruikt te worden of van samenstelling te veranderen.

- **gen :** de drager van een specifieke erfelijke eigenschap in een cel. Het is een aaneensluitende streng *DNA* die de samenstelling van één *proteïne* codeert.

- **genetische code:** een tabel die aangeeft hoe tripletten van aangrenzende nucleotidenbasen (*codons*) worden vertaald naar *aminozuren* tijdens de *proteïnesynthese*. Er zijn 64 verschillende codons, die coderen voor 20 verschillende aminozuren; voor de meeste aminozuren zijn er dus verschillende codons (tabel zie figuur 1.7).

- **genoom :** de verzameling van alle genen van een organisme. Het genoom beschrijft de combinatie van alle erfelijke factoren.

- **genomica :** de studie van het *genoom* van een organisme en het gebruik van de *genen*. Het behandelt het systematisch gebruik van genoom-informatie, geassocieerd met andere data, om antwoorden te verschaffen in de biologie, geneeskunde en industrie.

- **guanine (G) :** één van de basen in *DNA*. Samen met *adenine*, *thymine* en *cytosine* wordt deze base in een DNA keten gebonden. Guanine kan *waterstofbruggen* vormen met cytosine. Guanine en adenine worden *purinen* genoemd.

- ***hydrolyse* :** een type chemische reactie waarbij een chemische verbinding reageert met een water molecule en daarbij in tweeën gesplitst wordt.

- ***in silico* :** term die wordt gebruikt voor voor computersimulatie.

- ***in vitro* :** letterlijk : in glas. Term die wordt gebruikt voor biologische technieken die buiten het lichaam van het organisme worden toegepast (in een reageerbuis of ander laboratoriumglaswerk).

- **katalysator :** een substantie die chemische reacties versnelt of mogelijk maakt.

- **kDa :** kilo-Dalton, zie ook *Da*.

- **mRNA *(messenger RNA)* :** het 'stukje' molecule dat van het *DNA* afgelezen wordt en naar het *ribosoom* toe gestuurd wordt. Daar wordt het *aminozuur* gevormd.

- **nucleïnezuur :** nucleïnezuren zijn de bouwstenen van *DNA* en *RNA*. Ze coderen de informatie die nodig is om *proteïnen* aan te maken.

- **nucleotide :** een bouwsteen van een *DNA* molecule, bestaande uit:
  - een base : *purine* (*adenine*, *guanine*) of *pyrimidine* (*thymine* of *uracil*, *cytosine*),
  - een suiker : (desoxy)ribose,
  - fosfaatgroepen.

- **peptide :** een deel van een *proteïne* (*in vitro* bekomen).
- **peptidebinding :** binding tussen verschillende aminozuren waardoor een proteïne gevormd wordt.
- **polypeptide-keten :** andere benaming voor een *proteïne* omwille van haar structuur die bestaat uit een aaneenschakeling van aminozuren, verbonden door *peptidebindingen*.
- **posttranslationele wijziging :** verandering in massa en of lading van aaneengeschakelde aminozuren tijdens de proteïnesynthese, na de translatie, door het aantrekken van andere moleculen zoals water.
- **proteïne :** een proteïne of eiwit is een chemische verbinding bestaande uit een keten van *aminozuren*, verbonden door *peptidebindingen*, die in een organisme een bepaalde functie vervult. Typische proteïnen zijn enige tientallen tot vele honderden aminozuren lang. Proteïnen zijn de moleculen die het leven mogelijk maken.
- **proteïnesynthese :** proces waarbij proteïnen aangemaakt worden in een organisme.
- **proteomica :** de studie van het *proteoom*.
- **proteoom :** de verzameling van *proteïnen* die gevonden wordt in een bepaald celtype onder een bepaalde set omgevingsvoorwaarden.
- **purine :** een organische base (die bestaat uit twee organische ringen die stikstof bevatten). De in *DNA* en *RNA* voorkomende purines zijn *adenine* en *guanine*.
- **pyrimidine :** een organische base (die bestaat uit één organische ring die stikstof bevat). De in *DNA* en *RNA* voorkomende pyrimidines zijn: *thymine*, *uracil* en *cytosine*.
- **ribosoom :** celstructuur waarin een deel van de *proteïnesynthese* plaatsvindt.
- **RNA *(ribonucleic acid)* :** ribonucleïnezuur, bestaat uit een reeks van *nucleotiden*. RNA speelt een belangrijke rol in de aanmaak van *proteïnen* in de cel. Het RNA-systeem verschilt van *DNA*, o.a. doordat voor de nucleotiden gebruik gemaakt wordt van *uracil* in plaats van *thymine*. Een ander verschil is dat RNA bijna alleen maar in single-stranded (enkelstrengige) vorm wordt aangetroffen.
  Er zijn drie soorten RNA :
  - *mRNA (messenger RNA)*
  - *rRNA (ribosomaal RNA)*
  - *tRNA (transport RNA)*
- **rRNA (ribosomaal *RNA*) :** maakt de eigenlijke *proteïne* aan de hand van de informatie van *mRNA*. rRNA kan iedere proteïne maken als het via het mRNA wordt aangestuurd.
- **thymine (T) :** één van de basen in *DNA*. Samen met *adenine*, *cytosine* en *guanine* wordt deze base in een DNA keten gebonden. Thymine kan *waterstofbruggen* vormen met adenine. Thymine en cytosine worden *pyrimidinen* genoemd.
- **tRNA (transport *RNA*) :** dit soort RNA brengt de *aminozuren* over naar het *ribosoom*, waar de *proteïnesynthese* plaatsvindt.
- **trypsine :** een *enzyme* met een specifieke knipeigenschap voor *proteïnen*. Het splitst enkel *peptidebindingen* waarvan de koolstofgroep afkomstig is van de *aminozuren* K of R.
- **uracil (U) :** één van de vier basen waarmee in *RNA* genetische informatie gecodeerd is. Ten opzichte van *DNA* neemt het de plaats in van *thymine*. Uracil kan *waterstofbruggen* vormen met *adenine*. Uracil behoort tot de *pyrimidines*.

- **waterstofbrug :** een term om de wisselwerking aan te duiden die kan plaatsvinden tussen twee watermoleculen. Een waterstofbrug loopt altijd van het waterstofatoom (H) in het ene molecule naar het zuurstofatoom (O) van het andere molecule. In deze thesistekst zijn deze moleculen de basen A, G, C en T. De waterstofbruggen tussen deze basen zorgen ervoor dat de twee strengen van DNA bij elkaar blijven.

Bijlage B

Voorbeeld data file : output MS/MS

Hieronder wordt een voorbeeld gegeven van een data file die de gegevens bevat zoals ze uit de tandem massaspectrometer komen die in het Biomedisch Onderzoeksinstituut gebruikt wordt (campus Diepenbeek). Om de leesbaarheid te verhogen zijn er aan de originele gegevens spaties en witregels toegevoegd. We zien nu duidelijk dat er drie parent ionen geselecteerd werden. De eerste, vet gedrukte regel van iedere 'blok' gegevens beschrijft de massa van het parent ion en de lading ervan. De volgende regels bevatten de m/z waarden van de fragmentatie ionen en hun intensiteiten. Voor het derde parent ion zijn niet alle gegevens weergegeven. Onderstaande data file bevat dus eigenlijk drie massaspectra.

1332.87	**2**
194.5	1350.0
212.7	1717.0
229.7	1722.0
233.6	1848.0
242.0	722.0
254.1	3431.0
257.1	2410.0
275.1	1796.0
280.9	801.0
283.2	1737.0
287.4	2181.0
288.1	1188.0
297.2	2576.0
302.9	1692.0
304.3	15046.0

304.9	**1.0**
314.3	1395.0
323.2	7980.0
324.2	578.0
324.9	1748.0
340.2	16802.0
341.1	3161.0
342.1	3611.0
343.3	686.0
351.1	1482.0
352.9	11657.0
358.2	9237.0

358.9	**1.0**
375.5	18739.0
376.4	2047.0
394.2	2214.0
408.5	780.0
410.3	469.0
412.5	534.0
418.0	1212.0
421.3	1480.0
426.5	1306.0
429.3	2765.0
436.2	1253.0
451.1	8422.0
451.8	4760.0
453.4	4885.0
465.9	4728.0
467.1	2876.0
468.1	3775.0
469.2	7351.0
472.5	1721.0
473.4	978.0
477.6	1778.0
483.7	1346.0
486.4	17867.0
487.4	2292.0
491.2	2119.0
503.3	954.0
504.5	30954.0
505.3	5597.0
506.6	1096.0
513.8	1139.0
520.2	2216.0
…	

Bijlage C

Massa's van aminozuren residu's

Onderstaande tabel geeft de massa's weer in Dalton van de residu's van de twintig aminozuren die voorkomen bij proteïnen [40]. De massa's zijn afgerond tot op 1/100 Dalton.

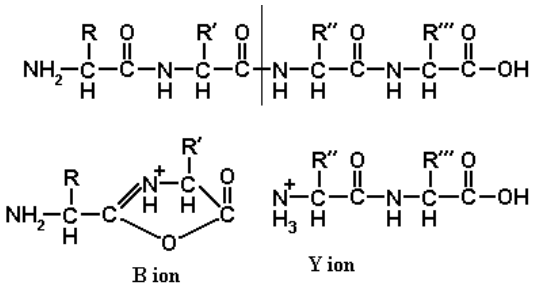
One-letter code	Three-letter code	Name	Mass (Da)
A	Ala	Alanine	71.04
C	Cys	Cysteine	103.01
D	Asp	Aspartic Acid	115.03
E	Glu	Glutamic Acid	129.04
F	Phe	Phenylalanine	147.07
G	Gly	Glycine	57.02
H	His	Histidine	137.06
I	Ile	Isoleucine	113.08
K	Lys	Lysine	128.09
L	Leu	Leucine	113.08
M	Met	Methionine	131.04
N	Asn	Asparagine	114.04
P	Pro	Proline	97.05
Q	Gln	Glutamine	128.06
R	Arg	Arginine	156.10
S	Ser	Serine	87.03
T	Thr	Threonine	101.05
V	Val	Valine	99.07
W	Trp	Tryptophan	186.08
Y	Tyr	Tyrosine	163.06

Tabel C.1: Aminozurenmassa's

Bijlage D

Massa-berekening van fragmentatie ionen

Om de fragmentatie van een parent ion in b- en y-ionen te begrijpen is wat extra kennis nodig over de structuur van atomen, met name over protonen, electronen en neutronen. Atomen hebben een kern bestaande uit protonen en neutronen. Rond deze kern 'zweven', bij een neutraal atoom, net evenveel electronen als er protonen in de kern aanwezig zijn. Een proton is een deeltje met een positieve lading $+1$ en een massa van 1 Da. Een electron heeft een negatieve lading van -1 maar heeft, in tegenstelling tot een proton, een verwaarloosbare massa. Een neutron heeft, net zoals een proton een massa van 1 Dalton, maar heeft geen lading. Het zijn de neutronen en de protonen die de massa van een atoom bepalen. De electronen daarentegen kunnen zorgen voor een positief (er zijn minder electronen dan protonen) of een negatief (er zijn meer electronen dan protonen) geladen atoom.
Een uitzondering bij de atomen is het waterstofatoom H dat *geen neutronen* bevat; de kern van een waterstofatoom bevat enkel één proton. Wordt de enige electron van dit atoom weggehaald, dan resulteert dit in het H^+ atoom, oftewel één enkele proton. Een proton is dus niets meer dan een waterstofatoom zonder zijn electron.
Met dit in het achterhoofd komen we nu tot de fragmentatie van een eenwaardig geladen parent ion in b- en y-ionen. Figuur D.1 illustreert het proces. Voor het b-ion zien we dat er geen atomen bijgekomen zijn, maar wel dat het hele fragmentatie ion positief geladen is (de '+' ter hoogte van N). De eenwaardige positieve lading op het b-ion wordt bekomen doordat er na de fragmentatie een electron 'tekort' is. Dit electron bindt zich met de proton die zorgde voor de lading van het parent ion, en vormt zo een doodgewoon waterstofatoom dat verloren gaat. Er is geen verandering in massa tussen het al dan niet geladen N-terminaal fragment.
Bij het vormen van een y-ion voegt de proton zich bij het C-terminale fragmentatie-ion. Bovendien trekt dit ion nog een volwaardige H-molecule aan. De fragmentatie vermeerdert de massa van het gefragmenteerde deel dus met 2 Da.



Figuur D.1: Fragmentatie van een parent ion in een b- en een y-ion

Bijlage E

Theoretisch spectrum voor de peptide HLITFSR

Alle gegevens van het theoretisch spectrum voor de peptide HLITFSR. De intensiteitswaarden zijn gelijk aan elkaar en daarom weggelaten.

855.47	1
138.06	
175.10	
251.14	
262.13	
364.22	
409.20	
465.27	
510.25	
612.34	
623.33	
699.37	
736.41	

Auteursrechterlijke overeenkomst

Opdat de Universiteit Hasselt uw eindverhandeling wereldwijd kan reproduceren, vertalen en distribueren is uw akkoord voor deze overeenkomst noodzakelijk. Gelieve de tijd te nemen om deze overeenkomst door te nemen en uw akkoord te verlenen.

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Algoritmen voor identificatie van peptiden in massaspectrometrie
Richting: **Licentiaat in de informatica** Jaar: **2006**
in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Deze toekenning van het auteursrecht aan de Universiteit Hasselt houdt in dat ik/wij als auteur de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij kan reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

U bevestigt dat de eindverhandeling uw origineel werk is, en dat u het recht heeft om de rechten te verlenen die in deze overeenkomst worden beschreven. U verklaart tevens dat de eindverhandeling, naar uw weten, het auteursrecht van anderen niet overtreedt.

U verklaart tevens dat u voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen hebt verkregen zodat u deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal u als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze licentie

Ik ga akkoord,

Suzanna CHA

Datum: