KU LEUVEN

FACULTEIT BIO-INGENIEURSWETENSCHAPPEN

Netwerk-gebaseerde analyse van reactiewegen betrokken in *Salmonella* biofilmvorming

Network based analysis of pathways involved in *Salmonella* biofilm formation

Promotor:
Prof. Kathleen Marchal
Departement Microbiële en Moleculaire Systemen
Centrum voor Microbiële en Plantengenetica

Masterproef voorgedragen
tot het behalen van het diploma van
Master of science in de bio-ingenieurswetenschappen:
cel- en gentechnologie

Nele Cosemans

juni 2014

Dankwoord

Bij deze wil ik graag alle mensen bedanken die me hielpen dit werk te realiseren. Om te beginnen mijn promotor Prof. Kathleen Marchal om mij dit geweldige onderwerp in Leuven aan te bieden. Ten tweede mijn begeleider Dries De Maeyer die me door het doolhof van nieuwe informatie leidde en van wie ik heel veel heb bijgeleerd. Ook aan de rest van de onderzoeksgroep zal ik zeker fijne herinneringen overhouden, in het bijzonder aan Qiang Fu en Bram Weytjens bij wie ik altijd terecht kon. Zowel voor mijn computerproblemen als voor een toffe babbel. Verder bedank ik dr. ir. Hans Steenackers voor het doorgeven van data en de bijhorende uitleg. En hoewel ik als bio-informaticus nooit in de labo's kwam, werd ik ook door de rest van het CMPG-team met open armen ontvangen, waarvoor dank.

Daarbuiten zijn er nog veel mensen aan wie ik dank verschuldigd ben. Vrienden en vriendinnen van bio-ingenieur, jullie verklaarden me allemaal zot met mijn keuze voor bio-informatica, maar boden toch enorm veel steun als bijvoorbeeld mijn laptop weer eens niet deed wat ik wou. Kotgenoten en vrienden van de harmonie, jullie boden altijd een luisterend oor en zorgden voor ontspannende momenten tussen het thesissen. Bedankt!

Tenslotte gaat mijn dank uit naar mijn vriend Jasper, mijn broer Pieter, mijn zus Hanne en mijn ouders. Ze hebben mij altijd enorm gesteund en waren steeds geïnteresseerd in de vooruitgang van mijn werk. Jasper, je LateX-template en de vele lieve berichtjes in de loop van dit thesisjaar waren een echte boost. Hanne, papa en mama, enorm bedankt voor de hulp bij het nalezen van mijn thesis.

Samenvatting

Hoge-doorvoerexperimenten laten toe lijsten van genen te genereren die worden geassocieerd met een te onderzoeken fenotype. De moeilijkheid bij dit type experimenten is het verkrijgen van een goed biologisch inzicht in deze genenlijsten. Een mogelijke manier om dergelijk inzicht te bekomen is de analyse van deze genenlijsten in het licht van bestaande publieke data. Deze studie maakt hiervoor gebruik van het subnetwerkselectie-algoritme PheNetic. Aan de hand van een interactienetwerk, opgesteld uit publiek beschikbare interactoomdata, selecteert deze methode een subnetwerk dat het mechanisme achter het onderzochte fenotype voorspelt. Dit leidt tot een snel verkregen inzicht in het fenotype dat onderzocht wordt en identificeert mogelijke aanwijzingen voor verder onderzoek.

De verschillende expressiedatasets gebruikt in deze studie laten toe de mogelijkheden van deze methode te toetsen en te illustreren. Specifiek wordt een analyse gedaan van de werking van imidazool, een anti-biofilm agens, en mitomycine C, een DNA cross-linker, in het organisme *Salmonella typhimurium* LT2. De studie start met een karakterisering van verschillende instellingen voor PheNetic aan de hand van een technische en een biologische validatie. Vertrekkende uit deze resultaten worden vervolgens experimentele datasets geanalyseerd die de werking van een imidazoolbehandeling en de bijhorende resistentiemechanismen in natuurlijke stammen bestuderen. Dit resulteert enerzijds in de identificatie van mogelijke regulatoren die een rol spelen in *Salmonella* biofilmvorming en anderzijds leidt dit tot inzichten in biologische functies die mogelijks beïnvloed worden door een imidazoolbehandeling.

Meer specifiek werd de mogelijke rol van regulatoren als *csg*D, *pho*P en *rpo*S in *Salmonella* biofilmvorming bevestigd. Bijkomend identificeerden de resulterende subnetwerken nieuwe potentiële regulatoren, namelijk *cys*B en *ycf*Q. Op functioneel vlak werken imidazoolbehandelingen dan weer in op essentiële processen zoals celbeweging en lipidenmetabolisme. De bekomen resultaten waren een rechtstreekse aanleiding voor bijkomend 'wet-lab' onderzoek wat de toepasbaarheid en mogelijkheden van PheNetic onderstreept.

Abstract

High-throughput experiments create the possibility to generate gene lists associated with a phenotype under investigation. However, the gathering of good biological insights for these gene lists encounters difficulties with this type of experiments. One possible way to achieve such understanding is to analyze these gene lists in light of existing public knowledge. Therefore, this study investigates the subnetwork selection method PheNetic. This algorithm selects, based on an interaction network compiled from public interactomics data, a subnetwork predicting the mechanism behind the studied phenotype. As a result, PheNetic obtains a quick understanding of the mechanism leading to this phenotype and identifies possible indications for further investigation.

This work tests and illustrates the possibilities of this method based on different expression data sets. In particular the effects of imidazole, an anti-biofilm agent, and mitomycin C, a DNA cross-linker, are analyzed in the organism *Salmonella typhimurium* LT2. The study first characterizes the different settings for PheNetic in a technical en biological validation. Starting from these results, follows the analysis of experimental data investigating the mechanism of imidazole treatments and the associated resistance in natural strains. This results in the identification of potential regulators involved in biofilm formation on the one hand and insights in the biological functions potentially affected by imidazole treatment on the other hand.

Specifically, the possible involvement of the regulators *csg*D, *pho*P and *rpo*S in biofilm formation was confirmed and the resulting subnetworks also identified new potential regulators like *cys*B and *ycf*Q. Besides, imidazole treatments functionally effect essential processes like cell movement and lipid metabolism. These obtained results gave immediately rise to additional wet lab research pointing out the applicability and the potential of PheNetic.

Lijst met afkortingen

C_b	'betweenness' centraliteitsparameter
C_d	graadscentraliteitsparameter
C_u	clusteringscoëfficiënt
Pr_{belief}	probabiliteit die het geloof in de werkelijke interactie tussen 2 knooppunten aangeeft
$Pr_{expressie}$	probabiliteit die de DE-waarden van het betreffende experiment in rekening brengt
Pr_{hub}	probabiliteit die rekening houdt met de 'hubiness' van een knooppunt
c	de kost voor de scoreberekening
COLOMBOS	COLlection Of Microarrays for Bacterial OrganismS
DE-waarden	differentiële expressiewaarden
e	genomen expressiefractie van de oorspronkelijke dataset
f	de selectiefrequentie van de interacties in de oplossing
KEGG	'Kyoto Encyclopedia of Genes and Genomes'
l	padlengte
mRNA	boodschapper ribonucleïnezuur
n	het aantal beste resultaten per genenpaar te selecteren
r	het aantal optimalisatieronden
sRNA	klein ribonucleïnezuur
STRING	'Search Tool for Recurring Instances of Neighbouring Genes'

Lijst van tabellen

Lijst van figuren

Inhoudsopgave

Context en doelstellingen

Huidige 'wet-lab' experimenten maken intensief gebruik van hoge-doorvoermethoden om inzicht te krijgen in specifieke fenotypes en biologische processen. Het interpreteren en analyseren van de resultaten uit deze experimenten is echter niet vanzelfsprekend. Dit door de grote hoeveelheid data die deze experimenten genereren en de inherente biologische en statistische ruis die aanwezig is in de bekomen resultaten. Methoden die toelaten deze resultaten te bekijken in het licht van de steeds groter wordende publieke kennis kunnen een oplossing vormen voor deze problemen. Biologische netwerken bundelen deze publieke data en kunnen in combinatie met *in silico* methoden gebruikt worden om inzichten te verwerven in de bekomen hoge-doorvoerresultaten. Dit werk bestudeert het subnetwerkselectie-algoritme PheNetic (hoofdstuk 5): een beslissingstheoretisch algoritme dat toelaat om vanuit genenlijsten, bekomen uit resultaten van hoge-doorvoerexperimenten, relevante subnetwerken te selecteren in deze biologische netwerken.

Meer bepaald onderzoekt deze studie de mogelijkheden om expressiedata te analyseren met PheNetic in *Salmonella*. Een eerste deel bestudeert de invloed van verschillende parameters die PheNetic gebruikt, op de geselecteerde subnetwerken (hoofdstuk 7) om een beter inzicht te krijgen in de toepasbaarheid van dit algoritme. Een tweede deel bestaat uit een biologische validatie van de geselecteerde subnetwerken aan de hand van een gekend biologisch proces (hoofdstuk 8). In een laatste deel worden de resultaten weergeven van een gevalstudie die de werking van imidazolen op de vorming van *Salmonella* biofilms probeert te ontrafelen (hoofdstuk 9).

Om dit alles beter te plaatsen, geeft de literatuurstudie een inleiding over biologische netwerken en hun constructie (hoofdstuk 1), computationele manieren om netwerken te doorzoeken (hoofdstuk 2) en een overzicht van de huidige stand van het onderzoek rond *in silico* analysemethoden voor subnetwerkselectie (hoofdstuk 3).

Deel I

Literatuurstudie

Hoofdstuk 1

Biologische netwerken

'High-throughput' technieken maken het mogelijk grote hoeveelheden genoom-wijde interactiedata tussen moleculaire entiteiten – zoals genen, eiwitten, metabolieten, ... – te verwerven. Netwerkvoorstellingen van dergelijke data laten toe deze informatie op een éénduidige manier te bundelen (Alm & Arkin, 2003; Alon, 2003; Barabási & Oltvai, 2004; Joyce & Palsson, 2006; Emmert-Streib & Dehmer, 2011; Cloots *et al.*, 2014). Dit hoofdstuk geeft een overzicht van de biologische netwerken die kunnen afgeleid worden uit deze data (sectie 1.1), de eigenschappen van deze netwerken (sectie 1.2) en hoe ze opgesteld kunnen worden (sectie 1.3).

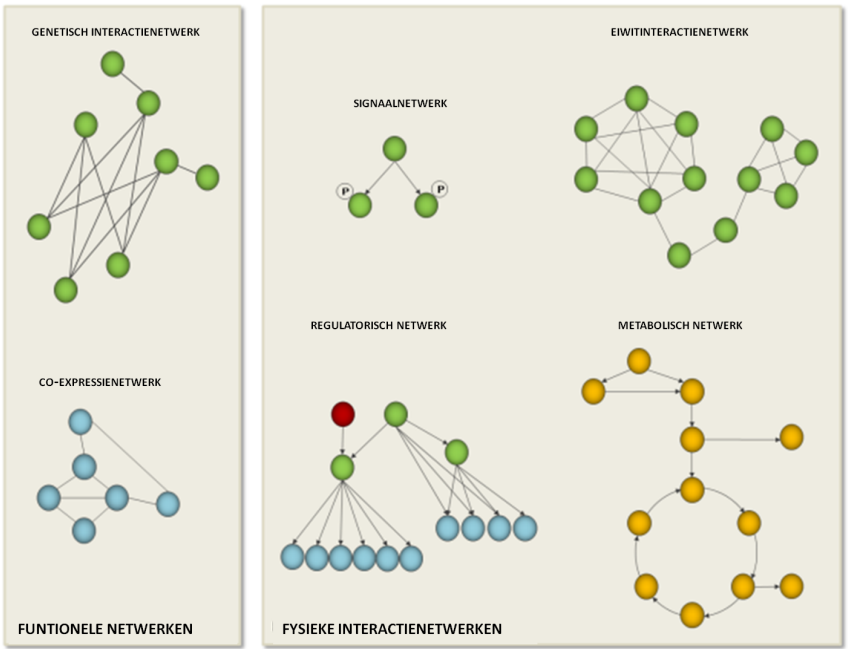
1.1 Voorstelling in biologische netwerken

Netwerken laten toe biologische systemen abstract voor te stellen (Alon, 2003) als een diagram van knooppunten en verbindingslijnen (Alm & Arkin, 2003; Yeang *et al.*, 2004). De knooppunten ('nodes') in deze biologische netwerken stellen – afhankelijk van het soort netwerk – genen, eiwitten of metabolieten voor. De verbindingslijnen ('edges') geven de relaties tussen de knooppunten weer, hetgeen directe fysieke bindingen of (indirecte) functionele interacties kunnen zijn (Barabási & Oltvai, 2004; Yeang *et al.*, 2004; Joyce & Palsson, 2006; Yeger-Lotem *et al.*, 2009). Enkele voorbeelden van biologische netwerken worden voorgesteld in figuur 1.1.

1.1.1 Functionele netwerken

Functionele netwerken geven een voorstelling van eender welke link die twee genen, of hun genproducten, kunnen vormen (Snel *et al.*, 2000). Ze zijn gebaseerd op alle beschikbare info uit zowel genetische, biochemische en computationele experimenten (Lee *et al.*, 2004, 2010; Joyce & Palsson, 2006). Klassieke voorbeelden van functionele netwerken zijn co-expressienetwerken en genetische interactienetwerken (figuur 1.1) (Cloots & Marchal, 2011).

Co-expressienetwerken zijn functionele netwerken waarin de knooppunten genen (of hun eiwitproducten) voorstellen en de verbindingslijnen significante co-expressie tussen deze genen in bepaalde omgevingscondities aangeven (Zhang & Horvarth, 2005; Zhang *et al.*, 2012; Gibbs *et al.*, 2013). Ze kunnen opgebouwd worden aan de hand van expressiecompendia (Cloots & Marchal, 2011).



Figuur 1.1. Voorstelling van enkele biologische netwerken. (links) Functionele netwerken: moleculaire entiteiten, voorgesteld door knooppunten in het netwerk, delen een functionele relatie die geen fysiek contact vereist. Genetisch interactienetwerk: verbindingslijnen geven het fenotype weer van de simultane inactivatie (dubbel-mutanten) van de verbonden knooppunten (genen). Co-expressienetwerk: verbindingslijnen verbinden knooppunten (genen) die samen voorkomen in expressieprofielen. (rechts) Fysieke interactienetwerken: de knooppunten in het netwerk interageren via fysiek contact. Signaalnetwerk: de knooppunten stellen eiwitten voor en de verbindingslijnen geven de signalisatie (bijvoorbeeld fosforylaties) ertussen aan. Eiwitinteractienetwerk: verbindingslijnen geven de fysieke interacties tussen de knooppunten die eiwitten voorstellen, weer. Regulatorisch netwerk: knooppunten stellen regulatoren (sRNA, transcriptiefactoren) of hun doelwitgenen voor, verbindingslijnen de regulator-doelwitinteracties. Metabolisch netwerk: verbindingslijnen presenteren metabolische interacties tussen knooppunten (enzymen) (Cloots *et al.*, 2014).

Een genetisch interactienetwerk beschrijft hoe verschillende genen elkaar reguleren en hoe omgevingscondities genexpressie beïnvloeden om bepaalde fenotypische eigenschappen te bekomen (Wessels *et al.*, 2001). De knooppunten in het netwerk stellen de genen voor en de verbindingslijnen de interacties tussen de genen. Deze functionele interacties kunnen geïdentificeerd worden door onder andere een 'synthetic gene array', een methode die de observatie van defecten in het fenotype van dubbelmutanten mogelijk maakt (Tong *et al.*, 2001).

Grote inspanningen voor het standaardiseren en samenvoegen van talrijke functionele interacties resulteerden in de 'Search Tool for Recurring Instances of Neighbouring Genes' (STRING) (Snel *et al.*, 2000; von Mering *et al.*, 2003, 2007; Jensen *et al.*, 2009; Franceschini *et al.*, 2013). Deze databank bevat onder andere functionele interacties gebaseerd op het gelijk voorkomen in publicaties, orthologie – een gemeenschappelijke functie door een gemeenschappelijke voorouder (Fitch, 1970) – en co-expressie.

1.1.2 Fysieke interactienetwerken

Fysieke interactienetwerken geven een feitelijke representatie van het interactoom: enkel fysieke bindingen zoals eiwit-eiwit- en eiwit-DNA-interacties worden hierin opgenomen (Ideker *et al.*, 2002; Yeang *et al.*, 2004; Lan *et al.*, 2011; Cloots *et al.*, 2014). Deze netwerken bundelen verschillende interactielagen waarvan deze sectie er enkele bespreekt, namelijk eiwitinteractienetwerken, regulatorische netwerken, metabolische netwerken en signaalnetwerken (figuur 1.1) (Cloots & Marchal, 2011; Cloots *et al.*, 2014). De interactienetwerken beschreven in het vervolg van deze studie zijn biologische fysieke interactienetwerken, tenzij anders gespecificeerd.

1.1.2.1 Eiwitinteractienetwerken

In eiwitinteractienetwerken stellen de knooppunten eiwitten voor en de verbindingslijnen interacties tussen de eiwitten (von Mering *et al.*, 2002; Peregrín-Alvarez *et al.*, 2009; Bonetta, 2010; Sardiu & Washburn, 2011). De verbindingslijnen zijn ongericht, d.i. ze bevatten geen richting van het ene naar het andere eiwit, omdat deze niet gekend is (Yeang *et al.*, 2004; Yeger-Lotem *et al.*, 2009; De Smet & Marchal, 2010). Deze interactienetwerken worden gebruikt om de samenwerking van eiwitten in cellulaire biologische processen af te leiden door clusters te identificeren (figuur 1.1) (Yeger-Lotem *et al.*, 2004; Pavlopoulos *et al.*, 2011). Yeast-to-Hybrid en massaspectrometrie-gebaseerde technieken zijn voorbeelden van experimentele technieken die de bepaling van eiwit-eiwitinteracties mogelijk maken (Alm & Arkin, 2003; Bonetta, 2010; Cloots & Marchal, 2011; Sardiu & Washburn, 2011) en verschillende databanken (sectie 1.3.3) integreren deze informatie.

In het kader van dit werk worden de eiwitinteractienetwerken opgesteld door Peregrín-Alvarez *et al.* (2009) en STRING gebruikt. Peregrín-Alvarez *et al.* (2009) ontwikkelden een eiwitinteractienetwerk specifiek voor *Escherichia coli*, het modelorganisme voor bacteriën. De vele experimentele data in STRING maken het mogelijk om zowel functionele als fysieke netwerken af te leiden gebaseerd op gecureerde kennis en computationeel voorspelde interacties (Snel *et al.*, 2000; von Mering *et al.*, 2003; Jensen *et al.*, 2009; Franceschini *et al.*, 2013).

1.1.2.2 Regulatorische netwerken

De knooppunten in regulatorische netwerken kunnen zowel regulatoren – zijnde transcriptiefactoren of sRNA's (kleine ribonucleïnezuren) – als de genen waarvan ze de expressie reguleren, voorstellen. De aard van de interacties hiertussen – de regulator migreert naar het gen – leidt tot gerichte verbindingslijnen waardoor het netwerk een hiërarchische structuur bevat (figuur 1.1) (Lee *et al.*, 2002; Babu *et al.*, 2004; Scott *et al.*, 2005; De Smet & Marchal, 2010; Beisel & Storz, 2010; Cloots *et al.*,

2014). Regulatorische netwerken leveren inzicht in de expressieregulatie in cellen: zowel transcriptiefactoren, post-transcriptionele modificaties als interacties met andere biomoleculen spelen een rol in dit proces (Pavlopoulos *et al.*, 2011).

Transcriptiefactoren binden op genen en deze eiwit-DNA-interacties kunnen bijvoorbeeld afgeleid worden uit experimenten met chromatine-immunoprecipitatie (Babu *et al.*, 2004; Scott *et al.*, 2005; Cloots & Marchal, 2011). sRNA's reguleren genexpressie door baseparing met het mRNA (boodschapper ribonucleïnezuur) van de overeenkomstige genen, wat leidt tot veranderingen in mRNA-stabiliteit en -translatie. Deze doelwitgenen kunnen bijvoorbeeld geïdentificeerd worden met microarray-analyses na overexpressie van de sRNA's (Beisel & Storz, 2010; Licatalosi & Darnell, 2010). Deze studie gebruikt de databank RegulonDB als informatiebron voor dit soort interacties in *E. coli* (Huerta *et al.*, 1998; Salgado *et al.*, 2013).

1.1.2.3 Metabolische netwerken

Een metabolisch netwerk bestaat uit knooppunten die de metabolieten in chemische reacties voorstellen. De verbindingslijnen geven de omzetting van de metabolieten, wat zich in het netwerk weerspiegelt als een cascade van interacties (figuur 1.1) (Jeong *et al.*, 2000; Schuster *et al.*, 2000; Guimerà & Amaral, 2005). Indien de reactie onomkeerbaar is, zijn de verbindingslijnen gericht van het reagens naar het product. Er is veel informatie beschikbaar uit studies van individuele enzymen, maar de aanwezige 'feedback'-lussen in deze interactienetwerken maken het geheel complex (Alm & Arkin, 2003). Deze informatie is onder andere terug te vinden in databanken als Biocyc (Caspi *et al.*, 2012), 'BRaunschweig ENzyme DAtabase' (BRENDA) (Schomburg *et al.*, 2002) en 'Kyoto Encyclopedia of Genes and Genomes' (KEGG), die gecureerde genoomannotaties en biochemische reacties bevatten (Kanehisa & Goto, 2000; Cloots & Marchal, 2011).

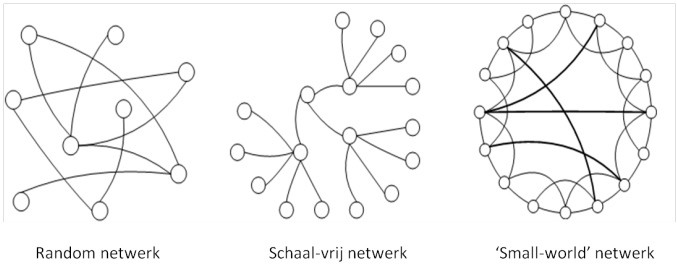
1.1.2.4 Signaalnetwerken

In signaalnetwerken staan de knooppunten voor kinasen of substraten en geven de gerichte verbindingslijnen de fosforylatie van substraten door kinases aan (figuur 1.1). Naast de regulatie door eiwit-eiwitinteracties, de regulatie van enzymen en de productie van secundaire boodschappers, is signaaloverdracht immers ook mogelijk door de fosforylatie van eiwitten (Bhalla & Iyengar, 1999; Olsen *et al.*, 2006; Macek *et al.*, 2008). En hoewel signaaloverdracht via fosforylatie sterk bestudeerd is in eukaryoten, bezitten prokaryoten ook een beperkt aantal fosforylaties die waarschijnlijk cruciaal zijn (Macek *et al.*, 2008). De benodigde informatie kan bekomen worden uit onder andere eiwit-chiparrays, massaspectrometrie (Olsen *et al.*, 2006; Cloots & Marchal, 2011) en sequentie-analyses, zoals opgenomen in de 'Phosphorylation site database' (PHOSIDA) van Gnad

et al. (2011), die fosforylatieplaatsen kunnen identificeren op basis van lineaire sequentiepatronen. Volledig beschreven fosforylatiedata – waar ook de componenten die de fosforylaties uitvoeren, gekend zijn – zijn echter weinig beschikbaar, waardoor deze tot op heden beperkt aanwezig zijn in databanken (Karp *et al.*, 2002; von Mering *et al.*, 2003), onder andere in STRING en KEGG.

1.2 Eigenschappen van biologische netwerken

Om biologische netwerken beter te begrijpen beschrijft deze sectie bepaalde eigenschappen van deze netwerken – namelijk de (bij benadering) schaal-vrije graadverdeling, de korte karakteristieke padlengten ('small world property') en de hoge clusteringscoëfficiënt (C_u) (Alm & Arkin, 2003; Mason & Verwoerd, 2008; Pavlopoulos *et al.*, 2011) – met behulp van figuur 1.2. Hoewel het soms moeilijk is deze grafische eigenschappen statistisch te valideren (Van Helden *et al.*, 2000; Khanin & Wit, 2006; Daudin *et al.*, 2008; Lima-Mendez & van Helden, 2009), worden deze toch aangenomen in de wetenschappelijke consensus.



Figuur 1.2. Een vergelijking van de verschillende soorten netwerken door Huang *et al.* (2005): (links) random netwerk waarin knooppunten willekeurig verbonden zijn en gemiddeld dezelfde graad bezitten, (midden) een schaal-vrij netwerk waarin enkele sterk verbonden knooppunten de rest verbinden en (rechts) een 'small-world' netwerk waarin de gemiddelde padlengte klein is in vergelijking met het aantal knooppunten.

1.2.1 Schaal-vrije graadverdeling

De term schaal-vrij verwijst naar de afwezigheid van een standaard knooppunt, waarmee alle andere knooppunten gekarakteriseerd kunnen worden (Barabási & Oltvai, 2004). De graadverdeling geeft de kans op een bepaalde graad – dit is het aantal verbindingslijnen die aankomen en/of vertrekken in een knooppunt – weer voor een netwerk. Deze volgt voor schaal-vrije netwerken een exponentiële verdeling: de kans op een bepaalde graad voor een knooppunt kan benaderd worden door $P(k) \sim k^{-\gamma}$ met de graadexponent $\gamma > 1$. Bijgevolg bezitten de meeste knooppunten geen gemiddelde graad (Mason & Verwoerd, 2008), zoals in random netwerkmodellen, en zijn er een eindig aantal sterk verbonden knooppunten, ook wel 'hubs' genoemd, in het systeem aanwezig (Pavlopoulos *et al.*, 2011). De genen of eiwitproducten voorgesteld door deze knooppunten zijn centrale regulatoren

die belangrijk zijn voor de overleving van het organisme (Yeger-Lotem & Margalit, 2003; Mason & Verwoerd, 2008; De Smet & Marchal, 2010). Belangrijke knooppunten in een netwerk zijn betrokken bij veel interacties, zoals de graadscentraliteitsparameter(C_d) aangeeft. Deze is gelijk aan de graad ('deg') van een knooppunt (Mason & Verwoerd, 2008) zoals formule 1.1 vermeldt.

$$C_d(u) = deg(u) \tag{1.1}$$

Hoewel biologische netwerken in het algemeen schaal-vrij verondersteld worden, wijken ze soms af van deze exponentiële verdeling. Tot op heden is er dus nog steeds discussie of deze schaalvrijheid een inherente eigenschap van een biologisch netwerk is (Khanin & Wit, 2006; Lima-Mendez & van Helden, 2009).

1.2.2 'Small world property'

De 'small world property' geeft aan dat de gemiddelde padlengte en diameter van het netwerk zeer klein zijn in vergelijking met de netwerkgrootte ($\sim$ log(n)) en dat bijgevolg twee knooppunten kunnen verbonden worden door een korter pad dan verwacht in een random netwerk met evenveel knooppunten en verbindingslijnen (Alm & Arkin, 2003; Mason & Verwoerd, 2008). Alm & Arkin (2003) stelden bovendien vast dat de netwerkdiameter niet varieert tussen verschillende organismen. In deze context is de 'betweenness' centraliteitsparameter (C_b), die aangeeft op hoeveel van de paden tussen andere knooppunten een bepaald knooppunt ligt (formule 1.2), een belangrijke maat om aan te geven hoe belangrijk dit knooppunt is voor het netwerk.

$$C_b(w) = \sum_{(u,v)\in V(w)} \frac{\sigma_{uv}(w)}{\sigma_{uv}} \tag{1.2}$$

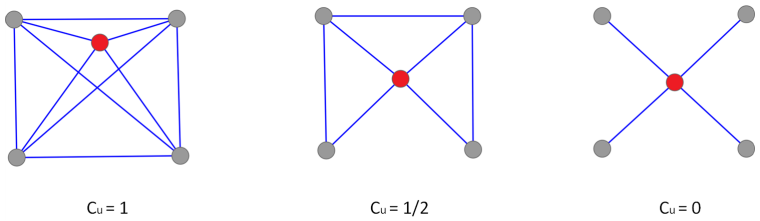
met σ_{uv} het aantal paden tussen knooppunt u en v en $\sigma_{uv}(w)$ het aantal paden tussen u en v dat via knooppunt w gaat (Mason & Verwoerd, 2008). Biologisch gezien kan dit in vraag gesteld worden omdat de weergegeven paden 'shortcuts' kunnen zijn die niet relevant zijn in de bestudeerde conditie (Lima-Mendez & van Helden, 2009).

1.2.3 Clusteringscoëfficiënt

De C_u geeft aan in welke mate een diagram bestaat uit clusters, dit zijn subsets van knooppunten met vele verbindingslijnen die deze knooppunten met elkaar verbinden (Pavlopoulos *et al.*, 2011). De berekening van de C_u gebeurt op basis van de verbindingen tussen de directe buren van het beoogde knooppunt (Ravasz *et al.*, 2002). Voor een volledig netwerk wordt C_u gedefinieerd volgens vergelijking 1.3.

$$C_u = \frac{1}{N} \sum_{i=1}^{N} \frac{E_i}{k_i(k_i - 1)} \tag{1.3}$$

met N het aantal knooppunten, E_i het aantal verbindingslijnen tussen de k buren van een bepaald knooppunt en k_i de graad van dat knooppunt. Figuur 1.3 maakt het mogelijk deze formule makkelijk te interpreteren. Indien alle buren van een bepaald knooppunt (hier in het rood aangegeven) met elkaar verbonden zijn, is de C_u gelijk aan 1. Voor verbindingen tussen de helft van de buren is deze gelijk aan 1/2 en als geen van de buren van het betreffende knooppunt met elkaar verbonden zijn is deze 0 (Ravasz *et al.*, 2002).



Figuur 1.3. Berekening van de clusteringscoëfficiënt voor een knooppunt in een netwerk: deze is gelijk aan respectievelijk 1, 1/2 of 0 als respectievelijk alle, de helft of geen van de buren van het betreffende knooppunt met elkaar verbonden zijn Ravasz *et al.* (2002).

Een netwerk heeft meer neiging om clusters te vormen als C_u meer naar 1 nadert (Pavlopoulos *et al.*, 2011). Voor biologische netwerken geldt dat $C_u(k) \sim k^{-1}$ zodat C_u zal dalen naarmate de graad van een knooppunt stijgt. Bijgevolg is de omgeving van knooppunten met een lage graad sterk geclusterd en die van 'hubs' dun geclusterd. Dense clusters van knooppunten met een lage graad vormen individuele modulen die door 'hubs' verbonden worden tot een netwerk. Deze eigenschap is inherent aan eiwitinteractienetwerken, maar minder of niet aanwezig in metabolische en regulatorische netwerken omwille van hun hiërarchische structuur (figuur 1.1) (Lee *et al.*, 2002; Guimerà & Amaral, 2005; Mason & Verwoerd, 2008; Peregrín-Alvarez *et al.*, 2009).

1.3 Constructie van biologische netwerken

Sectie 1.1 haalde reeds enkele experimenten aan die interactiedata kunnen leveren voor de opbouw van biologische netwerken. Daarnaast kunnen de data die de biologische netwerken onderbouwen ook afkomstig zijn uit de literatuur of voorspeld worden op basis van gelijkenissen met andere organismen. Sectie 1.3.1 beschrijft hoe de informatie beschikbaar in de literatuur onttrokken kan worden. De voorspelling van interacties op basis van geconserveerde co-expressie en orthologie wordt beschreven in sectie 1.3.2. Sectie 1.3.3 behandelt de integratie van al deze kennis in databanken.

1.3.1 Interactiedata uit literatuur

De literatuur vormt een basisinformatiebron van functionele en fysieke – omdat de resultaten van vele experimenten in de literatuur beschreven worden – interactiedata (Donaldson *et al.*,

2003; Cohen & Hunter, 2008; Clark *et al.*, 2012; Miljkovic *et al.*, 2012). Zo construeerden experts handmatig verschillende biologische netwerken, waaronder het macrofagenactivatiemodel van Raza *et al.* (2010), door middel van een grondige inspectie van de literatuur (Miljkovic *et al.*, 2012). Tegenwoordig kan deze enorme hoeveelheid beschikbare informatie doorzocht worden met programma's die biologische termen herkennen in de literatuurdocumenten, zoals bijvoorbeeld informatie-extractiesystemen preBIND, Textomy (Donaldson *et al.*, 2003) en EVEX (Van Landeghem *et al.*, 2011).

1.3.2 Voorspellen van fysieke interacties

Interactiedata zijn meestal slechts beschikbaar voor een beperkt aantal modelorganismen, maar kunnen door geconserveerde co-expressiepatronen en orthologie overgedragen worden naar andere organismen (Snel *et al.*, 2004; von Mering *et al.*, 2007; Schneider *et al.*, 2007; Tirosh *et al.*, 2007).

Co-expressie van genen is geconserveerd over species heen en genen met geconserveerde co-expressiepatronen tussen species hebben waarschijnlijk een gemeenschappelijke functie (Snel *et al.*, 2004). Correlaties in genenprofielen kunnen bijgevolg vergeleken worden tussen species om de interacties tussen genen te voorspellen (Tirosh *et al.*, 2007). Verschillende databanken voor expressiedata zijn GEO (Gene Expression Omnibus) (Edgar *et al.*, 2002), ArrayExpress (Brazma *et al.*, 2003) en COLOMBOS (COLlection Of Microarrays for Bacterial OrganismS) (Meysman *et al.*, 2014).

Genen in verschillende species die afkomstig zijn van eenzelfde gen in de laatste gemeenschappelijke voorouder zijn orthologen. Ze behouden dezelfde functie doorheen de evolutie waardoor identificatie van orthologen tussen species het mogelijk maakt genenfuncties te voorspellen (Fitch, 1970). Bovendien kunnen interacties, bewezen in modelorganismen, overgedragen worden naar andere organismen die orthologen van de interagerende eiwitten, genen of metabolieten bevatten (von Mering *et al.*, 2007). Er bestaan reeds verschillende algoritmen die hiervoor gebruikt kunnen worden. Enkele voorbeelden voor prokaryoten zijn de COG-methode van Tatusov *et al.* (1997), KEGG Orthology (Kanehisa *et al.*, 2004), RoundUp (DeLuca *et al.*, 2006), OMA (Schneider *et al.*, 2007) en eggNOG (Jensen *et al.*, 2008).

1.3.3 Integratie in databanken

Om de toegang tot al de beschikbare interactie-informatie te vereenvoudigen hebben verschillende groepen en consortia integrerende databanken ontwikkeld (Sardiu & Washburn, 2011). Tabel 1.1 geeft een overzicht van een aantal veel gebruikte biologische databanken.

Tabel 1.1. Eigenschappen van een aantal veel gebruikte biologische databanken

Databank	Organisme	Soort interactie	Soort data	Databron	Referentie
BioCyc	algemeen	eiwit-eiwit eiwit-DNA metabolisch	functioneel	literatuur	Caspi *et al.* (2012) Karp *et al.* (2005)
BioGRID	algemeen[1,7]	eiwit-eiwit eiwit-DNA	fysiek genetisch	literatuur Y2H, MS	Stark *et al.* (2006)
BRENDA	algemeen	metabolisch	functioneel	literatuur	Schomburg *et al.* (2002)
DIP	algemeen[1-7]	eiwit-eiwit	fysiek	literatuur	Xenarios *et al.* (2000)
Ecocyc	*E. coli* K-12	eiwit-eiwit eiwit-DNA metabolisch	functioneel	literatuur	Karp *et al.* (1997) Karp *et al.* (2002)
FlyNet	*D. melanogaster*	eiwit-eiwit eiwit-DNA	fysiek genetisch	ChIP	Sanchez *et al.* (1999) Tian *et al.* (2009)
HPRD	*H. sapiens*	eiwit-eiwit	fysiek	literatuur	Wilson (2004)
KEGG	algemeen	metabolisch signaal	functioneel	literatuur orthologie	Kanehisa & Goto (2000) Kanehisa *et al.* (2004)
MINT	algemeen[1,4,5,7]	eiwit-eiwit	functioneel	literatuur Y2H	Zanzoni *et al.* (2002)
PHOSIDA	algemeen	signaal	fysiek	sequenties	Gnad *et al.* (2011)
RegulonDB	*E. coli*	eiwit-DNA sRNA sigmafactoren	fysiek	literatuur ChiP micro-arrays	Huerta *et al.* (1998) Salgado *et al.* (2013)
SGD	*S. cerevisiae*	eiwit-eiwit eiwit-DNA metabolisch	functioneel	literatuur Y2H, MS ChIP	Cherry *et al.* (2012)
STRING	algemeen	eiwit-eiwit	functioneel fysiek	literatuur orthologie co-expressie	Snel *et al.* (2000) von Mering *et al.* (2003)

Legende: BioGRID 'Biological General Repository for Interaction Datasets', DIP 'Database for Interacting Proteins', HPRD 'Human Protein Reference Database', MINT 'Molecular INTeraction database', SGD '*Saccharomyces* Genome Database', Y2H Yeast-to-Hybrid, MS massaspectrometrie, ChIP chromatine-immunoprecipitatie, [1] *Saccharomyces cereviae*, [2] *Helicobacter pylori*, [3] *Escherichia coli*, [4] *Caenorhabditis elegans*, [5] *Drosophila melanogaster*, [6] *Mus musculus*, [7] *Homo sapiens*.

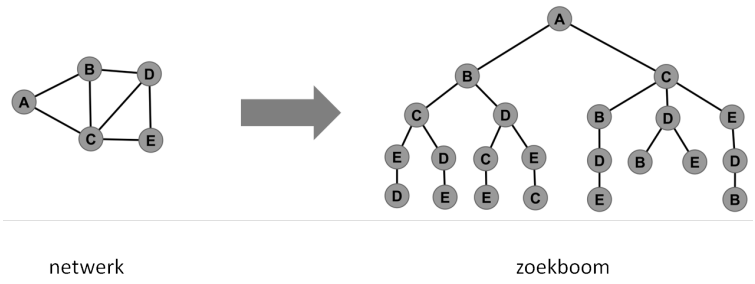
Hoofdstuk 2

Zoekalgoritmen

Dit hoofdstuk bespreekt een aantal algoritmen die in staat zijn de interactienetwerken, besproken in hoofdstuk 1, te doorzoeken. Dit zijn enerzijds padzoekende algoritmen (sectie 2.1) en anderzijds optimalisatiealgoritmen (sectie 2.2).

2.1 Padzoekende algoritmen

Een pad is een opeenvolging van verbindingslijnen die twee gegeven knooppunten (bijvoorbeeld A en E) in een bepaalde zoekruimte met elkaar verbindt (Russel & Norvig, 2003). Een padzoekend algoritme start hiervoor bijvoorbeeld in knooppunt A en moet vanaf hier in elk knooppunt een volgende verbindingslijn kiezen totdat het knooppunt E bereikt wordt. Om deze zoektocht geordend te laten verlopen wordt de zoekruimte voorgesteld als een zoekboom (figuur 2.1), waarin elke aftakking staat voor een mogelijk te kiezen volgende verbindingslijn (Bader & Madduri, 2006; Downsland, 2014). In het geval van deze studie is de zoekruimte een interactienetwerk en stelt een pad een cascade van interacties voor.

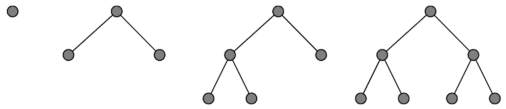


Figuur 2.1. Opstelling van een zoekboom voor een eenvoudig netwerk: (rechts) een eenvoudig netwerk met 5 knooppunten en 7 ongerichte verbindingslijnen, (links) de zoekboom voor een zoekactie startend in knooppunt A.

2.1.1 'Breadth-first' versus 'depth-first' zoeken

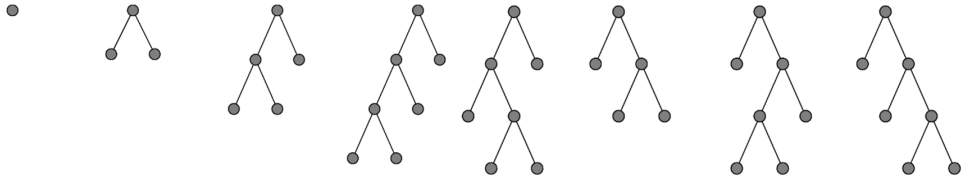
Twee naïeve en exhaustieve zoekalgoritmen zijn 'breadth-first search' en 'depth-first search'. Ze worden naïef en exhaustief genoemd omdat ze systematisch alle mogelijke verbindingslijnen evalueren zonder er rekening mee te houden of de volgende verbindingslijn wel naar het beoogde knooppunt leidt (Russel & Norvig, 2003).

De 'breadth-first' zoekstrategie werd geformuleerd door Moore (1959) en werkt, zoals de naam het zegt, in de breedte. Alle knooppunten op een bepaalde diepte in de zoekboom worden uitgebreid vooraleer de knooppunten van het volgende niveau uitgebreid worden zoals aangegeven in figuur 2.2 (Yoo *et al.*, 2005; Bader & Madduri, 2006). Een 'breadth-first search' bestudeert al de verschillende mogelijkheden en is in staat duplicaten die tot dezelfde toestand leiden te detecteren. Er is echter veel geheugenopslag vereist, omdat elk knooppunt in het geheugen moet opgeslagen blijven (Cormen *et al.*, 2001; Russel & Norvig, 2003; Korf & Schultze, 2005; Downsland, 2014).



Figuur 2.2. Uitbreiding van de zoekboom in 'breadth-first search'.

Een algoritme dat 'depth-first' zoekt, gaat daarentegen eerst helemaal omlaag in de zoekboom naar het diepste knooppunt dat geen opvolgers meer heeft. Vervolgens gaat de zoektocht terug omhoog naar het meest recent bezochte knooppunt dat nog onontgonnen opvolgers heeft (figuur 2.3). Het is een vorm van 'backtracking' ontwikkeld door Golomb & Baumert (1965) en werd voor het eerst toegepast in de algoritmen van Tarjan (1972) en Hopcroft & Tarjan (1973).

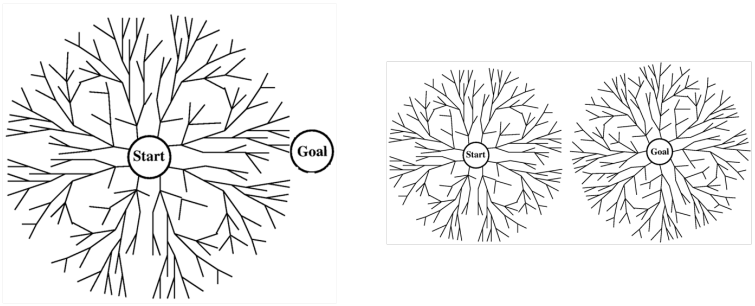


Figuur 2.3. Uitbreiding van de zoekboom in 'depth-first search'.

De geheugenvereiste is een stuk kleiner omdat een knooppunt uit het geheugen verwijderd wordt als al zijn nakomelingen volledig verkend zijn. Het algoritme is echter computationeel zeer intensief zodat er veel interesse is in parallelle versies ervan (Rao & Kumar, 1987). Bovendien kan het feit dat het algoritme onmiddellijk één pad volgt ook een nadeel zijn. Het kan bijvoorbeeld vastlopen in een zeer diep pad zonder oplossing, terwijl een ander pad een oplossing vlakbij het bronknooppunt kan bevatten (Korf, 1985; Cormen *et al.*, 2001; Russel & Norvig, 2003). Dit probleem kan opgelost worden door de toekenning van een dieptelimiet zoals in 'iterative deepening search' (Korf, 1985).

2.1.2 Uni-directioneel versus bi-directioneel zoeken

Een bi-directionele zoektocht vormt gelijktijdig paden vanuit twee vertrekpunten, het bronknooppunt en het doelknooppunt, totdat de twee fronten elkaar ontmoeten (Nelson & Toptsis, 1992). De straal van elk lopend front is maar de helft van deze bij uni-directionele zoektochten, hetgeen veel gunstiger is wat complexiteit betreft (tabel 2.1: $2 * O(b^{d/2}) <<< O(b^d)$). Figuur 2.4 visualiseert dit door middel van de grootte van de oppervlakte van de lopende fronten.



Figuur 2.4. Vergelijking complexiteit uni-directioneel (links) en bi-directioneel (rechts) zoeken.

De ontmoeting treedt bij 'wave-shaping' algoritmen op in het midden van het pad, door telkens het knooppunt dat het dichtst bij het ander front ligt uit te breiden (Sint & de Champeaux, 1977). Deze techniek vindt steeds een oplossing, maar is computationeel veeleisend omdat in elke stap de berekening van het meest geschikte knooppunt voor uitbreiding vereist is (Nelson & Toptsis, 1992). Deze beperking wordt opgelost in 'non-wave-shaping' algoritmen die snel tot een oplossing komen door knooppunten die waarschijnlijk niet tot een oplossing leiden te elimineren (Kwa, 1989).

Nicholson (1966) introduceerde het principe, waarna het door Pohl (1970) toegepast werd op de algoritmen die het kortste pad zoeken (Pijls & Post, 2009). Een optimaal geheugengebruik is de motivatie voor deze strategie. Dit idee van zoeken in twee richtingen wordt echter niet toegepast in 'depth-first search' algoritmen omdat deze de lopende fronten niet opslaan (Felner *et al.*, 2010).

2.1.3 Evaluatie

De prestatie van een algoritme kan aan de hand van vier parameters geëvalueerd worden, namelijk de volledigheid, de optimaliteit, de tijds- en de ruimtecomplexiteit. De volledigheidsparameter geeft aan of het algoritme een oplossing kan vinden als er één aanwezig is. Het vinden van de optimale oplossing wordt weergegeven in de optimaliteitsparameter. De complexiteit in tijd en ruimte geven respectievelijk aan hoe lang het duurt om een oplossing te vinden en hoeveel geheugen hiervoor nodig is (Russel & Norvig, 2003). Tabel 2.1 geeft een overzicht van deze parameters voor de besproken padzoekende algoritmen.

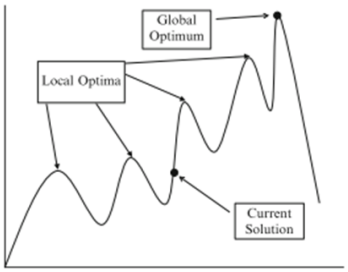
Tabel 2.1. Evaluatie van padzoekende strategieën door Russel & Norvig (2003)

Zoekstrategie	Volledig?	Tijd	Ruimte	Optimaal?
'Breadth-first'	Ja	$O(b^{d+1})$	$O(b^{d+1})$	Ja
'Depth-first'	Nee	$O(b^m)$	$O(bm)$	Nee
'Iterative deepening'	Ja	$O(b^d)$	$O(bd)$	Ja
Bi-directioneel	Ja	$O(b^{d/2})$	$O(b^{d/2})$	Ja

Legende: b is de vertakkingsfactor; d is de diepte van de meest oppervlakkige oplossing; m is de maximum diepte van de zoekboom; l is de dieptelimiet.

2.2 Optimalisatiealgoritmen

Optimalisatiealgoritmen zoeken naar de best mogelijke oplossing – bepaald door een scorefunctie – gegeven al de oplossingen (de zoekruimte) voor een bepaalde probleemstelling (Thede, 2004; Burke & Kendall, 2014). Een mogelijke probleemstelling in het kader van dit werk is bijvoorbeeld de verbinding van zo veel mogelijk gegeven knooppunten met zo weinig mogelijk verbindingslijnen. De zoekruimte bestaat dan uit alle paden die twee ingevoerde genen met elkaar verbinden en om de oplossing te vinden moeten alle combinaties van paden die twee genen uit de input verbinden, getest worden. Deze zoektocht kan volledig ('global search') of lokaal ('local search') gebeuren. Lokale zoekalgoritmen bewegen steeds naar naburige of verwante oplossingen van de huidige oplossing (Glover, 1989) zodat ze kunnen vastlopen in een lokaal optimum (figuur 2.5).



Figuur 2.5. Vergelijking lokaal en globaal optimum: lokale zoekalgoritmen bewegen enkel naar naburige oplossingen zodat ze kunnen vastlopen in een lokaal optimum (Burke & Kendall, 2014).

2.2.1 'Hill climbing'

Het 'hill climbing' zoekalgoritme is een lokaal optimalisatiealgoritme dat steeds zal bewegen naar een verwante oplossing met een hogere score, namelijk heuvelopwaarts. De best mogelijke oplossing wordt gevonden als een piek wordt bereikt waar geen enkele buur-oplossing nog een hogere score heeft (Glover, 1989). Het algoritme moet enkel de huidige oplossing en bijhorende score onthouden, hetgeen minder geheugen vereist dan een hele zoekboom.

'Stochastic hill climbing' (Rosete-Suarez & Ochao-Rodriguez, 1999) en 'enforced hill climbing' (Akramifar & Ghassem-Sani, 2010) zijn varianten op deze strategie. Al deze algoritmen zijn echter onvolledig, daar ze vast kunnen lopen in een lokaal maximum zonder een oplossing te vinden (figuur 2.5). Het 'random-restart hill climbing' algoritme (Boyan & Moore, 2001) lost deze beperking op door nieuwe 'hill climbing' zoektochten te starten vanaf willekeurige beginposities totdat een oplossing gevonden is.

2.2.2 'Simulated annealing'

'Simulated annealing' is een algoritme dat 'hill climbing' (efficiënt, maar onvolledig) combineert met een 'random walk' (volledig, maar inefficiënt) en de voordelen van beide methoden benut (Kirkpatrick, 1984). Dit algoritme beschouwt in iedere stap van het iteratieproces een volgende naburige oplossing t' en beslist probabilistisch of het naar deze oplossing zal overgaan of in de huidige oplossing t zal blijven. Indien een overgang naar de nieuwe oplossing t' optreedt, worden alle naburige oplossingen gerandomiseerd en begint het proces opnieuw. Hierbij wordt de probabiliteit om een slechtere oplossing te kiezen geleidelijk aan kleiner, zodat het algoritme de meest optimale oplossing benadert (Kirkpatrick *et al.*, 1983).

Indien het algoritme oneindig blijft lopen zal uiteindelijk het globale optimum bekomen worden. In de meeste gevallen wordt de optimalisatie echter beperkt door een computationeel limiet en zal de beste oplossing die op dat moment aanwezig is, het resultaat zijn (Granville *et al.*, 1994). Deze techniek is wel beter dan een gewoon 'hill climbing' algoritme, aangezien het niet kan vastlopen in een lokaal optimum en dus wel volledig is (Kirkpatrick, 1984).

2.2.3 'Beam search'

Het 'local beam search' algoritme ontstond als een variant van dynamisch programmeren voor spraakherkenning in het HARPY systeem (Lowerre, 1990) en een verwant algoritme wordt besproken door Pearl (1984). Dit algoritme houdt in plaats van slechts 1 oplossing, k oplossingen bij in zijn geheugen. De zoektocht start vanaf k willekeurig gekozen oplossingen. In elke stap worden de k beste opvolgers van deze k oplossingen gekozen totdat het algoritme zijn doel bereikt (Glover, 1989). Op deze manier worden onvruchtbare zoektochten snel beëindigd en worden alle middelen ingezet op de zoektochten die het meeste vooruitgang boeken. Het geheugen moet wel k oplossingen en hun scores onthouden, zodat deze algoritmen meer opslagruimte vereisen (Russel & Norvig, 2003). Bovendien kan 'local beam search' bij een slechte keuze van de k beginoplossingen geconcentreerd worden in een zone met weinig variatie in oplossingen. Dit probleem werd opgelost door er een 'stochastic beam search' van te maken (Wang & Lim, 2007).

2.2.4 Genetische algoritmen

Genetische algoritmen passen de principes van natuurlijke selectie toe om een bepaalde scorefunctie te optimaliseren. De startpopulatie van kandidaat-oplossingen wordt meestal willekeurig gekozen, maar kan ook reeds gebaseerd zijn op voorkennis (Goldberg & Holland, 1988; Thede, 2004; Albayrak & Allahverdi, 2011; Sastry *et al.*, 2014). Uit deze startpopulatie selecteert een genetisch

algoritme de kandidaat-oplossingen met de beste fitness om mee verder te werken. De fitness van een kandidaat-oplossing is een maat voor hoe goed deze oplossing het probleem oplost. Deze is uiteraard afhankelijk van het op te lossen probleem en wordt weergegeven in de te optimaliseren scorefunctie (Thede, 2004).

Het is de bedoeling betere oplossingen te bevoordelen ten opzichte van slechtere (Sastry *et al.*, 2014), hetgeen kan gebeuren via (i) fitness evenredige of (ii) ordinale selectie (Goldberg & Holland, 1988). Voorbeelden van fitness evenredige selectie zijn 'roulette wheel selection' en 'stochastic universal selection' waar de kandidaat-oplossingen met betere eigenschappen een hogere kans hebben om willekeurig geselecteerd te worden (Lipowski & Lipowska, 2012). Bij ordinale selectie worden de zoveel beste kandidaat-oplossingen geselecteerd zoals bijvoorbeeld in 'truncation selection' en 'tournament selection' (Sivaraj & Ravichandran, 2011).

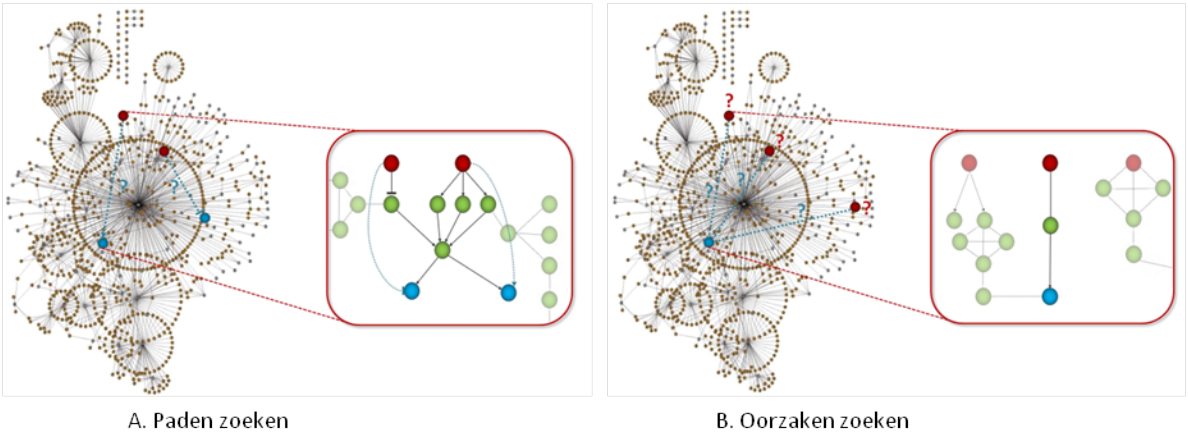
Vervolgens laat het algoritme variatie in de geselecteerde kandidaat-oplossingen toe door recombinatie en mutatie. Er bestaan verschillende recombinatiemethoden die meestal willekeurig twee kandidaat-oplossingen selecteren en deze recombineren met een bepaalde probabiliteit (Sastry *et al.*, 2014). Mutatiemethoden maken het mogelijk lokale oplossingen te vermijden door de kandidaat-oplossingen lichtjes aan te passen (Albayrak & Allahverdi, 2011).

De startpopulatie wordt vervangen door de geëvolueerde populatie, waarna het proces van selectie, recombinatie en mutatie opnieuw kan beginnen voor de volgende generaties.

Hoofdstuk 3

Subnetwerkselectie

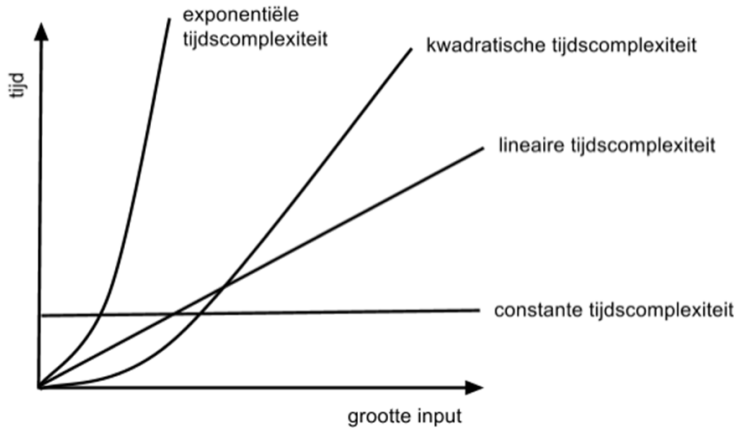
Hoewel interactienetwerken (hoofdstuk 1) een overzicht geven van al de beschikbare publieke kennis, zijn de delen die actief zijn onder bepaalde condities – subnetwerken genaamd – juist interessant. Dit hoofdstuk beschrijft verschillende methoden (figuur 3.1) die dit mechanisch inzicht proberen te verwerven door interactienetwerken te doorzoeken (hoofdstuk 2) op basis van functionele data. De geselecteerde subnetwerken bevatten dat deel van de fysieke interacties dat een bepaald fenotype verklaart. Hierbij wordt verondersteld dat genen die in elkaars nabijheid liggen in het interactienetwerk gerelateerd zijn met hetzelfde proces, ook wel het 'guilt-by-association' principe genoemd (Oliver, 2000).



Figuur 3.1. Twee werkwijzen om mechanisch inzicht te verwerven in interactienetwerken met behulp van functionele data: (A) zoeken naar paden op basis van oorzaak-effect paren en (B) zoeken naar oorzaken op basis van geobserveerde effecten. Oorzaken zijn in het rood aangegeven, gevolgen in het blauw, de blauwe stippellijn stelt het ongekende onderliggende pad tussen oorzaak en gevolg voor, pijlen tussen knooppunten wijzen op gerichte interacties die activerend (standaard pijl) of inhiberend (afgebroken pijl) kunnen zijn (Cloots *et al.*, 2014).

De methode om functionele data op het interactienetwerk van een organisme te verbinden verschilt naargelang de aard van het experiment. Het is enerzijds mogelijk dat de data een bepaalde oorzaak en zijn gevolgen presenteren, bijvoorbeeld de differentiële expressie volgend op de uitschakeling van een bepaald gen. De subnetwerkselectie bestaat dan uit het vinden van de onderliggende paden die de connectie tussen oorzaak en gevolg verklaren (figuur 3.1 A) en wordt aan de hand van voorbeelden besproken in sectie 3.1. Anderzijds kan het zijn dat er verschillende effecten geobserveerd worden waarvan de oorzaak niet gekend is. De subnetwerkselectiemethoden gedefinieerd in sectie 3.2 gaan op zoek naar de oorzaken verantwoordelijk voor de geobserveerde data (figuur 3.2 B).

Een subnetwerkselectieprobleem is echter 'non-deterministic polynomial-time hard' (NP-hard): het heeft een exponentiële tijdscomplexiteit zoals geïllustreerd in figuur 3.2. Voor eenvoudige datasets is het probleem oplosbaar, maar voor grote datasets bestaat er geen algoritme dat dit efficiënt kan doen (Bailly-Bechet *et al.*, 2009). De methoden beschreven in dit hoofdstuk kunnen de oplossing dus enkel proberen te benaderen.



Figuur 3.2. Vergelijking verschillende soorten tijdscomplexiteit: grootte van de input heeft geen invloed op de tijd nodig om het probleem op te lossen bij een constante tijdscomplexiteit, de tijdstoename is recht evenredig met de inputgrootte voor een lineaire tijdscomplexiteit, kwadratisch evenredig voor een kwadratische tijdscomplexiteit en exponentieel evenredig voor een exponentiële tijdscomplexiteit.

3.1 Identificatie van paden tussen oorzaak en gevolg

3.1.1 Fysieke netwerkmodellen

Fysieke netwerkmodellen werden initieel ontwikkeld door Yeang & Jaakkola (2003) als een kader om transcriptionele regulatie af te leiden. Ze beperken de complexiteit van biologische netwerken door enkel verifieerbare moleculaire eigenschappen van het onderliggende biologisch systeem te presenteren. Het voorkomen van eiwit-eiwit- en eiwit-DNA-interacties in regulatorische processen, de richting van signaaltransductie in eiwit-eiwitinteracties en het regulatorisch effect van deze interacties zijn voorbeelden van dergelijke verifieerbare moleculaire eigenschappen (Yeang & Jaakkola, 2003; Yeang *et al.*, 2004; Ourfali *et al.*, 2007).

Vervolgens stelden Yeang *et al.* (2004) bepaalde beperkingen aan deze eigenschappen die vertaald werden als potentiaalfuncties op het netwerk. Deze annotatie maakt het mogelijk het uiteindelijke subnetwerk te extraheren door optimalisatie van het product van de individuele potentiaalfuncties. Hoewel de resulterende subnetwerken destijds overeen kwamen met de kennis omtrent de transcriptionele regulatie in gist, leverden deze niet veel bijdrage in de zoektocht naar nieuwe inzichten (Yeang & Jaakkola, 2003; Yeang *et al.*, 2004; Ourfali *et al.*, 2007). De resulterende netwerken zijn waarschijnlijk te klein hiervoor omdat de opgelegde potentiaalfuncties zeer strikt zijn.

Het algoritme voor 'Signaling Pathway INferencE', SPINE, breidt dit concept verder uit door eveneens aan elk knooppunt een eigenschap – activatie of repressie – toe te kennen (Ourfali *et al.*, 2007). Op deze manier brachten Ourfali *et al.* (2007) naast de aard van de regulatorische eiwit-DNA-interacties ook de aard van de eiwit-eiwitinteracties in rekening. Het actieve subnetwerk wordt vervolgens gevonden door het verwachte aantal oorzaak-gevolgparen (s, t) dat ten minste één consistent pad ($K_{s,t} = 1$) bevat, te maximaliseren (formule 3.1). Ook hier bleek dat de gekozen potentiaalfuncties te strikt waren en nog verdere uitbreiding nodig is.

$$E(\sum_{s,t \in X} K_{s,t}) = \sum_{s,t \in X} E(K_{s,t}) = \sum_{s,t \in X} p(K_{s,t} = 1) \tag{3.1}$$

3.1.2 ResponseNet

ResponseNet is een stroming-gebaseerd ('flow') algoritme ontwikkeld door Lan *et al.* (2011) dat zoekt naar signaal- en regulatorische reactiewegen vertrekkende van genetische screenings die genetische oorzaken en fenotypische effecten identificeren. Hierbij wordt het interactienetwerk geïnterpreteerd als een stromingsnetwerk, een gerichte graaf waarin elke verbindingslijn een capaciteit en een kost bezit. Een belangrijke voorwaarde hierbij is dat de stroming door een verbindingslijn nooit groter kan zijn dan de capaciteit waardoor deze een limiet legt op de stroming die kan passeren (Cormen *et al.*, 2001). De kost van een verbindingslijn is bovendien gelijk aan het negatieve logaritme van zijn probabiliteit (Pr_{belief}), een waarde die elke verbindingslijn toegewezen krijgt om het geloof in de werkelijke interactie tussen de verbonden knooppunten aan te geven (Yeger-Lotem *et al.*, 2009). De signaal- en regulatorische reactiewegen worden geïdentificeerd op basis van specifieke netwerkmotieven zoals de 'mixed-feedback loops', 'cliques', 'coregulations' en 'regulatory complexes' gedefinieerd door Yeger-Lotem & Margalit (2003).

Door het interactienetwerk te modelleren als een stromingsnetwerk kan het actieve subnetwerk geselecteerd worden door de optimale stroming te berekenen (Yeger-Lotem *et al.*, 2009; Huang *et al.*, 2011; Lan *et al.*, 2011; Basha *et al.*, 2013). Hiervoor gebruikt ResponseNet een 'minimum-cost flow optimization' algoritme dat op zoek gaat naar de maximale stroming tussen twee knooppunten waarbij de kost van het verbindende pad minimaal is (Cormen *et al.*, 2001)(formule 3.2).

$$min\{\sum_{i,j \in V} -log(w_{ij}) * f_{ij}\} - \gamma * \sum_{i \in Source} f_{Si} \tag{3.2}$$

met w_{ij} en f_{ij} respectievelijk de Pr_{belief} van en de stroming door de verbindingslijn die knooppunt i en j verbindt, $-log(w_{ij})$ de kost voor deze verbindingslijn, S een hulpknooppunt dat de doelknooppunten voorstelt en γ een tuningsparameter die de grootte van het resulterende subnetwerk bepaald (Yeger-Lotem *et al.*, 2009; Huang *et al.*, 2011).

De implementatie van ResponseNet is momenteel beschikbaar als ResponseNet2.0 en eResponseNet (Huang *et al.*, 2011; Basha *et al.*, 2013). Deze bevatten naast het oorspronkelijke gewogen interactienetwerk voor *S. cerevisiae* (Lan *et al.*, 2011) ook een gewogen humaan interactienetwerk (Basha *et al.*, 2013) en de mogelijkheid om een eigen interactienetwerk in te laden. Bovendien kan de gebruiker van eResponseNet de tuningsparameter optimaliseren voor kandidaat-genen van de bestudeerde ziekte (Huang *et al.*, 2011). Deze implementaties hanteren echter geen causaliteitsverband tussen de oorzaken en resulterende effecten die de gebruiker invoert.

3.1.3 PheNetic

De Maeyer *et al.* (2013) ontwikkelden een algoritme dat subnetwerken kan selecteren voor prokaryoten en doopten dit PheNetic. Dit algoritme doorzoekt een interactienetwerk, verkregen uit publiek beschikbare data, met bron-effectgenenparen ('cause-effect pairs'). De bron stelt een mutatie voor in een bepaald gen die een verandering in stroomafwaarts gelegen genen in gang zet. Het effect is een gen dat verandering van expressieniveau ondervindt zoals zichtbaar is in expressieprofielen. Zo zal het algoritme uiteindelijk een regulatorisch pad definiëren dat de data het best beschrijft. In tegenstelling tot de ResponseNet-implementaties houdt dit algoritme dus rekening met de causaliteit tussen oorzaak en effect en moet deze aangebracht worden als een lijst van bron-effectgenenparen. Het gebruikte interactienetwerk bevat, naar analogie met ResponseNet, een Pr_{belief} voor elke verbindingslijn die het geloof erin uitdrukt. Verder krijgt ook elk knooppunt een probabiliteit toegewezen die zijn centraliteit in het netwerk aangeeft (C_b).

PheNetic modelleert het subnetwerkselectieprobleem als een theoretisch beslissingsprobleem, een formeel probleem met slechts twee mogelijke antwoorden: ja of nee (equivalent aan 1 of 0) (Darwiche, 2009). Het is de bedoeling het beste subnetwerk van interacties te selecteren uit een set van alle mogelijke interacties. Om te beslissen welke interacties het meest waarschijnlijk zijn en dus het beste subnetwerk uitmaken, kenden De Maeyer *et al.* (2013) een score toe aan elk subnetwerk. Deze is gelijk aan de som van de beloningen voor elk verklaard bron-effectpaar min een kostenterm die ervoor zorgt dat zo weinig mogelijk verbindingslijnen worden opgenomen. De maximalisatie (formule 3.3) hiervan resulteert in het beste subnetwerk. De beloning (positieve term) is hierin afhankelijk van de differentiële expressiegraad van het effect en de waarschijnlijkheid dat het regulatorisch pad bestaat in het geselecteerde subnetwerk.

$$S(D) = max\{(\sum_{(x,y)\in I} abs(A_{difex}(x,y))^n * p(path(x,y)|D,E)) - |D| * x_c\} \tag{3.3}$$

met $S(D)$ de totale score van het geselecteerde subnetwerk, D en E de knooppunten respectievelijk de verbindingslijnen hierin aanwezig, (x, y) een bron-effectgenenpaar in de input I.

3.2 Identificatie van oorzaken voor geobserveerde effecten

3.2.1 eQTL analyse

'Expression quantitative trait loci' (eQTL) zijn loci op het genoom waarvan de genetische variatie geassocieerd wordt met onder andere de hoeveelheid variatie in genexpressie (Gilad *et al.*, 2008). Technieken om deze eQTL te identificeren zijn overzichtelijk weergegeven in Hirschhorn & Daly (2005), maar het bepalen van de causale genen hierin blijft een uitdaging (Rockman & Kruglyak, 2006; Schadt & Lum, 2006). Bovendien reguleren de genen in de eQTL de genexpressie via de regulatie van transcriptiefactoren (Brem *et al.*, 2002; Yvert *et al.*, 2003), zodat netwerken die de signaaltransductie tussen deze transcriptiefactoren en de causale genen weergeven, een oplossing kunnen bieden.

3.2.1.1 Het algoritme van Tu

Tu *et al.* (2006) stelden hiertoe een netwerk-gebaseerd stochastisch algoritme voor dat het actieve subnetwerk voor de geobserveerde effecten bekomt door de uitvoering van 'random walks' (Doyle & Snell, 2000) in het interactienetwerk. Een 'random walk' in een netwerk zoekt op een willekeurige wijze een pad van een bepaald knooppunt naar een ander knooppunt. In elke stap van dit proces wordt de keuze van de volgende verbindingslijn bepaald door een opgelegde voorwaarde, bijvoorbeeld de waarschijnlijkheid dat deze interactie bestaat (Lovász, 1993). De opeenvolging van geselecteerde knooppunten is een Markov ketting, met als toestanden de knooppunten van het netwerk, en kan met behulp van de computer opgelost worden (Latapy & Pons, 2005).

Om de oorzaak van de differentiële expressie van een bepaald gen te identificeren worden twee invoerlijsten opgesteld: één met alle genen van de overeenkomstige eQTL en één met alle transcriptiefactoren die dit gen binden. De 'random walk' start vervolgens in een bepaalde transcriptiefactor en zal de genen in de eQTL met verschillende frequenties bezoeken (Lovász, 1993; Tu *et al.*, 2006). De selectie van de meest bezochte knooppunten en verbindingslijnen geeft aanleiding tot het actieve subnetwerk dat de data verklaart – dit is de oorzaken identificeert en de overeenkomstige reactiewegen aangeeft.

3.2.1.2 eQED

Suthram *et al.* (2008) maakten gebruik van de analogie tussen 'random walks' en elektrische circuits. Het verwachte aantal doortochten van een 'random walk' door een bepaald knooppunt of een bepaalde verbindingslijn is namelijk proportioneel met de hoeveelheid stroom die door deze

knooppunten of verbindingslijnen vloeit (Doyle & Snell, 2000). Het is dus mogelijk om biologische netwerken voor te stellen als elektrische circuits, die de gewichten op de verbindingslijnen als conducties ($1/resistentie$) modelleren. De kans op een associatie tussen verschillende knooppunten in dit elektrisch netwerk is dan gelijk aan de stroom die doorheen deze associatie loopt. Deze modellering heeft het voordeel dat de individuele stromen door elk knooppunt en elke verbindingslijn gemakkelijk te berekenen zijn met de wetten van Ohm en Kirchhoff (Irwin & Nelms, 2008).

De toepassing van deze theorie resulteerde in het algoritme 'eQTL electrical diagrams' (eQED) dat het actieve subnetwerk extraheert als de verzameling van knooppunten en verbindingslijnen waardoor het meeste stroom vloeit (Suthram *et al.*, 2008). Het optimale pad dat het causale gen verbindt met het doelwitgen is gedefinieerd als de kortste route met de hoogste totale som van stromen (formule 3.4).

$$min\{\sum_{u,v\in D}(d(u,v)-(V(u)-V(v)))\} \tag{3.4}$$

met $d(u,v)$ de gerichte verbindingslijn tussen knooppunten u en v en $V()$ de spanning over bepaalde knooppunten in het elektrische circuit. De combinatie van alle optimale paden vormt het uiteindelijke regulatorisch netwerk.

3.2.2 'Steiner trees'

Een 'Steiner tree' (ST) is een 'minimum spanning tree' (MST) – een grafische boom die het kortste pad tussen een gegeven set punten weergeeft – met extra knooppunten en verbindingslijnen ('Steiner nodes' en 'Steiner edges') die de totale lengte van de boom verkorten (Hwang *et al.*, 1992). Het vinden van een ST is triviaal (Scott *et al.*, 2005), het vinden van de meest optimale is moeilijker en het doel van het ST-probleem. Scott *et al.* (2005) pasten deze strategie voor het eerst toe op netwerken om, vertrekkende van een groot (sub)netwerk, met behulp van een ST een minimaal subnetwerk te selecteren (Bailly-Bechet *et al.*, 2009). Dit is de verzameling van de kortste verbindingen tussen de gegeven set van knooppunten (Hwang *et al.*, 1992), de termini genaamd, bekomen door de set van differentieel geëxpresseerde genen te verbinden op het interactienetwerk (Huang & Fraenkel, 2009).

3.2.2.1 Naïeve 'Steiner trees'

Een eerste (eenvoudige) methode om dit probleem op te lossen is de toepassing van een kortste paden benadering zoals deze van Takahashi & Matsuyama (1980) en Klein & Ravi (1995). Deze benaderingen gaan op zoek naar een opgegeven aantal kortste verbindingen tussen twee knooppunten, waarbij de lengte van een verbinding gelijk is aan de som van de lengtes van de gebruikte verbindingslijnen (Oliveira & Pardalos, 2011). Het algoritme van Takahashi & Matsuyama (1980)

past hiervoor het 'Recursive Enumeration Algorithm' (REA) van Jiménez & Marzal (1999) toe dat alle paden tussen twee bepaalde knooppunten ordent volgens hun lengte. Klein & Ravi (1995) gebruikten een 'all-to-all' kortste pad algoritme zoals dit van Dijkstra (1959) om de afstand tussen elk paar van knooppunten in een netwerk te bepalen. Het subnetwerk is dan gelijk aan de bekomen verzameling van kortste paden tussen twee knooppunten.

Een andere methode die een minimaal subnetwerk selecteert op basis van ST's is het 'kWalks' algoritme (Dupont *et al.*, 2006). Deze generische benadering van het ST-probleem voegt de verbindingslijnen die het meest gebruikt worden in een 'random walk' tussen twee knooppunten, samen in een subnetwerk. Deze methode kan gebruikt worden om een intermediair subnetwerk met vooraf vastgelegde grootte te onttrekken, vooraleer er een kortste pad algoritme op los te laten dat de uiteindelijke reactieweg zal bepalen (Faust *et al.*, 2010).

Het algoritme van Kou, Markowsky en Barman (1981) – in de literatuur bekend als de KMB heuristiek – is echter het meest populaire algoritme om het ST-probleem op te lossen (Oliveira & Pardalos, 2011). Dit algoritme zoekt eerst een MST, waarbij de kost van de verbindingslijnen gelijk gesteld wordt aan de minimale afstand tussen de opbouwende knooppunten. Er wordt een nieuw netwerk opgesteld dat voor elk paar van knooppunten in deze MST al de tussenliggende verbindingslijnen uit het oorspronkelijke netwerk toevoegt. Vervolgens worden alle redundante verbindingslijnen verwijderd, startend bij diegene met de hoogste kost (Kou *et al.*, 1981).

3.2.2.2 'Prize-collecting Steiner Trees'

Het ST-probleem zoals toegepast in de vorige algoritmen beperkt de oplossing echter tot de rechtstreekse of onrechtstreekse verbinding van de termini via de onderliggende verbindingslijnen in het netwerk (Huang & Fraenkel, 2009). Het 'prize-collecting Steiner tree' (PCST) probleem is een variant hiervan die deze voorwaarden versoepelt zodat niet alle termini in de oplossing moeten voorkomen (Bailly-Bechet *et al.*, 2009; Huang & Fraenkel, 2009; Tuncbag *et al.*, 2012). Hiertoe wordt aan elke interactie een kost gegeven die de geloofwaardigheid ervan weergeeft en aan elk knooppunt een sanctie die het geloof in de expressiedata voorstelt. Het doel van het PCST algoritme is de gelijktijdige minimalisatie van de kosten voor de geselecteerde verbindingslijnen (w_l) en de sancties voor de niet-geselecteerde termini (w_n) (Huang & Fraenkel, 2009; Tuncbag *et al.*, 2012). Het algoritme probeert met andere woorden het subnetwerk te vinden waarvoor formule 3.5 geminimaliseerd wordt (Bailly-Bechet *et al.*, 2009).

$$C = \sum_{links} w_l - \lambda \sum_{nodes} w_n \tag{3.5}$$

met λ als parameter om de balans tussen de optimalisatie van de termen te regelen.

Deel II

Materiaal en Methoden

Hoofdstuk 4

Netwerken en datasets

Dit hoofdstuk geeft een overzicht van de interactienetwerken en de datasets gebruikt in deze studie. Het organisme onder studie betreft *Salmonella enterica* enterica serovar Typhimurium str. LT2, verder *Salmonella* LT2 genoemd. Voor dit organisme wordt een interactienetwerk opgesteld zoals beschreven in sectie 4.1. De gebruikte functionele datasets zijn afkomstig van publiek beschikbare data (sectie 4.2) en experimenten in het eigen laboratorium (sectie 4.3).

4.1 *Salmonella* LT2 netwerken

Op basis van publiek beschikbare interactiedata (tabel 4.1) werden voor *Salmonella* LT2 twee interactienetwerken opgesteld die enkel verschillen in het eiwitinteractienetwerk.

Tabel 4.1. Samenstelling van het globale interactienetwerk voor *Salmonella* LT2

bron	soort interactie	organisme	# interacties	Pr_{belief}
Literatuur (netwerk 4.1.1) (Peregrín-Alvarez *et al.*, 2009)	eiwit-eiwit	*E. coli**	6644	origineel
STRING (netwerk 4.1.2) (Franceschini *et al.*, 2013)	eiwit-eiwit	*Salmonella* LT2	7600	origineel
RegulonDB (Salgado *et al.*, 2013)	eiwit-DNA, sRNA, sigmafactoren	*E. coli**	4924	origineel
KEGG (Kanehisa *et al.*, 2014)	metabool, fosforylatie, methylatie, ubiquitinatie	*Salmonella* LT2	3384	1
EVEX (Van Landeghem *et al.*, 2011)	eiwit-eiwit, eiwit-DNA	*Salmonella* LT2	636	0,1

Legende: # aantal, * omgezet naar *Salmonella* LT2 met behulp van orthologie 'mapping' (Schneider *et al.*, 2007).

4.1.1 Ortholoog eiwitinteractienetwerk gebaseerd op *E. coli*

Als basis voor dit netwerk dienen het eiwitinteractienetwerk van *E. coli* beschreven door Peregrín-Alvarez *et al.* (2009) en het regulatorisch netwerk van *E. coli* afkomstig van RegulonDB (Salgado *et al.*, 2013). Deze worden met behulp van orthologie 'mapping' (Schneider *et al.*, 2007) omgezet naar een basisnetwerk voor *Salmonella* LT2. De probabiliteit (Pr_{belief}) voor de beschreven interacties, wordt overgenomen uit de originele publicaties.

Een metabolisch netwerk en een signaalnetwerk specifiek voor *Salmonella* LT2 worden ontleend aan KEGG (Kanehisa *et al.*, 2014) en de geloofwaardigheid van deze interacties wordt gelijk gesteld aan 1. Uiteindelijk worden specifieke interacties voor *Salmonella* LT2 die niet aanwezig zijn in *E. coli* onttrokken uit de EVEX-data van Van Landeghem *et al.* (2011). Deze krijgen een geloofwaardigheid van 0,1. Het zo bekomen totale netwerk bestaat uit 2699 genen (of hun overeenkomstige genproducten) verbonden door 15709 interacties.

4.1.2 Eiwitinteractienetwerk specifiek voor *Salmonella* LT2

Tijdens deze studie werd ervoor geopteerd om het orthologe eiwitinteractienetwerk van *E. coli* te vervangen door een fysiek eiwitinteractienetwerk specifiek voor *Salmonella* LT2 beschikbaar via de STRING-databank (Franceschini *et al.*, 2013). Het regulatorisch, metabolisch en signaalinteractienetwerk zijn hetzelfde als beschreven in sectie 4.1.1. Het zo bekomen totale netwerk bevat 2829 genen (of hun overeenkomstige genproducten) en 14265 interacties.

4.2 Mitomycine dataset

De dataset om dit netwerk te toetsen aan de werkelijkheid werd gehaald op de COLOMBUS databank (Meysman *et al.*, 2014). Het betreft een experiment (GSE622) waar op verschillende tijdstippen na behandeling met mitomycine C de differentiële genexpressie drievoudig gemeten wordt voor 4432 *Salmonella* LT2 genen (2,1% missende waarden) (Frye *et al.*, 2005). Deze studie gebruikt de gemiddelde waarde van de metingen 30 minuten na behandeling met mitomycine C.

4.3 Imidazool datasets

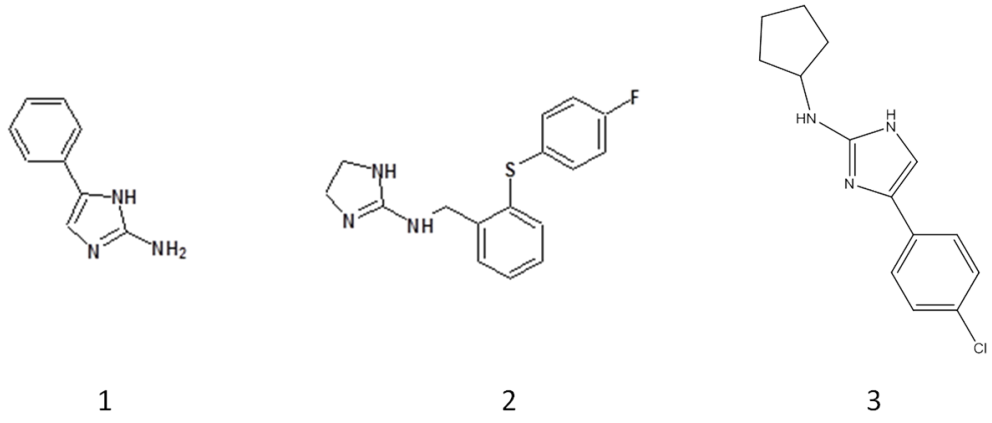
De onderzoeksdatasets bevatten de expressiedata van microarray experimenten en werden geleverd door de 'Salmonella & Probiotica' groep, deel van het Centrum voor Microbiële en PlantenGenetica te Leuven. Figuur 4.1 geeft de chemische structuur van de gebruikte componenten weer.

4.3.1 Behandeld versus controle

In een eerste reeks experimenten stelden onderzoekers vrijlevende wildtype *Salmonella* LT2 cellen (stam: ATC14220) bloot aan een imidazool (component 1), een imidazoline (component 2) en een controle solvent. Voor zowel de behandeling met imidazool als imidazoline werd de differentiële expressie van 5577 genen bepaald ten opzichte van de controleconditie. De metingen vonden vroeg in de exponentiële groeifase plaats. Enkel de genen met een p-waarde lager dan 0,1 werden meegenomen in de analyse (2111 voor component 1 en 1651 voor component 2).

4.3.2 Gevoelig versus ongevoelig

Een tweede reeks experimenten mat het verschil in differentiële expressie in de laat exponentiële groeifase tussen een voor imidazool (component 3) gevoelige en ongevoelige *Salmonella*-stam. Beiden stammen zijn afkomstig uit de SAR collectie van het 'Salmonella Genetic Stock Center'. De gevoelige stam (SGSC 2192) staat bekend als SAR-A12 en de ongevoelige stam (SGSC2460) als SAR-B3. De dataset beslaat 4574 genen, waarvoor de differentiële expressie van de gevoelige ten opzichte van de ongevoelige stam bepaald werd.



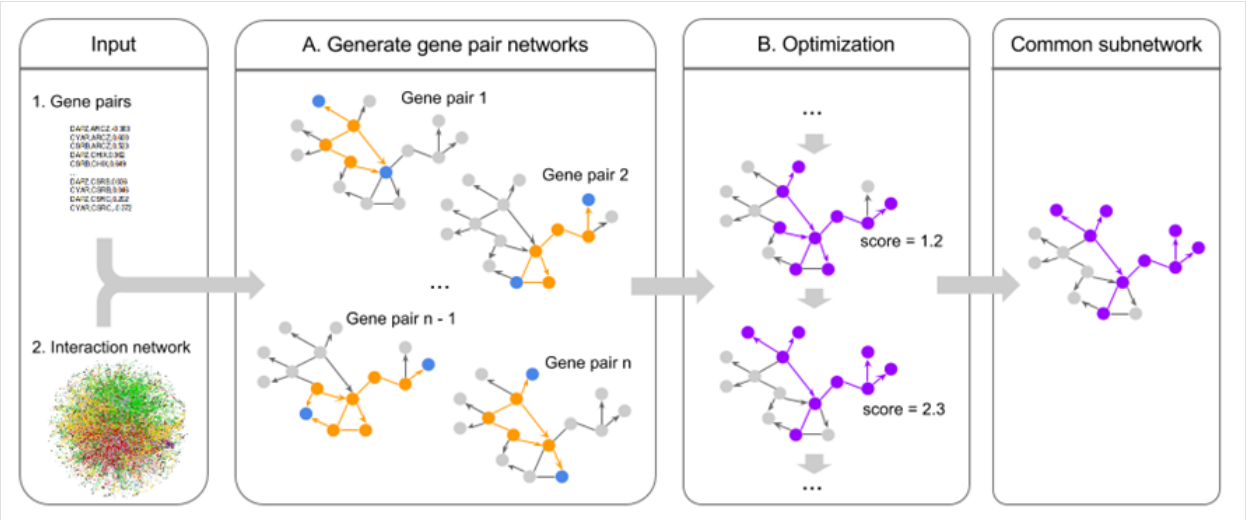
Figuur 4.1. Chemische structuren gebruikte componenten: (1) 4-fenyl-2-aminoimidazool, (2) *N*-[[2-[(4-fluorofenyl)thio]fenyl]methyl]-aminoimidazoline en (3) 5-(4-chlorofenyl)-2-(*N*-cyclopentyl)-aminoimidazool.

Hoofdstuk 5

Methode: PheNetic

Het in deze studie gebruikte PheNetic-algoritme is een uitbreiding van het algoritme beschreven in sectie 3.1.3. Deze nieuwe versie van het subnetwerkselectie-algoritme probeert een ingevoerde lijst van genenparen te verklaren door gemeenschappelijke genen en paden te zoeken in het totale interactienetwerk van het bestudeerde organisme (werkwijze B in figuur 3.1). Bovendien worden twee nieuwe probabiliteiten ingevoerd. Dit hoofdstuk is bedoeld om de theorie achter dit PheNetic-algoritme te bespreken, de gebruikte instellingen worden later per experiment meegegeven.

Het werkingsmechanisme van dit algoritme bestaat uit vier stappen (figuur 5.1), waarvan de constructie van genenpaarnetwerken (A) en de optimalisatie ervan (B) het eigenlijke kernalgoritme vormen. Dit hoofdstuk start met een korte beschrijving van het kernalgoritme waarna de vier verschillende stappen, zoals aanwezig op de webservice (http://bioinformatics.intec.ugent.be/phenetic/phenetic/index.html), besproken worden.



Figuur 5.1. Werkingsmechanisme van het subnetwerkselectie-algoritme PheNetic. Op basis van een opgegeven genenparenlijst en interactienetwerk selecteert PheNetic het optimale subnetwerk, dit is het subnetwerk dat zoveel mogelijk genenparen met elkaar verbindt met zo weinig mogelijk verbindingslijnen. (A) Eerst worden alle mogelijke paden per genenpaar (blauwe knooppunten) samengevoegd in genenpaarnetwerken die elk een stuk van de oplossing bevatten. (B) In de optimalisatie worden verschillende combinaties van deze deeloplossingen geëvalueerd tot het optimale subnetwerk gevonden wordt.

Het algoritme vertrekt van een opgegeven lijst van genenparen en een interactienetwerk voor het bestudeerde organisme, de generatie hiervan wordt besproken in sectie 5.1. Op basis van deze informatie zal het algoritme eerst alle mogelijke paden zoeken tussen elk genenpaar (blauwe knooppunten in figuur 5.1). Deze paden worden samengevoegd in een netwerk per genenpaar (oranje gekleurd in figuur 5.1), verder een genenpaarnetwerk genoemd. Dit genenpaarnetwerk is een klein deeltje van het totale interactienetwerk en bevat ook informatie over de uiteindelijke oplossing. Sectie 5.2 bespreekt hoe dit "zoeken naar paden" in zijn werk gaat en hoe het algoritme hieruit de input voor de daaropvolgende optimalisatiestap haalt. De optimalisatie, besproken in sectie 5.3, combineert al deze individuele genenpaarnetwerken opnieuw tot één netwerk dat vervolgens in een tekstbestand wordt gegoten (sectie 5.4).

5.1 Input genereren

De datasets gedefinieerd in het vorige hoofdstuk vereisen aanpassing tot de juiste invoerformaten voor PheNetic. Sectie 5.1.1 beschrijft een manier om dit te doen. Verder worden de probabiliteiten (Pr_{belief}) van het gebruikte interactienetwerk opnieuw gewogen om een optimaal resultaat te bekomen. Sectie 5.1.2 geeft een aantal suggesties hiervoor.

5.1.1 Genenparen

De beschikbare lijst van genen met hun differentiële expressiewaarden voor de verschillende experimenten, moet omgezet worden in een lijst van genenparen vooraleer ze als input voor PheNetic kan dienen. Dit is een komma-gescheiden tekstbestand dat alle mogelijke combinaties met twee differentieel geëxpresseerde genen uit de oorspronkelijke lijst bevat. Het stukje Scala-code in bijlage A.1 doet dit voor één richting, dit wil zeggen dat elk genenpaar slechts één keer voorkomt, en berekent de score als de som van de individuele differentiële expressiewaarden. Voor een genenparenlijst die de twee richtingen in rekening brengt, kan de code in bijlage A.2 gebruikt worden.

5.1.2 Interactienetwerk

De gegenereerde interactienetwerken voor *Salmonella* LT2 bevatten voor elke interactie een bijhorende Pr_{belief}. PheNetic behandelt deze als de kans dat de interactie werkelijk optreedt en gebruikt deze om de gevonden paden te evalueren (sectie 5.2). Het is mogelijk deze probabiliteiten aan te passen aan het specifiek probleem onder studie, herweging genoemd. Deze herweging van de probabiliteiten met voorafgaande kennis maakt het mogelijk het gegeven netwerk te manipuleren om zo meer relevante paden te vinden. In deze studie worden alle interacties opnieuw gewogen op basis van formule 5.1.

$$Pr_{gewogen} = Pr_{belief} * Pr_{hub} * Pr_{expressie} \tag{5.1}$$

Deze herweging kan enerzijds lagere probabiliteiten toekennen aan interacties die minder relevant geacht worden, bijvoorbeeld deze via 'hubs' (Pr_{hub}). Secties 5.1.2.1, 5.1.2.2 en 5.1.2.3 bespreken verschillende mogelijkheden hiervoor. Anderzijds kunnen hogere probabiliteiten toegekend worden aan meer relevant geachte interacties, zoals deze die leiden naar differentieel geëxpresseerde genen ($Pr_{expressie}$), hetgeen sectie 5.1.2.4 zal bespreken.

5.1.2.1 Specifieke 'hubs' verwijderen

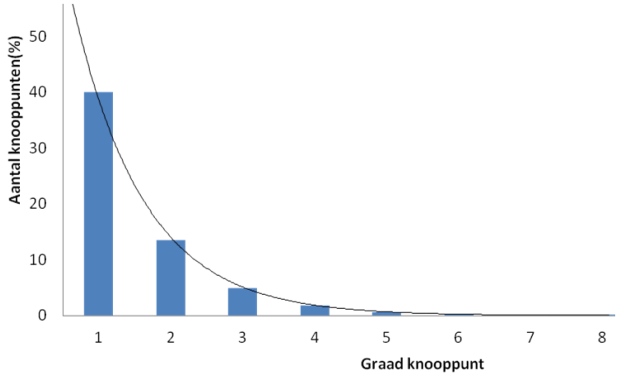
Zoals beschreven in sectie 1.2.1 kunnen biologische netwerken een aantal 'hubs' bevatten die bij heel veel interacties betrokken zijn en essentieel zijn voor de overleving van het organisme. Daar ze bij heel veel interacties betrokken zijn, zullen padzoekende algoritmen deze 'hubs' altijd terugvinden, zodat ze makkelijk tot de oplossing worden gerekend. Dit kan echter een vertekend beeld geven en andere betrokken genen maskeren. De eenvoudigste oplossing hiervoor is de verwijdering van deze specifieke 'hubs' ($Pr_{hub} = 0$) uit het interactienetwerk (code in bijlage A.3).

5.1.2.2 'Hubs' vermijden op basis van de exponentiële graadverdeling

Een tweede methode weegt de probabiliteiten van het netwerk opnieuw met behulp van de exponentiële graadverdeling van de knooppunten. Elke interactie in het netwerk wordt gewogen met de kans (P_{exp}) op een graad (deg) hoger dan deze van het eindknooppunt (k) van de interactie (vergelijking 5.2).

$$Pr_{hub}(k) = P_{exp}(X > deg(k)) = 1 - P_{exp}(X \leq deg(k)) \tag{5.2}$$

Met een stijgende graad daalt het aantal knooppunten exponentieel, zoals aangegeven in figuur 5.2. De oppervlakte onder de curve, $P_{exp}(X \leq deg(k))$, stijgt echter met de graad zodat de oppervlakte rechts van een bepaalde graad, $P_{exp}(X > deg(k))$, de kans weergeeft dat een willekeurig knooppunt een hogere graad bevat. Voor 'hubs' zal deze kans uiteraard klein zijn.

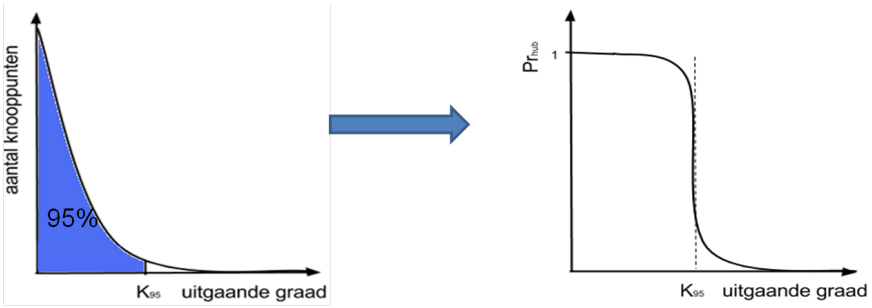


Figuur 5.2. De exponentiële graadverdeling in een biologisch netwerk.

Een punt van kritiek is wel dat de probabiliteit in het begin van de graadverdelingscurve zeer sterk daalt. De gebruikte code is weergegeven in bijlage A.4.

5.1.2.3 'Hubs' vermijden op basis van de sigmoïdale graadverdeling

Het grote nadeel van de herweging met behulp van de exponentiële graadverdeling is de snelle daling van de probabiliteiten. Hierdoor worden knooppunten met een relatief lage graad (bijvoorbeeld 10) al sterk benadeeld, hoewel deze niet echt als 'hubs' beschouwd worden. Een mogelijke oplossing is de verschuiving van de curve naar een sigmoïdale graadverdeling zoals voorgesteld in figuur 5.3.



Figuur 5.3. Opstelling sigmoïdale graadverdeling op basis van de exponentiële graadverdeling. Als buigpunt voor de sigmoïdale graadverdeling wordt de uitgaande graad waaronder 95% van de knooppunten valt, gekozen.

Op basis van de exponentiële graadverdeling wordt de uitgaande graad bepaald waaronder 95% van de knooppunten valt (K_{95}). Deze wordt gebruikt al buigpunt in een sigmoïdale graadverdeling die knooppunten met een lagere graad hoge probabiliteiten geeft en knooppunten met een hogere graad – de werkelijke 'hubs' – afstraft met zeer lage probabiliteiten. Deze 'hub'-probabiliteiten (Pr_{hub}) worden berekend met formule 5.3.

$$Pr_{hub}(k) = P_{sig}(X > deg(k)) = \sqrt{\frac{1}{1 + \frac{exp(deg(k) - K_{95})}{b}}} \qquad met \quad P_{exp}(X \leq K_{95}) = 0,95 \quad (5.3)$$

De parameter b werd empirisch bepaald en gelijk gesteld aan $\frac{1}{4} * K_{95}$ zodat knooppunten met tien uitgaande verbindingslijnen nog steeds een hoge Pr_{hub} hebben. Bovendien stelt de code in bijlage A.5 een ondergrens in voor deze herwegingsparameter gelijk aan 0,01.

5.1.2.4 Herweging met differentiële expressiewaarden

De herweging van het netwerk met de differentiële expressiewaarden (DE-waarden) van de inputfile maakt het mogelijk om meer genenparen te verklaren. Op deze manier worden immers de verbindingslijnen naar differentieel geëxpresseerde genen geloofwaardiger of belangrijker geacht, zodat de kans dat ze in de oplossing voorkomen groter is. Om correcte schattingen te maken is het wel belangrijk om de differentiële expressie van zoveel mogelijk genen in het totale netwerk te meten.

Formule 5.4 geeft de gebruikte herwegingsparameters ($P_{expressie}$) weer voor DE-waarden groter en kleiner dan nul.

$$DE > 0 : Pr_{expressie} = (P_{norm}(X \leq DE(k)) - 0,5) * 2 \tag{5.4a}$$

$$DE < 0 : Pr_{expressie} = (P_{norm}(X > DE(k)) - 0,5) * 2 \tag{5.4b}$$

Deze herwegingsparameter is gebaseerd op de normaalverdeling (P_{norm}) van de DE-waarden van alle genen in de gebruikte dataset. In dit geval willen we genen met hoge DE-waarden bevoordelen met hoge $Pr_{expressie}$-waarden zodat voor positieve DE-waarden de oppervlakte links van deze waarde genomen wordt en omgekeerd voor negatieve DE-waarden. Voor genen waarvoor geen metingen beschikbaar zijn, wordt deze herwegingsparameter gelijk gesteld aan 0,5. Merk op dat alle herwegingsparameters voor $Pr_{expressie}$ nu tussen 0,5 en 1 liggen, waardoor een kleine correctie moet toegepast worden om een waarde tussen 0 en 1 te krijgen.

Uiteraard bestaat elke interactie in het gebruikte interactienetwerk uit twee knooppunten, elk met een eigen herwegingsparameter. De herweging wordt daarom uitgevoerd met de hoogste van beide herwegingsparameters van de knooppunten die deel uitmaken van de interactie. De code weergegeven in bijlage A.6 maakt het mogelijk deze herweging uit te voeren.

5.2 Paden zoeken en genenpaarnetwerken genereren

Nu de input van PheNetic gespecificeerd is, kan het kernalgoritme van PheNetic op zoek gaan naar paden tussen elk opgegeven genenpaar. Deze zoektocht gebeurt door middel van een bi-directionele 'iterative deepening search' (sectie 2.1). Sectie 5.2.1 bespreekt de parameters die de gebruiker van PheNetic kan meegeven om de zoektocht te specificeren. Op basis van de gevonden paden wordt per genenpaar een klein netwerk opgesteld dat een deel van de oplossing bevat (sectie 5.2.2). De bijhorende functie voor de verbindingslijnen wordt omgezet in logische codetaal, besproken in sectie 5.2.3, die het algoritme verder kan analyseren.

5.2.1 Paden zoeken

PheNetic vraagt een paddefinitie voor de paden waar het algoritme naar moet zoeken, dit is een specificatie van het soort paden. Er is keuze tussen simpele paden, regulatorische paden en bi-directionele regulatorische paden. Bij selectie van de optie "simpele paden" zoekt het algoritme naar alle mogelijke paden ongeacht de soorten interacties die de opbouwende verbindingslijnen beschrijven. Voor de optie "regulatorische paden" probeert het algoritme de genenparen te verbinden via paden die enkel uit regulatorische interacties – eiwit-DNA-interacties, sigma en sRNA interacties – bestaan. En de optie "bi-directionele regulatorische paden", afgekort als bi-regulatorische paden, vereist dat elk pad start en eindigt met een regulatorische interactie.

Verder is het belangrijk om de maximale lengte van het pad vast te leggen. De meest relevante paden zijn toch slechts van een beperkte lengte, een gevolg van de 'small world property' van biologische netwerken (Alm & Arkin, 2003; Mason & Verwoerd, 2008) en zo vergt het algoritme minder computerkracht.

5.2.2 Genenpaarnetwerken

Elk gevonden pad krijgt een score op basis van de probabiliteiten van de opbouwende verbindingslijnen. Deze worden gerangschikt en de beste paden, waarvan het aantal zelf te bepalen is, zullen samengevoegd worden in een genenpaarnetwerk (figuur 5.1 A). Dit genenpaarnetwerk is een kleine mogelijke oplossing van het grote netwerk en geeft de meest relevante paden tussen het beschouwde genenpaar weer.

Een genenpaarnetwerk geeft eigenlijk, gegeven de zoveel beste paden in dit netwerk, de kans weer dat de twee genen in het genenpaar verbonden zijn en kan bijgevolg uitgedrukt worden als de functie in formule 5.5. Deze functie is uiteraard afhankelijk van het aantal verbindingslijnen (E) in het genenpaarnetwerk. PheNetic hanteert deze functie in de vorm van een logische codetaal, besproken in de volgende sectie.

$$f(E) = P(gen1, gen2) \tag{5.5}$$

5.2.3 CNF-conversie

'Conjunctive normal form' (CNF) is een formulering van logische clausules die door computers kan begrepen worden (Darwiche & Marquis, 2002). De CNF-conversie in PheNetic zet de gevonden paden dus om naar computerinterpreteerbare logische clausules. Deze logische codetaal hanteert een EN-OF-NIET-systeem om het pad te beschrijven. Een pad is meestal opgebouwd uit meerdere verbindingslijnen, die allemaal echt (of waar) moeten zijn opdat het pad bestaat in het beschreven netwerk. Figuur 5.4 geeft een voorbeeld van een CNF-output die de verbinding tussen twee genen in *E.coli*, *aceE* en *cyoA*, definieert in logische taal. Elke regel bestaat uit een OF-uitspraak zodat altijd aan exact één van de twee voorwaarden voldaan is. De connectie tussen de verschillende regels zijn EN-uitspraken en een minteken voor een voorwaarde staat voor een NIET-uitspraak.

Een computer zal de eerste drie regels in dit voorbeeld lezen als: ofwel bestaat pad 1 (aux0) niet ofwel bestaan er verbindingslijnen tussen (*hns* en *aceE*) en tussen (*hns* en *gadE*) en tussen (*gadE* en *cyoA*). Een pad bestaat dus enkel als al de verbindingslijnen die het pad uitmaken bestaan. De vierde regel wordt vertaald als: ofwel bestaat pad 1 ofwel bestaat er geen verbindingslijn tussen (*hns* en *aceE*) of tussen (*hns* en *gadE*) of tussen (*gadE* en *cyoA*). Als er één verbindingslijn ontbreekt

bestaat het pad niet. Regels vijf tot acht geven dezelfde logica weer voor een ander pad (aux1) dat mogelijk *aceE* met *cyoA* verbindt. De logische vertaling van de laatste drie regels is: ofwel bestaat er een regulatorisch pad tussen *aceE* en *cyoA* ofwel bestaat pad 1 niet, ofwel bestaat er een regulatorisch pad tussen *aceE* en *cyoA* ofwel bestaat pad 2 niet, ofwel bestaat er geen regulatorisch pad ofwel bestaat pad 1 ofwel bestaat pad 2.

```
-aux_0 edge_pp(hns,acee) 0
-aux_0 edge_pd(hns,gade) 0
-aux_0 edge_pd(gade,cyoa) 0
aux_0 -edge_pp(hns,acee) -edge_pd(hns,gade) -edge_pd(gade,cyoa) 0
-aux_1 edge_pp(ssb,acee) 0
-aux_1 edge_pp(cra,ssb) 0
-aux_1 edge_pd(cra,cyoa) 0
aux_1 -edge_pp(ssb,acee) -edge_pp(cra,ssb) -edge_pd(cra,cyoa) 0
regulatory_path(strain_acee,acee,cyoa) -aux_0 0
regulatory_path(strain_acee,acee,cyoa) -aux_1 0
-regulatory_path(strain_acee,acee,cyoa) aux_0 aux_1 0
```

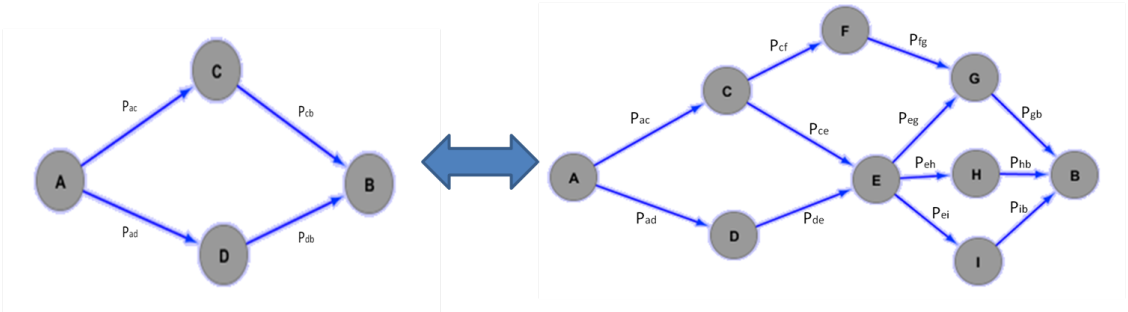
Figuur 5.4. CNF-output van de interactie tussen de genen *aceE* en *cyoA* in *E.coli*.

5.3 Optimalisatie

De optimalisatie bedraagt de samenvoeging van al de individuele genenpaarnetwerken in één globaal netwerk dat zoveel mogelijk genenparen verklaart met zo weinig mogelijk verbindingslijnen. Om deze grote hoeveelheid aan data snel te evalueren voert PheNetic eerst een kenniscompilatie uit besproken in sectie 5.3.1. Dan worden verschillende subnetwerken uitgetest die telkens gescoord worden (sectie 5.3.2), waarbij de keuze van de kost een belangrijke rol speelt (sectie 5.3.3).

5.3.1 Kenniscompilatie

Een computer kan het netwerk omgevormd in CNF-code interpreteren en hiermee berekeningen uitvoeren. Zo kan een algoritme bijvoorbeeld de kans berekenen dat er een pad bestaat tussen gen A en gen B in een eenvoudig netwerk (figuur 5.5 links). Er zijn twee paden van A naar B, via C of via D, elk met hun eigen probabiliteiten. De kans dat er een pad bestaat tussen A en B in het gegeven netwerk, kan dus berekend worden als $P_{ac} * P_{cb} + P_{ad} * P_{db}$. In een complexer netwerk (figuur 5.5 rechts) is deze berekening al zeer moeilijk en vereist ze veel rekenkracht.



Figuur 5.5. Vergelijking probabiliteiten in een eenvoudig (links) en een complex (rechts) netwerk.

Dit probleem kan sterk vereenvoudigd worden via kenniscompilatie ('knowledge compilation'), een techniek die vaak toegepast wordt in de artificiële intelligentie om computationeel moeilijke beredeneringsproblemen op te lossen. De oplossing van het probleem wordt in twee fasen uitgevoerd: er wordt een geschikte datastructuur opgebouwd waaruit de nodige informatie om het probleem op te lossen geëxtraheerd wordt (Cadoli & Donini, 1997). Voor PheNetic zet het programma DSHARP de CNF-code om in een sd-DNNF('smooth, deterministic, decomposable negation normal form'), een formaat dat het best te vergelijken is met een EN-OF boom waar het algoritme in elk punt een keuze moet maken op basis van de aanwezige interacties in het op dat moment geselecteerde subnetwerk (Muise *et al.*, 2012). Reeds bestaande algoritmen kunnen – door de conversie van deze sd-DNNF naar een arithmetisch circuit – de probabiliteit dat er een connectie is tussen een genenpaar, gegeven een set van geselecteerde interacties, efficiënt bepalen (Darwiche, 2009).

5.3.2 Beslissing van de beste strategie

Om terug een globaal netwerk te maken start PheNetic op een willekeurige plaats in het interactienetwerk en gaat het de gevonden paden willekeurig aan- en uitschakelen. Voor elk subnetwerk E wordt bepaald hoe goed deze de differentieel geëxpresseerde genen verklaart. Telkens er een beter subnetwerk gevonden wordt, neemt het algoritme dit als nieuw startpunt om vervolgens weer willekeurig paden aan en uit te schakelen. De correctheid van ieder subnetwerk schat PheNetic op basis van formule 5.6, die een beloning geeft voor elke verklaard genenpaar en een afstraffing voor elke extra verbindingslijn.

$$S(E) = \sum_{(x,y)\in I} \{P(x,y) * D(x,y)\} - |E| * x_c \tag{5.6}$$

De beloning (positieve term) voor een verklaard genenpaar is afhankelijk van, het aantal verklaarde genenparen, de waarschijnlijkheid dat de paden, die deze genenparen verbinden, bestaan in het geselecteerde subnetwerk ($P(x,y)$) en de graad van differentiële expressie van de verbonden genen ($D(x,y)$)). De kost (negatieve term) hangt af van de opgegeven genselectiekost (x_c) en het aantal geselecteerde verbindingslijnen in het subnetwerk.

Een 'hill-climbing'-implementatie (sectie 2.2.1) optimaliseert deze score ter selectie van het meest optimale subnetwerk. Door de toegepaste weging zal het algoritme reactiewegen via 'hubs' vermijden omdat deze door een lage $Pr_{gewogen}$ een lagere beloning krijgen. Bovendien kunnen ze nog steeds geselecteerd worden als er voldoende paden tussen de genenparen via het 'hub'-gen gaan.

5.3.3 Bepaling van de kost

De kans dat het algoritme een geschikt subnetwerk vindt dat de genenparen kan verklaren is afhankelijk van de opgegeven kost. Indien deze te hoog genomen wordt, zal geen enkel subnetwerk aan de strenge vereisten voldoen. Vergelijking 5.7 geeft de voorwaarde voor een positieve score.

$$kost < \frac{score * probabiliteit}{|E|} \tag{5.7}$$

5.4 Analyse-stap

De uiteindelijke oplossing zal afhankelijk zijn van het willekeurig gekozen startpunt voor de optimalisatie (sectie 5.3.2), vandaar wordt deze stap vaak meerdere malen gelopen. In de analyse-stap vergelijkt PheNetic de afzonderlijke resultaten voor elke optimalisatieronde om vervolgens alle geselecteerde verbindingslijnen op te lijsten samen met het aantal rondes waarin deze geselecteerd werden. De output is een tab-gescheiden tekstbestand dat voor elke interactie (elke lijn) het knooppunt waarin deze start, het type interactie, het knooppunt waarin deze eindigt en de frequentie waarmee deze geselecteerd werd in de optimalisatierondes, weergeeft. Dit bestand kan dan met visualisatie- en analyseprogramma's bestudeerd worden om het mechanisme dat tot de waargenomen differentieel geëxpresseerde genen leidt te ontrafelen. Bovendien kan de minimale frequentie waarmee een verbindingslijn in de oplossingen moet voorkomen, meegegeven worden.

5.5 Experimentele proefopzet

De genenparenlijsten voor experimenten die zoeken naar bi-regulatorische paden moeten elk genenpaar slechts éénmaal bevatten (code A.1). Daar bi-regulatorische paden symmetrisch zijn is het pad tussen A en B immers gelijk aan dit tussen B en A. Zoektochten naar simpele paden starten van genenparenlijsten die de twee richtingen bevatten (code A.2).

De gebruikte interactienetwerken worden herwogen door (1) de 'hubs' *rpo*D en *gro*EL te verwijderen (sectie 5.1.2.1), (2) de andere interacties te herwegen op basis van de uitgaande graad van de eindknooppunten en (3) de DE-waarden van de gebruikte dataset in rekening te brengen (sectie 5.1.2.4). Stappen (1) en (3) zijn hetzelfde voor elk experiment. Voor stap (2) bestaat nog de keuze tussen een herweging op basis van de exponentiële graadverdeling (sectie 5.1.2.2) of één op basis van de sigmoïdale graadverdeling (sectie 5.1.2.3) en wordt per experiment aangegeven in deel III.

Ook de gebruikte instellingen voor de padlengte (l), het aantal beste resultaten per genenpaar te selecteren (n), de kost voor de scoreberekening (c), het aantal optimalisatieronden (r) en de selectiefrequentie van de interacties in de oplossing (f) worden per experiment weergegeven.

Hoofdstuk 6

Analyse tools

De netwerken bekomen uit de subnetwerkselectie met PheNetic worden geanalyseerd met het visualisatieprogramma Cytoscape (versie 2.8.3) en het statistisch programma R (versie 3.0.2). De hieruit resulterende figuren worden aangepast met behulp van Microsoft Office 2007.

6.1 Vergelijking resultaten

De bekomen resultaten in deel III worden zowel met willekeurige netwerken als onderling vergeleken. De vergelijking met willekeurig genomen netwerken van dezelfde grootte – bijvoorbeeld in een hypergeometrische test (sectie 6.1.1) – geeft een idee hoe specifiek de resultaten zijn voor de gegeven probleemstelling. De netwerken kunnen onderling vergeleken worden door de Jaccard-index te berekenen hetgeen sectie 6.1.2 bespreekt.

6.1.1 Hypergeometrische test

Een hypergeometrische test geeft de kans dat een bepaald resultaat – bijvoorbeeld de aanwezigheid van bepaalde genen – kan bekomen worden door willekeurige (random) selectie. Deze test (formule 6.1) berekent de kans op minstens m successen voor n trekkingen zonder teruglegging uit een populatiegrootte N, die M successen (en dus $N - M$ mislukkingen) bevat.

$$P(X > m) = 1 - \frac{\binom{M}{m} * \binom{N-M}{n-m}}{\binom{N}{n}} \tag{6.1}$$

Kleine waarden voor een bepaald resultaat geven aan dat de kans dat dit resultaat met een willekeurige selectie bekomen wordt, klein is.

6.1.2 Jaccard-index

De Jaccard-index geeft de verhouding tussen de doorsnede en de unie van twee verzamelingen, in deze studie de subnetwerken. De doorsnede bevat alle verbindingslijnen die beide subnetwerken gemeenschappelijk hebben en de unie bevat alle verbindingslijnen die in één of beide subnetwerken voorkomen. Formule 6.2 berekent de Jaccard-index voor de subnetwerken A en B. Dit resulteert in een waarde tussen 0 – voor geen gelijkenis – en 1 voor volledig hetzelfde.

$$Jaccard(A, B) = \frac{|A| \bigcap |B|}{|A| \bigcup |B|} \tag{6.2}$$

6.2 Functionele annotatie

De genen teruggevonden in deze studie worden functioneel geannoteerd op basis van een verrijkingsanalyse gevolgd door een manuele indeling in functionele groepen.

6.2.1 Verrijkingsanalyse in Cytoscape

De 'Biological Networks Gene Ontology' (BiNGO) tool (versie 2.44) van Cytoscape geeft aan welke functionele groepen, volgens de 'Gene Ontology' (GO) termen, verrijkt zijn in een gegeven set van genen (Maere *et al.*, 2005). Dit gebeurt ook op basis van een hypergeometrische test zoals gedefinieerd in sectie 6.1.1. De resulterende p-waarden worden bovendien aangepast met een Benjamini-Hochberg correctie om het aantal vals-positieve resultaten te controleren (Benjamini & Hochberg, 1995).

6.2.2 BioCyc

De onderlinge afhankelijkheid van de functionele groepen in de GO hiërarchie resulteert vaak in meerdere verwante verrijkte groepen (Maere *et al.*, 2005), zodat een manuele opzuivering nodig is voor een éénduidige indeling. Deze gebeurt aan de hand van de BioCyc databank voor *Salmonella* LT2 (versie 18.0) (Karp *et al.*, 2005; Caspi *et al.*, 2012).

6.3 Biologische validatie

Om de subnetwerken bekomen met PheNetic, biologisch te valideren wordt de precisie en de gevoeligheid bepaald op basis van een validatie-set. De precisie geeft aan welk percentage van het geselecteerde subnetwerk de genen uit de validatie-set beslaan. De gevoeligheid geeft aan hoeveel genen van de valdiatie-set teruggevonden worden.

Deel III

Resultaten en Discussie

Hoofdstuk 7

Parameteranalyse

De gebruiker van PheNetic moet voor de uitvoering van het algoritme enkele parameters ingeven (hoofdstuk 5). Initieel is het belangrijk de invloed van deze parameters op de bekomen resultaten te karakteriseren. Dit hoofdstuk analyseert daarvoor de resultaten van elf verschillende parameterinstellingen (7.1) voor imidazool. Dit hoofdstuk vergelijkt de grootte (sectie 7.2.1), de samenstelling (sectie 7.2.2), de verklarende kracht (sectie 7.2.3) en de inhoud (sectie 7.2.4) van de bekomen netwerken. Sectie 7.3 bediscussieert de bekomen resultaten.

7.1 Experimenten

De input voor deze parameteranalyse zijn de imidazooldataset gedefinieerd in 4.3.1 en het *Salmonella* LT2-interactienetwerk met het eiwitinteractienetwerk gebaseerd op *E. coli* (sectie 4.1.1). We bepalen het regulatorisch netwerk door op bi-regulatorische paden (sectie 5.2.1) te zoeken. In hoofdstuk 5 werden de vier parameters aangegeven die de gebruiker van PheNetic zelf dient te definiëren:

- de padlengte (l) als maximale lengte om paden te zoeken (bi-directioneel)
- de n meest waarschijnlijke paden (n) per genenpaar te selecteren
- de kost (c) waarmee de score berekend wordt
- het aantal keer dat de optimalisatie herhaald wordt (r)

Om de grootte van de inputdataset te beperken en zo de analyse te versnellen wordt nog een extra parameter ingevoerd: de expressiefractie (e). Dit is het percentage meest differentieel geëxpresseerde genen dat als invoer voor PheNetic gebruikt wordt.

Initieel wordt van een standaard parameterinstelling vertrokken. Deze gebruikt vijf procent van de oorspronkelijke dataset (105 genen), zoekt paden met een maximale lengte van zes verbindingslijnen en selecteert hieruit de vijf beste paden per genenpaar, die vervolgens in een optimalisatie van tien herhalingen gescoord worden met een kost van één tienduizendste. Vanuit deze parameterinstelling wordt telkens een specifieke parameter gevarieerd om de invloed ervan te karakteriseren. Tabel 7.1 geeft de parameterinstelling per experiment weer.

Tabel 7.1. Overzicht experimenten parameteranalyse

experiment	e	l	n	c	r	experiment	e	l	n	c	r
standaard	0,05	3	5	10^{-4}	10	e = 0,01	0,01	3	5	10^{-4}	10
n = 3	0,05	3	3	10^{-4}	10	e = 0,10	0,10	3	5	10^{-4}	10
n = 10	0,05	3	10	10^{-4}	10	c = 1E-3	0,05	3	5	10^{-3}	10
n = 20	0,05	3	20	10^{-4}	10	c = 1E-5	0,05	3	5	10^{-5}	10
l = 2	0,05	2	5	10^{-4}	10	c = 1E-6	0,05	3	5	10^{-6}	10
l = 4	0,05	4	5	10^{-4}	10						

Legende: e expressiefractie, l padlengte, n aantal beste paden, c kost, r aantal optimalisatierondes.

7.2 Resultaten

Tabel 7.2 geeft een overzicht van de analyse van de geselecteerde netwerken, die in de volgende secties besproken wordt aan de hand van figuren. De term "startdata" wordt gebruikt voor de mogelijk te verklaren input gegeven het interactienetwerk en de paddefinitie.

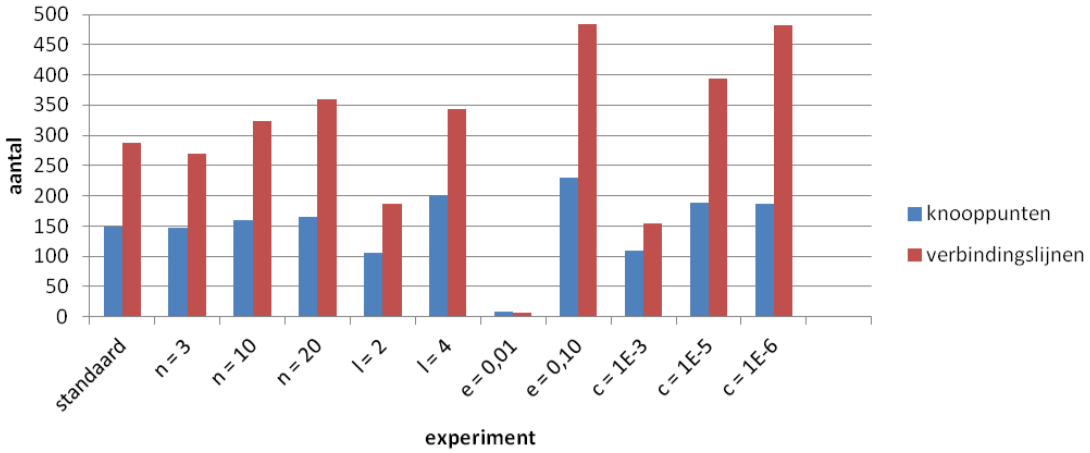
Tabel 7.2. Netwerkanalyse van de experimenten in de parameteranalyse: overzicht van (i) het aantal knooppunten, (ii) het aantal verbindingslijnen, waarvan het aantal eiwit-DNA-, eiwit-eiwit-, sigma-, sRNA-, fosforylatie-, metabolische en overige interacties en (iii) het percentage verklaarde input en startdata.

experiment	# KP	# VL	# eiwit-DNA	# eiwit-eiwit	# sigma	# sRNA	# fosforylatie	# metabool	# overig	% verklaarde input	% verklaarde startdata
standaard	149	288	157	106	8	12	4	1	0	55	97
n = 3	148	270	155	87	6	12	5	4	1	54	95
n = 10	159	324	160	136	8	9	5	6	0	55	97
n = 20	166	360	170	154	13	12	6	4	1	55	97
l = 2	106	186	135	37	7	6	1	0	0	54	97
l = 4	201	344	151	148	6	19	6	14	0	57	100
e = 0,01	8	6	6	0	0	0	0	0	0	24	44
e = 0,10	230	483	247	176	24	26	6	4	0	50	97
c = 1E-3	110	154	78	62	5	2	3	4	0	44	78
c = 1E-5	189	393	190	173	12	10	7	1	0	56	98
c = 1E-6	187	482	241	203	13	16	8	1	0	55	98

Legende: # aantal, KP knooppunten, VL verbindingslijnen, startdata de mogelijk te verklaren input gegeven het interactienetwerk en de paddefinitie.

7.2.1 Grootte van de bekomen netwerken

Ten eerste wordt de grootte – in enerzijds het aantal knooppunten en anderzijds het aantal verbindingslijnen – van de subnetwerken bekomen uit de verschillende experimenten, vergeleken (figuur 7.1). Op één na resulteren alle experimenten in meer verbindingslijnen – in de meeste gevallen zelfs dubbel zoveel – dan knooppunten. De netwerken resulterend uit parametervariaties die de expressiefractie verlagen, de padlengte verkorten of de kost verhogen zijn kleiner.

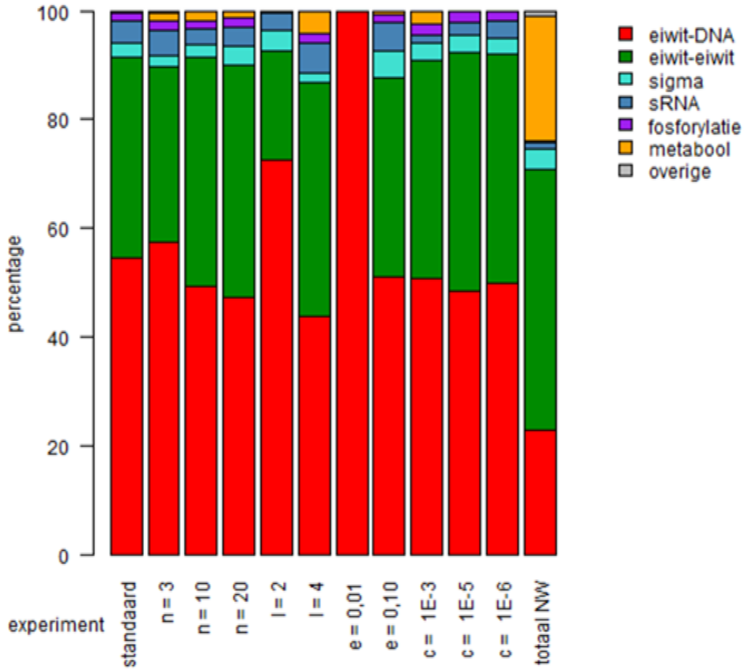


Figuur 7.1. Vergelijking van de netwerkgroottes bekomen in de parameteranalyse: voor elk experiment wordt het aantal knooppunten en verbindingslijnen aangegeven in een blauw respectievelijk rood gekleurde balk.

7.2.2 Samenstelling van de bekomen netwerken

Naast het verschil in aantal verbindingslijnen kan ook gekeken worden naar het type interacties dat deze beschrijven. Figuur 7.2 geeft voor elk experiment een strookdiagram dat de relatieve aantallen van de verschillende soorten interacties weergeeft. Ter vergelijking wordt ook voor het totale netwerk (zonder *rpo*D en *gro*EL) de samenstelling weergegeven.

Eiwit-DNA- en eiwit-eiwitinteracties zijn het sterkst aanwezig in de resultaten. Voor de eiwit-eiwitinteracties komt de relatieve aanwezigheid overeen met deze in het totale netwerk. Eiwit-DNA-interacties zijn daarentegen in verhouding meer vertegenwoordigd in de bekomen subnetwerken. Ook de verhouding van de metabole interacties in de resultaten strookt niet met deze in het totale netwerk: deze zijn ondervertegenwoordigd in de resultaten. Bovendien geven een verhoging van de expressiefractie, een verlenging van de padlengte en een verlaging van de kost aanleiding tot diversere interactietypes.

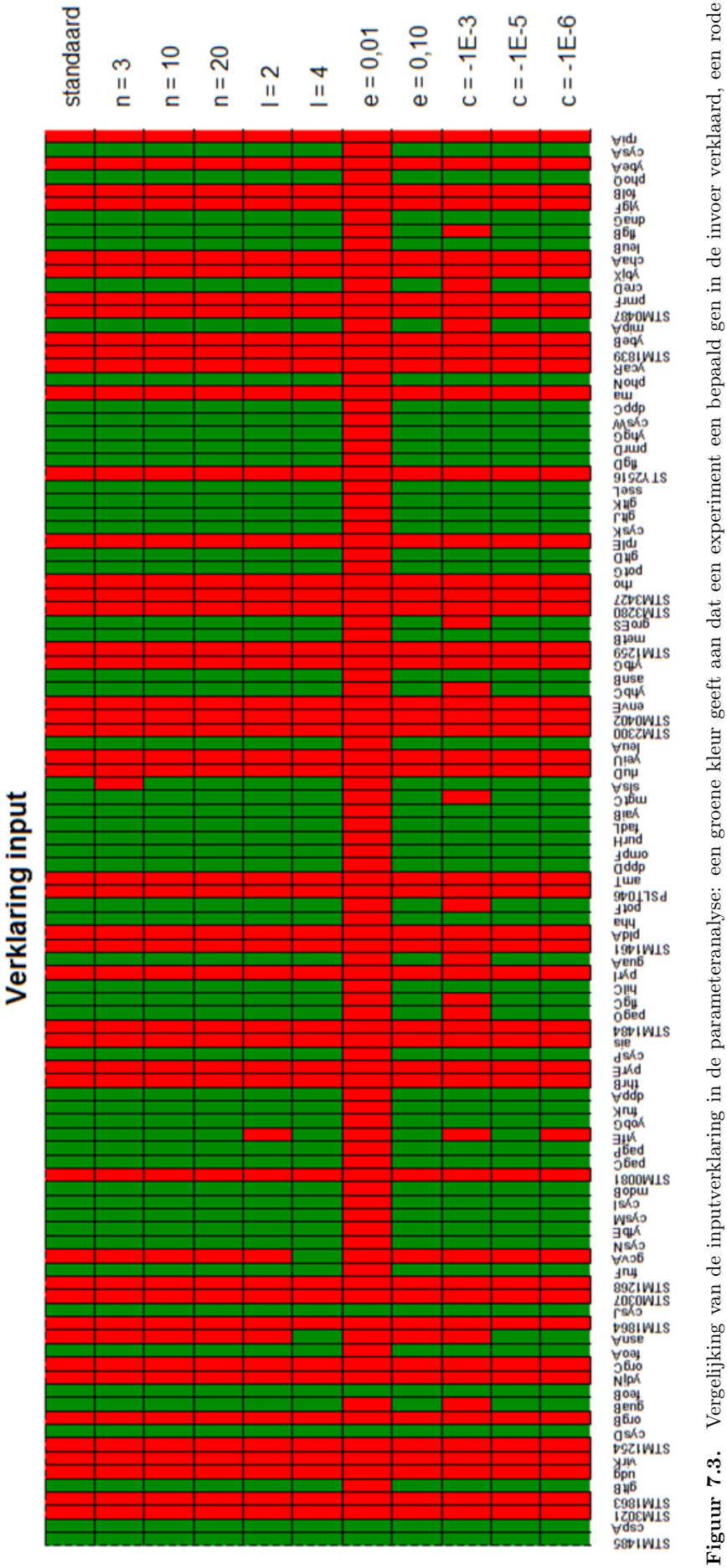


Figuur 7.2. Vergelijking van de interactietypes van de verbindingslijnen voor verschillende parameterinstellingen: eiwit-DNA-interacties in rood, eiwit-eiwitinteracties in groen, sigma-interacties in turkoois, sRNA-interacties in lichtblauw, fosforylaties in paars, metabole interacties in oranje en overige interactietypes in grijs. Als referentie wordt de samenstelling van het gebruikte interactienetwerk weergegeven.

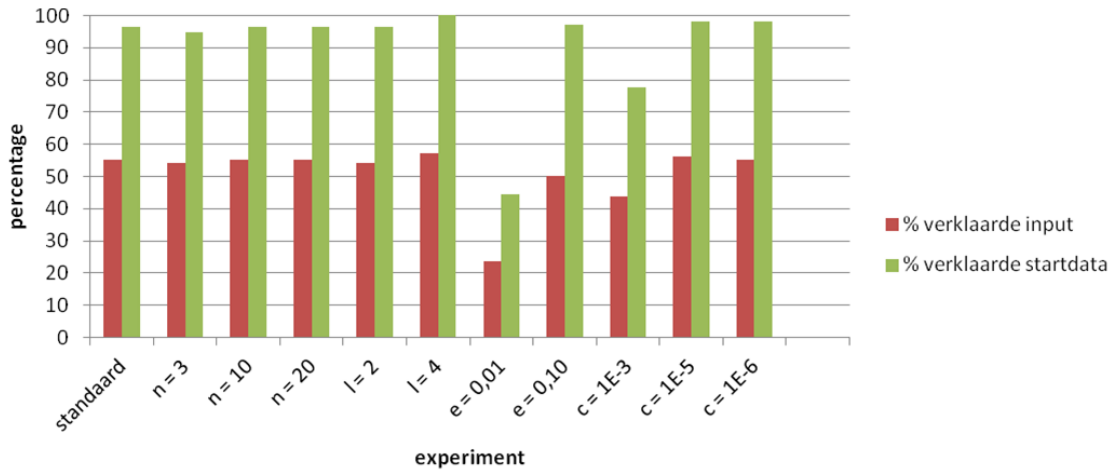
7.2.3 Verklarende kracht van de bekomen netwerken

Het doel van PheNetic is de gegeven input zoveel mogelijk te verklaren met een zo klein mogelijk subnetwerk. Figuur 7.4 geeft voor elke parametervariatie weer hoeveel van de input verklaard wordt. PheNetic kan iets meer dan de helft van de ingevoerde genen verklaren. Als we echter rekening houden met het feit dat sommige ingevoerde genen niet in het gebruikte interactienetwerk voorkomen of niet gevonden kunnen worden met de gebruikte paddefinitie, kan PheNetic het grootste deel van de input verklaren. Een lagere expressiefractie vindt minder ingevoerde genen terug, maar een hogere expressiefractie verklaart niet meer input/startdata. Eveneens leidt de verhoging van het aantal beste paden of een verlaging van de kost niet tot meer verklaarde genenparen.

Het leek ook interessant eens te kijken welke genen van de ingevoerde set aanwezig zijn in de oplossing. Figuur 7.3 daartoe voor elk verklaard gen een groene kleur toe per experiment en voor elk gen dat niet voorkomt in de oplossing een rode kleur. De weergegeven genen zijn deze met de vijf procent beste DE-waarden. Merk dus op dat de twee experimenten die de expressiefractie variëren eigenlijk vertrekken van een grotere of kleinere subset. De meeste experimenten zijn in staat om ongeveer dezelfde genen te verklaren. Bij strengere selectiecondities komen enkele genen niet meer terug in de oplossing. Bovendien is voor de verklaring van sommige genen een langere padlengte of lagere kost nodig vooraleer deze geselecteerd worden in de oplossing.



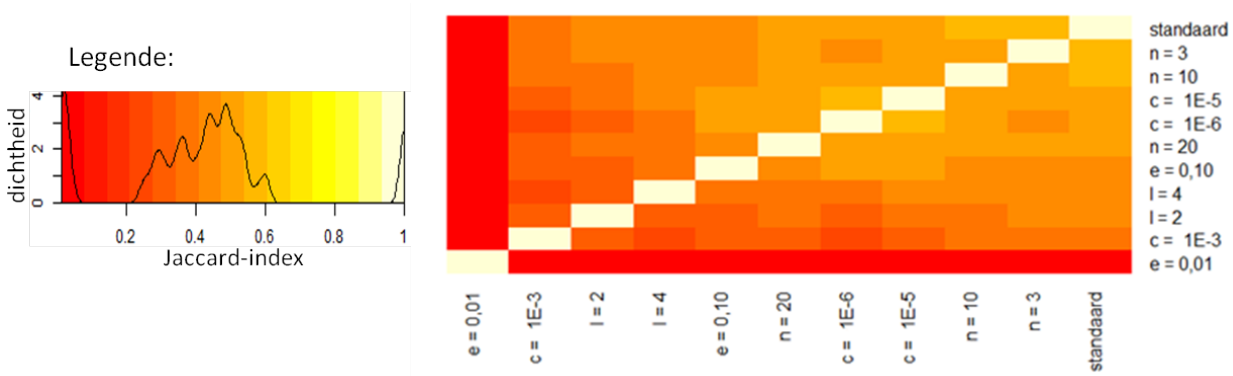
Figuur 7.3. Vergelijking van de inputverklaring in de parameteranalyse: een groene kleur geeft aan dat een experiment een bepaald gen in de invoer verklaard, een rode kleur geeft aan dat dit gen niet meer in de resultaten voorkomt.



Figuur 7.4. Vergelijking verklarende kracht van de experimenten in de parameteranalyse: voor elke parametervariatie wordt weergegeven hoeveel van de ingevoerde genen (rood) en van de startdata – de mogelijk te verklaren input gegeven het interactienetwerk en de paddefinitie – (groen) verklaard kunnen worden.

7.2.4 Vergelijking van de bekomen netwerken

Tenslotte kan de manier waarop de verschillende netwerken de gegeven DE-waarden verklaren, vergeleken worden. Dit gebeurt op basis van de Jaccard-index (sectie 6.1.2). Hogere waarden voor deze index wijzen op meer gelijkenis en worden in figuur 7.5 weergegeven in lichtere kleuren. De volgorde van de experimenten is eveneens gebaseerd op de graad van overeenkomst.



Figuur 7.5. Vergelijking van de netwerken bekomen in de parameteranalyse met behulp van de Jaccard-index: lichtere kleuren wijzen op meer gelijke resultaten.

De verandering van het aantal beste paden per genenpaar lijkt in eerste instantie de minste variatie te geven. Drastische veranderingen in deze parameter ($n = 5$ naar $n = 20$) leiden wel tot een meer verschillende oplossing. Verder is opmerkelijk dat een verlaging van de kost minder verandering geeft dan een verhoging van de kost. Analoog leidt een verlaging in de expressiefractie tot een meer verschillende oplossing dan een verhoging. Een verlaging van de expressiefractie, een verkorting van de padlengte en een verhoging van de kost leiden bovendien tot de grootste verschillen in de resulterende netwerken, niet alleen ten opzichte van de rest maar ook ten opzichte van elkaar.

7.3 Discussie

In deze sectie worden de observaties, gevonden in sectie 7.2, besproken en worden mogelijke verklaringen geformuleerd.

7.3.1 Grootte van de bekomen netwerken

Bepaalde parametervariaties zoals een verlaging van de expressiefractie, een verkorting van de padlengte en een verhoging van de kost, resulteren in kleinere subnetwerken. Deze parametervariaties leiden met andere woorden tot strengere selectiecondities ten opzichte van het standaardexperiment. Parametervariaties zoals een verhoging van de expressiefractie, een verlenging van de padlengte en een verlaging van de kost, stellen minder eisen voor selectie en zijn in staat grotere subnetwerken te selecteren. In het vervolg van deze discussie worden de termen "strengere selectiecondities" en "mildere selectiecondities" gebruikt om deze parametervariaties te bespreken ten opzichte van het standaardexperiment. Of grotere subnetwerken ook betere biologische inzichten geven, kan in dit stadium nog niet beslist worden. Hiervoor wordt een biologische validatie uitgewerkt in het volgende hoofdstuk.

7.3.2 Samenstelling van de bekomen netwerken

Mildere selectiecondities leiden tot het terugvinden van meer soorten interacties, zelfs deze – fosforylaties en sRNA-interacties – die ondervertegenwoordigd zijn in het totale netwerk. De afwijking van de verhoudingen in het totale netwerk is vooral te wijten aan de zoekstrategie. PheNetic wordt immers ingesteld om te zoeken naar bi-directionele regulatorische paden. Daar deze langs beide kanten moeten starten met een regulatorische interacties is het logisch dat eiwit-DNA, sRNA- en sigma-interacties sterker vertegenwoordigd zijn in de oplossing. Eiwit-DNA-interacties worden hierdoor het sterkst geselecteerd, metabolische interacties het zwakst.

7.3.3 Verklarende kracht van de bekomen netwerken

Indien de geselecteerde input kleiner is, is het logisch dat minder startdata verklaard kunnen worden. Als de genen niet in de input zitten zal PheNetic ze immers niet proberen te verklaren. Het gebruik van een hogere expressiefractie of een lagere kost zorgt echter eerder voor de selectie van meerdere parallelle paden zodat er niet noodzakelijk meer genen verklaard worden, maar vaak wel meer dan één verklaring per genenpaar beschikbaar is.

7.3.4 Vergelijking van de bekomen netwerken

De verandering van een bepaalde parameter leidt tot variatie in de bekomen resultaten. Veranderingen in expressiefractie of padlengte leiden tot meer variatie dan veranderingen in het aantal beste paden of de kost. Voor de expressiefractie ligt de verklaring voor de hand: een andere input leidt logischerwijs tot andere inzichten. De experimenten die de padlengte variëren, vertrekken weliswaar van dezelfde input voor de zoektocht naar paden, maar deze zoektocht verloopt anders zodat de invoer voor de optimalisatie-stap in PheNetic verschillend is. Deze redenering zou je ook kunnen doortrekken voor een verandering in het aantal beste paden, maar in dit geval blijkt de uiteindelijke variatie kleiner. Een mogelijke verklaring hiervoor is dat de meest waarschijnlijke paden cruciaal zijn in de optimalisatie en de minder waarschijnlijke paden in de top vijf en de top tien een te lage probabiliteit hebben om geselecteerd te worden. De grote verandering voor de optimalisatie van genenpaarnetwerken op basis van de twintig meest waarschijnlijke paden kan ook hierdoor verklaard worden. Hoe meer paden met een lage waarschijnlijkheid er opgenomen worden in de genenpaarnetwerken, hoe meer overlap er mogelijk is tussen de paden waardoor ze toch meegenomen worden in de oplossing. Bovendien resulteert het gebruik van mildere parameterinstellingen ten opzichte van een standaardexperiment in minder variatie dan strengere, althans in deze parameteranalyse. Waarschijnlijk zijn deze laatste in het algemeen te streng gekozen of in het geval van de expressiefractie van één procent gewoon te klein om te vergelijken.

7.4 Conclusie

De initiële schatting van de parameterinstellingen scoort goed in vergelijking met de parametervariaties hierop. Deze instelling selecteert een kleiner subnetwerk, maar kan nog steeds veel startdata verklaren. Bovendien vereist deze selectieconditie minder computerkracht dan mildere varianten die meer kunnen verklaren, zodat deze parameterinstellingen als basis gebruikt worden in de volgende experimenten. Om grotere subnetwerken – die mogelijks meer verklaren – te verkrijgen wordt best de kostparameter verlaagd. Een aanpassing van het aantal beste paden heeft pas zin vanaf meer dan twintig beste paden, omdat het dan mogelijk is andere combinaties van paden te vinden die beter scoren. Dit vereist echter veel meer computerkracht (meer dan 20 dagen) en wordt om die reden niet gebruikt in het vervolg van deze studie. Tenslotte wordt verwacht dat enkel een verhoging van de padlengte nuttig is om effectief meer input te verklaren, maar er is verder onderzoek nodig om dit te bevestigen.

Hoofdstuk 8

Validatie-analyse

De parameteranalyse in hoofdstuk 7 kon de invloed van verschillende instellingen van PheNetic karakteriseren, maar niet besluiten welke parameterinstellingen resulteren in goede biologische inzichten. De validatie-analyse in dit hoofdstuk verschaft meer inzicht in de biologische relevantie van de bekomen subnetwerken.

8.1 Experimenten

Deze validatie-studie past PheNetic toe op expressiedata bekomen na behandeling van *Salmonella* LT2 met mitomycine C. Om de bekomen subnetwerken biologisch te valideren, wordt getest of deze de gekende doelwitten van mitomycine C bevatten.

8.1.1 Input

Deze analyse start van de DE-waarden uit de mitomycine dataset van Frye *et al.* (2005), beschreven in sectie 4.2. De gebruikte component, mitomycine C, is een natuurlijk antibioticum, geïsoleerd uit *Streptomyces caespitosus*, en vormt 'cross-links' tussen de complementaire strengen van de DNA-dubbele helix. Deze werking is schadelijk voor bacteriën en induceert de DNA-herstel(SOS)-respons (Tomasz, 1995). De metingen in deze dataset gebeurden specifiek voor *Salmonella* LT2, zodat het interactienetwerk met de organisme-specifieke eiwit-eiwitinteracties uit sectie 4.1.1 gebruikt wordt.

8.1.2 Proefopzet

Deze validatie-analyse wordt zowel toegepast op experimenten die het gebruikte interactienetwerk opnieuw wegen op basis van een exponentiële graadverdeling (sectie 5.1.2.2) als op experimenten die de sigmoïdale graadverdeling (sectie 5.1.2.3) gebruiken. Voor elke herwegingsmethode worden gelijkaardige experimenten uitgevoerd, zoals vermeld in 8.1.

Gezien de computationele limieten wordt slechts een beperkt deel, de twee of drie percent meest differentieel geëxpresseerde genen, van de dataset gebruikt. Om een algemeen beeld te krijgen wordt initieel gezocht op simpele paden met een maximale padlengte van zes verbindingslijnen, overeenkomstig de resultaten van de parameteranalyse (hoofdstuk 7). Voor zoektochten op simpele paden

blijkt dit echter te lang, zodat overgeschakeld wordt op paden met maximaal vier verbindingslijnen om de grootte van de bekomen subnetwerken te beperken. Het aantal beste paden dat meegenomen wordt in de optimalisatierondes wordt niet gevarieerd, aangezien deze parameter het minste invloed had op de resultaten in de parameteranalyse. Voor de kostparameter wordt ook vertrokken van de parameteranalyse, maar deze moet nog sterk gevarieerd worden om goede resultaten te bekomen. Het aantal optimalisatierondes wordt dan weer bepaald door de computationele limieten, waardoor enkel interessant ogende experimenten voor meerdere rondes gelopen worden.

Tabel 8.1. Herwegingsmethode en parameters gebruikt in de validatie-studie

experiment	herweging	e	soort pad	l	n	c	r
exp_e002_sim_l4_n5_c5_r1	exponentieel	2 %	simpel	4	5	10^{-5}	1
exp_e002_sim_l4_n5_c10_r1	exponentieel	2 %	simpel	4	5	10^{-10}	1
exp_e002_sim_l4_n5_c20_r1	exponentieel	2 %	simpel	4	5	10^{-20}	1
exp_e002_sim_l6_n5_c20_r1	exponentieel	2 %	simpel	6	5	10^{-20}	1
exp_e002_sim_l6_n5_c20_r5	exponentieel	2 %	simpel	6	5	10^{-20}	5
exp_e003_reg_l6_n5_c20_r1	exponentieel	3 %	regulatorisch	6	5	10^{-20}	1
exp_e003_sim_l4_n5_c20_r1	exponentieel	3 %	simpel	4	5	10^{-20}	1
exp_e003_sim_l4_n5_c20_r5	exponentieel	3 %	simpel	4	5	10^{-20}	5
exp_e003_sim_l6_n5_c20_r1	exponentieel	3 %	simpel	6	5	10^{-20}	1
sig_e002_sim_l4_n5_c3_r1	sigmoïdaal	2 %	simpel	4	5	10^{-3}	1
sig_e002_sim_l4_n5_c4_r1	sigmoïdaal	2 %	simpel	4	5	10^{-4}	1
sig_e002_sim_l4_n5_c5_r1	sigmoïdaal	2 %	simpel	4	5	10^{-5}	1
sig_e002_sim_l4_n5_c5_r5	sigmoïdaal	2 %	simpel	4	5	10^{-5}	5
sig_e002_sim_l6_n5_c4_r1	sigmoïdaal	2 %	simpel	6	5	10^{-4}	1
sig_e003_reg_l6_n5_c5_r1	sigmoïdaal	3 %	regulatorisch	6	5	10^{-5}	1
sig_e003_sim_l4_n5_c4_r1	sigmoïdaal	3 %	simpel	4	5	10^{-4}	1
sig_e003_sim_l4_n5_c5_r1	sigmoïdaal	3 %	simpel	4	5	10^{-5}	1
sig_e003_sim_l4_n5_c5_r5	sigmoïdaal	3 %	simpel	4	5	10^{-5}	5

Legende: e expressiefractie, l padlengte, n aantal beste paden, c kost, r aantal optimalisatierondes, exponentieel herwegingsmethode beschreven in §5.1.2.2, sigmoïdaal herwegingsmethode beschreven in §5.1.2.3.

8.1.3 Validatie-set

Om de gevoeligheid en de precisie (sectie 6.3) van de bekomen subnetwerken te bepalen, wordt een validatie-set opgesteld. Daar mitomycine C in bacteriën de SOS-respons induceert (Tomasz, 1995), bevat deze de SOS-responsgenen van *Salmonella* LT2 (Smith *et al.*, 1991; Benson *et al.*, 2000). De voorgestelde set in tabel 8.2 is gebaseerd op de indeling in de BioCyc databank voor *Salmonella* LT2. Eveneens wordt voor elk validatie-gen de DE-waarde uit de gebruikte dataset en het voorkomen in het netwerk en de input weergegeven.

Tabel 8.2. Validatie-set voor behandeling van *Salmonella* LT2 met mitomycine C

genaam	STM-nummer	beschrijving*	DE-waarde	in netwerk?	in input?
lexA	STM4237	repressor LexA	0,558	✓	
recA	STM2829	recombinase A	2,853	✓	✓
recF	STM3836	recombinatie-eiwit F	-0,108	✓	
recQ	STM3958	ATP-afhankelijk DNA-helicase RecQ	0,456	✓	
ruvA	STM1895	'Holliday junction' DNA-helicase RuvA	0,175	✓	
ruvB	STM1894	'Holliday junction' DNA-helicase RuvB	2,032	✓	
STM1309	STM1309	endonuclease voor herstel van nucleotide-excisies	0,469	✓	
sulA	STM1071	SOS-inhibitor van de celdeling	4,015	✓	✓
umuC	STM1997**	DNA-polymerase V subeenheid UmuC	-0,285	✓	
umuD	STM1998	DNA-polymerase V subeenheid UmuD	1,021	✓	
uvrA	STM4254	excinuclease ABC subeenheid A	1,824	✓	
uvrB	STM0798	excinuclease ABC subeenheid B	0,490	✓	
uvrC	STM1946	excinuclease ABC subeenheid C	-0,035	✓	
uvrD	STM3951	DNA-afhankelijk helicase II	2,228	✓	

Legende: *volgens de BioCyc databank voor *Salmonella* LT2, **ook deel op plasmide (PSLT054) maar de BioCyc databank rekent dit niet mee in de SOS-respons.

8.2 Resultaten

De numerieke resultaten van deze validatie-analyse zijn terug te vinden in tabel 8.3. Hier valt af te lezen dat de precisie (sectie 6.3) van de gebruikte subnetwerkselectiemethoden zeer laag is: er worden veel meer niet-relevante genen teruggevonden. Anderzijds zijn alle p-waardes (sectie 6.1.1) significant voor een significantieniveau van vijf percent, wat betekent dat de teruggevonden oplossing specifiek is voor het geanalyseerde probleem.

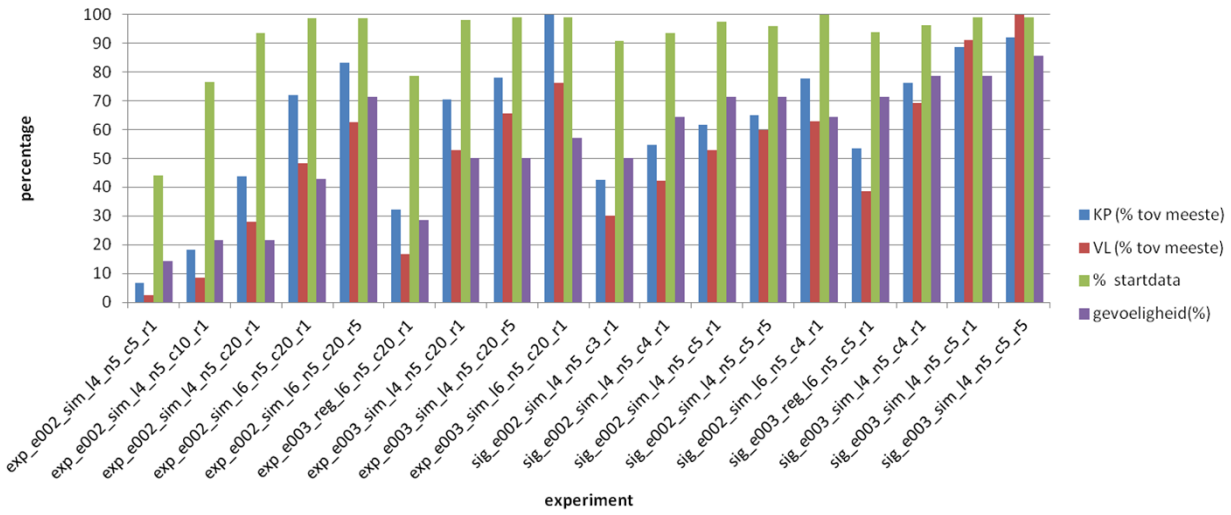
De verschillende experimenten worden met elkaar vergeleken in figuur 8.1. De groottes van de netwerken worden hiervoor relatief ten opzichte van elkaar uitgedrukt door de verhouding weer te geven met het hoogste aantal teruggevonden knooppunten en verbindingslijnen. De verklaarde input wordt uitgedrukt in percentages van het mogelijk te verklaren aantal genen, namelijk deze die ook in het gebruikte interactienetwerk voorkomen.

In eerste instantie lijkt het verklaren van meer startdata de gevoeligheid te beïnvloeden. Er zijn echter veel experimenten die ongeveer evenveel van de startdata verklaren, maar toch verschillen in gevoeligheid. De grootte van de netwerken lijkt in deze gevallen ook een sterke rol te spelen.

Tabel 8.3. Numerieke resultaten netwerkanalyse van de experimenten in de validatie-studie

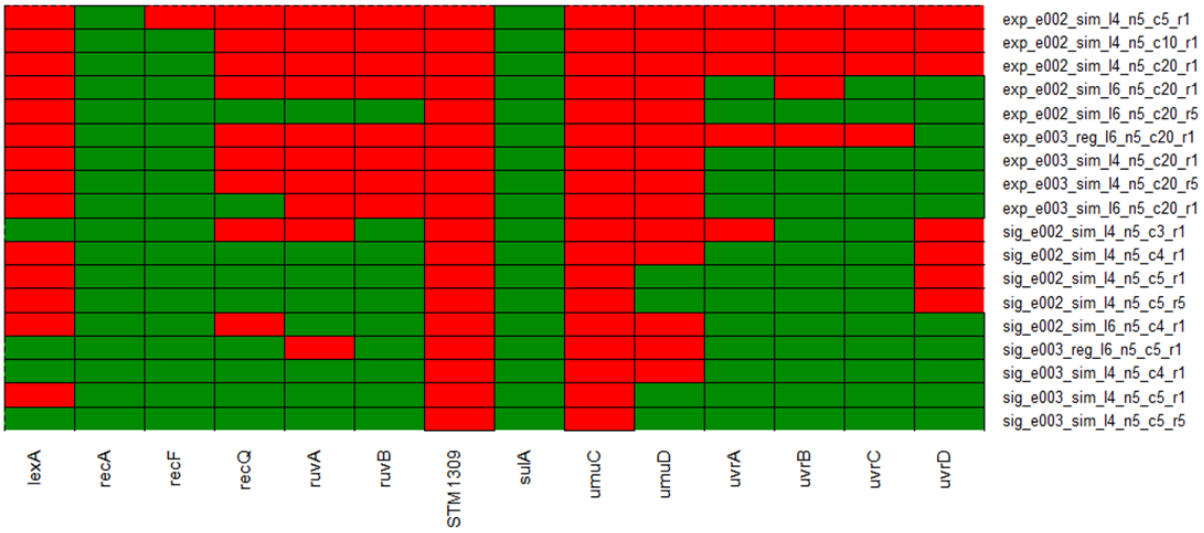
experiment	# KP	# VL	% verklaarde input	% verklaarde startdata	precisie * (%)	gevoeligheid * (%)	p-waarde **
exp_e002_sim_l4_n5_c5_r1	64	71	40	44	3,1	14,3	$3,5 * 10^{-3}$
exp_e002_sim_l4_n5_c10_r1	173	260	69	77	1,7	21,4	$8,8 * 10^{-3}$
exp_e002_sim_l4_n5_c20_r1	413	854	85	94	0,7	21,4	$1,4 * 10^{-1}$
exp_e002_sim_l6_n5_c20_r1	680	1475	89	99	0,9	42,9	$3,3 * 10^{-2}$
exp_e002_sim_l6_n5_c20_r5	786	1911	89	99	1,3	71,4	$1,3 * 10^{-4}$
exp_e003_reg_l6_n5_c20_r1	304	511	69	79	1,3	28,6	$1,3 * 10^{-2}$
exp_e003_sim_l4_n5_c20_r1	666	1617	86	98	1,1	50,0	$7,6 * 10^{-3}$
exp_e003_sim_l4_n5_c20_r5	736	2010	87	99	1,0	50,0	$1,4 * 10^{-2}$
exp_e003_sim_l6_n5_c20_r1	943	2330	87	99	0,8	57,1	$1,9 * 10^{-2}$
sig_e002_sim_l4_n5_c3_r1	400	920	82	91	1,8	50,0	$2,3 * 10^{-4}$
sig_e002_sim_l4_n5_c4_r1	515	1294	85	94	1,7	64,3	$2,1 * 10^{-5}$
sig_e002_sim_l4_n5_c5_r1	582	1619	88	97	1,7	71,4	$5,9 * 10^{-6}$
sig_e002_sim_l4_n5_c5_r5	613	1835	87	96	1,6	71,4	$1,0 * 10^{-5}$
sig_e002_sim_l6_n5_c4_r1	733	1921	91	100	1,2	64,3	$5,1 * 10^{-4}$
sig_e003_reg_l6_n5_c5_r1	503	1185	82	94	2,0	71,4	$1,3 * 10^{-6}$
sig_e003_sim_l4_n5_c4_r1	718	2124	84	96	1,5	78,6	$4,2 * 10^{-6}$
sig_e003_sim_l4_n5_c5_r1	836	2787	87	99	1,3	78,6	$2,4 * 10^{-5}$
sig_e003_sim_l4_n5_c5_r5	868	3060	87	99	1,4	85,7	$2,4 * 10^{-6}$

Legende: # aantal, KP knooppunten, VL verbindingslijnen, startdata mogelijk te verklaren input, * berekend volgens §6.3, ** hypergeometrische test (§6.1.1).



Figuur 8.1. Netwerkanalyse van de experimenten in de validatie-studie: voor elk experiment wordt het aantal knooppunten (KP) en verbindingslijnen (VL) relatief ten opzichte van het hoogste aantal knooppunten respectievelijk verbindingslijnen dat is teruggevonden, weergegeven. De term "startdata" verwijst naar de ingevoerde genen die verklaard kunnen worden.

Figuur 8.2 geeft vervolgens aan welke genen van de validatie-set genen wel (groen) en niet (rood) teruggevonden worden. De genen *rec*A en *sul*A worden in elke proefopzet teruggevonden, de genen *umuC* en STM1309 daarentegen in geen enkele. Op één na kunnen ook alle experimenten *rec*F terugvinden, maar voor de andere genen zijn de resultaten verdeeld. Wel lijken de experimenten die gebruik maken van de sigmoïdale herwegingsmethode beter te scoren.



Figuur 8.2. Teruggevonden validatie-genen per experiment: voor elk experiment krijgen de genen in de validatie-set een groene respectievelijk rode kleur naargelang ze wel of niet teruggevonden worden in het betreffende experiment.

8.3 Discussie

De lage precisiewaarden zijn te verklaren door de te kleine validatie-set (slechts veertien genen) en de te grote geselecteerde subnetwerken. Ze kunnen eventueel wel gebruikt worden om resultaten met een gelijke gevoeligheid te vergelijken, maar individueel zijn ze allemaal even slecht. Hoewel grotere netwerken sowieso een hogere kans hebben om de gewenste genen te selecteren, zijn de resultaten wel specifieker dan de selectie van willekeurige netwerken (significante p-waarden).

De verklaring van meer startdata kan inderdaad gerelateerd zijn met de gevoeligheid. Als je meer differentieel geëxpresseerde genen kan verklaren is de kans ook groter dat je meer verantwoordelijke genen terug vindt. Dit is in de huidige opzet echter niet voldoende, zodat grotere subnetwerken en daarvoor ook sterkere computerkracht nodig zijn.

Het gebruikte interactienetwerk bevat geen rechtstreekse connecties tussen de genen in de validatie-set en de differentieel geëxpresseerde genen, hetgeen ook een verklaring kan zijn voor de nood aan grote netwerken. De waargenomen differentiële expressie kan het resultaat zijn van meerdere 'downstream' gelegen processen en de gebruikte proefopzet gaat misschien niet diep genoeg terug

('downstream'). Experimenten die paden langer dan vier verbindingslijnen onderzoeken zouden hier een uitkomst kunnen bieden.

Deze validatie-analyse maakt het wel mogelijk de gebruikte herwegingsmethoden te valideren en geeft aan dat een sigmoïdale herwegingsmethode duidelijk meer overeenkomt met de biologische realiteit. Een mogelijke verklaring hiervoor is dat de probabiliteiten na een herweging op basis van de exponentiële graadverdeling zo laag worden dat PheNetic ze op voorhand al negeert en zo vaak niet eens aan vijf beste resultaten per genenpaar komt.

8.4 Conclusie

PheNetic kan genen (en reactiewegen) die biologisch relevant zijn terugvinden, maar de gevoeligheid is afhankelijk van de gebruikte parameterinstellingen. Bovendien vindt PheNetic ook veel andere genen terug die misschien inderdaad biologisch niet relevant zijn. Ze kunnen echter ook voor de signaaloverdracht tussen de oorzaak en het effect zorgen of nieuwe nog niet geëxploreerde inzichten leveren.

Tot slot leunt een sigmoïdale herweging van de 'hubs' in een interactienetwerk dichter aan bij de biologische realiteit dan een herweging op basis van de exponentiële graadverdeling.

Hoofdstuk 9

Gevalstudie: *Salmonella* biofilms

De vorige hoofdstukken analyseerden en valideerden de subnetwerkselectiemethode die PheNetic hanteert. De volgende vraag is of PheNetic ook toepasbaar is voor biologische gevalstudies. Zijn de resulterende netwerken met andere woorden logisch te verklaren en leveren ze ook een mogelijke oplossing voor de probleemstelling? Daartoe proberen we in deze gevalstudie met behulp van het subnetwerkselectiemechanisme PheNetic de werking van imidazolen op de vorming van *Salmonella* biofilms te ontrafelen. Meer specifiek gaan we op zoek naar regulatoren betrokken in biofilmvorming die uitgeschakeld worden door imidazolen.

De verwijdering van biofilms vormt een hedendaagse problematiek. Deze organisaties van bacteriën zijn enerzijds moeilijk te verwijderen door hun beschermende matrix. Anderzijds zorgt de grote variatie van bacteriën in deze organisaties voor resistentie-ontwikkeling tegen reeds bestaande antibiofilmagentia. De goedkeuring van nieuwe antibiofilmagentia kampt bovendien met lange proceduretijden en de kans op resistentie-ontwikkeling blijft bestaan. Daardoor is er steeds meer interesse voor de combinatie van reeds bestaande antibiofilmagentia op een manier die de kans op resistentie-ontwikkeling laag houdt. Inzicht in de werking van de reactiewegen betrokken in biofilmvorming is hiervoor cruciaal. Dit inzicht kan bekomen worden door de uitvoering van hoge-doorvoerexperimenten, zodat het een interessante toepassing is voor een netwerk-gebaseerde analyse.

9.1 Experimenten

Het doel van de experimenten in deze gevalstudie is een inzicht te krijgen in het werkingsmechanisme van imidazolen en de bijhorende cellulaire respons. Hiertoe voerden onderzoekers van de 'Salmonella & Probiotica' groep hoge-doorvoerexperimenten uit die (i) een imidazoolbehandeling vergeleken met een controleconditie en (ii) de imidazoolbehandeling van een gevoelige en een ongevoelige stam vergeleken. Dit resulteerde in drie datasets, beschreven in sectie 4.3. Om de werking van deze imidazolen op de vorming van *Salmonella* biofilms te ontrafelen, worden vijf experimenten uitgevoerd die variëren in de keuze van de dataset, het gebruikte netwerk en de specifieke parameters. Deze sectie geeft de redenering achter de keuze van de verschillende proefopzetten:

Controle versus imidazoolbehandeling (dataset in sectie 4.3.1)

- imidazool_1: controle versus behandeling met component 1 – ronde 1
- imidazool_2: controle versus behandeling met component 1 – ronde 2
- imidazoline: controle versus behandeling met component 2

Gevoelig versus ongevoelig voor imidazoolbehandeling (dataset in sectie 4.3.2)

- sensitiviteit_1: gevoelig versus ongevoelig voor behandeling met component 3 – ronde 1
- sensitiviteit_2: gevoelig versus ongevoelig voor behandeling met component 3 – ronde 2

9.1.1 Input

Om te beginnen verschilt de input waarvan elk experiment vertrekt in (i) de gebruikte actieve component, (ii) de genomen expressiefractie, (iii) het aantal genen dat hierna overblijft in de startdata, (iv) het gebruikte netwerk en (v) het percentage van de startdatagenen dat hierin voorkomt. Tabel 9.1 geeft voor elke experiment een overzicht van de gebruikte startdata.

Tabel 9.1. Eigenschappen van de input gebruikt voor de experimenten in de gevalstudie

experiment	component[1]	e	# genen	netwerk	genen in netwerk
Controle versus imidazoolbehandeling[2]					
imidazool_1	1	5 %	105	ortholoog	77 %
imidazool_2	1	fc $>$ 2	152	specifiek	74 %
imidazoline	2	5 %	81	ortholoog	64 %
Gevoelig versus ongevoelig voor imidazoolbehandeling[3]					
sensitiviteit_1	3	5 %	228	specifiek	43 %
sensitiviteit_2	3 [4]	5 %	136	specifiek	100 %

Legende: e expressiefractie, # aantal, fc 'foldchange', ortholoog het netwerk beschreven in §4.1.1, specifiek het netwerk beschreven in §4.1.2, [1]zie figuur 4.1, [2]dataset in §4.3.1, [3]dataset in §4.3.2, [4]gefilterd op genen in het netwerk.

Omwille van computationele limieten wordt slechts een deel van elke dataset gebruikt, namelijk de vijf percent meest differentieel geëxpresseerde genen, gebaseerd op de resultaten van de parameteranalyse (hoofdstuk 7). Op vraag van de onderzoekers zelf wordt in experiment imidazool_2 deze startdataset uitgebreid tot alle genen met een 'foldchange' (of 1/'foldchange') hoger dan twee.

Initieel wordt het *Salmonella* LT2-interactienetwerk met het eiwitinteractienetwerk afgeleid van *E. coli* (sectie 4.1.1) gebruikt. In de loop van de studie wordt echter overgeschakeld op een nieuw interactienetwerk voor *Salmonella* LT2 met organisme-specifieke eiwit-eiwitinteracties van STRING

(sectie 4.1.2). Er is immers meer inzicht in dit netwerk, omdat dit ook getest werd in de validatie-analyse (hoofdstuk 8). Daar het experiment voor de behandeling met component 1 toch herhaald wordt met een hogere expressiefractie, wordt ineens ook het netwerk aangepast.

Let wel dat de gebruikte interactienetwerken niet volledig zijn. Dit wordt geïllustreerd doordat niet alle genen in de input erin voorkomen. Voor de gevoeligheidsexperimenten komt zelfs minder dan de helft van de genen terug in het netwerk (experiment sensitiviteit_1). Dit kan zijn omdat een netwerk specifiek opgesteld voor *Salmonella* LT2 gebruikt wordt en dit experiment andere *Salmonella*-stammen gebruikt. Daarom wordt voor experiment sensitiviteit_2 de startdataset gefilterd op genen die in het gebruikte interactienetwerk voorkomen.

9.1.2 Proefopzet

Op aanvraag van de labo-onderzoekers gaat deze gevalstudie op zoek naar regulatoren betrokken in *Salmonella* biofilmvorming, wat betekent dat PheNetic in eerste instantie op bi-directionele regulatorische paden moet zoeken. De gebruikte parameters worden initieel gekozen op basis van de parameteranalyse (hoofdstuk 7). Tabel 9.2 geeft voor elk experiment de gebruikte herwegingsmethode en parameterinstellingen, die in de rest van deze sectie gemotiveerd worden.

Tabel 9.2. Herwegingsmethode en parameters gebruikt voor de experimenten in de gevalstudie

experiment	herweging	soort pad	l	n	c	r	freq
Controle versus imidazoolbehandeling							
imidazool_1	exponentieel	bi-regulatorisch	6	5	10^{-4}	10	$\geq 0{,}3$
imidazool_2	sigmoïdaal	bi-regulatorisch	6	5	10^{-4}	20	$\geq 0{,}9$
imidazool_2–simpel	sigmoïdaal	simpel	4	5	10^{-1}	5	$\geq 0{,}9$
imidazoline	exponentieel	bi-regulatorisch	6	5	10^{-4}	10	$\geq 0{,}3$
Gevoelig versus ongevoelig voor imidazoolbehandeling							
sensitiviteit_1	sigmoïdaal	bi-regulatorisch	6	5	10^{-4}	1	$\geq 0{,}9$
sensitiviteit_2	sigmoïdaal	bi-regulatorisch	6	5	10^{-5}	20	$\geq 0{,}9$
sensitiviteit_2–simpel	sigmoïdaal	simpel	4	5	10^{-1}	5	$\geq 0{,}9$

Legende: l padlengte, n aantal beste paden, c kost, r aantal optimalisatierondes, freq frequentiegrenswaarde voor betrouwbare interacties –simpel gezocht op simpele paden voor de betreffende dataset, exponentieel herweging beschreven in §5.1.2.2, sigmoïdaal hetweging beschreven in §5.1.2.3.

9.1.2.1 Herwegingsmethode

Tijdens de studie vindt een overschakeling plaats van een herweging op basis van een exponentiële graadverdeling (sectie 5.1.2.2) naar één op basis van een sigmoïdale graadverdeling (sectie 5.1.2.3). Daar de validatie-analyse (hoofdstuk 8) aantoont dat deze laatste betere resultaten levert, wordt deze gebruikt in de latere experimenten.

9.1.2.2 Parameterinstellingen

De parameterinstellingen voor experimenten imidazool_1 en imidazoline zijn gebaseerd op de resultaten van de parameteranalyse (hoofdstuk 7), omdat deze specifiek is uitgevoerd voor deze data. PheNetic probeert hierbij de ingevoerde genen te verbinden via bi-directionele regulatorische paden met een maximale lengte van zes verbindingslijnen. De vijf beste paden per genenpaar worden in tien optimalisatieronden gescoord met een kost van één tienduizendste.

Ook experiment imidazool_2 wordt uitgevoerd met de paddefinitie, de padlengte, het aantal beste paden en de kost gedefinieerd in de parameteranalyse. Het aantal optimalisatieronden wordt echter verdubbeld als een eerste poging om meer van de ingevoerde genen te verklaren. Een tweede poging om meer ingevoerde genen te verklaren gebeurt door een zoektocht op simpele paden (experiment imidazool_2–simpel) waarvan de maximale lengte – gebaseerd op de validatie-analyse (hoofdstuk 8) – wordt ingesteld op vier verbindingslijnen. De resulterende vijf beste paden per genenpaar worden in een optimalisatie van vijf herhalingen gescoord met een kost van één tiende.

Hoewel de parameteranalyse uitgevoerd is voor de experimenten die een behandelingsconditie vergelijken met een controleconditie, wordt deze ook uitgetest voor de gevoeligheidsanalyse (sensitiviteit_1). Dergelijke test start met één optimalisatieronde om de duur van het experiment te beperken. Naast het feit dat deze test aantoont dat de gebruikte dataset niet specifiek genoeg is voor het interactienetwerk van *Salmonella* LT2 (tabel 9.1), blijkt ook de kost gedefinieerd in de parameteranalyse te hoog. Experiment sensitiviteit_2 wordt daarom uitgevoerd met een kost van één honderdduizendste. Analoog als in de tweede ronde van de imidazoolexperimenten wordt getracht meer ingevoerde genen te verklaren door enerzijds het aantal optimalisatieronden te verhogen en anderzijds te zoeken op simpele paden (experiment sensitiviteit_2–simpel).

9.1.2.3 Betrouwbare interacties

De uitvoering van meerdere optimalisatieronden maakt het mogelijk de betrouwbaarheid van de gevonden verbindingslijnen in te schatten. Verbindingslijnen die PheNetic in minstens 90% van de optimalisatieronden selecteert, worden betrouwbaar geacht. Na toepassing van deze grenswaarde op experimenten imidazool_1 en imidazoline schiet er echter niets meer over, zodat de grenswaarde hier verlaagd wordt tot 0,3.

9.2 Resultaten

Daar deze studie zich voornamelijk focust op subnetwerkselectie worden de bekomen resultaten eerst besproken in termen van netwerken, zoals in hoofdstukken 7 en 8. Sectie 9.2.1 neemt deze bespreking voor zijn rekening. Vervolgens bespreekt sectie 9.2.2 beknopt enkele biologische bevindingen die resulteren uit deze netwerken.

9.2.1 Netwerkanalyse

De bekomen netwerken worden geanalyseerd op (i) de grootte in termen van aantal knooppunten en verbindingslijnen, (ii) het percentage ingevoerde genen dat verklaard wordt, (iii) het percentage verklaarde startdata – dit zijn de ingevoerde genen die verklaard kunnen worden op basis van het gebruikte netwerk en de opgegeven paddefinitie –, (iv) het aantal aanwezige regulatoren, (v) het gemiddeld aantal doelwitten per gevonden regulator en (vi) gemiddeld aantal ingevoerde genen dat per regulator verklaard wordt. Deze eigenschappen worden gekozen omdat ze interessant lijken met het oog op de biologische analyse en de resultaten zijn weergegeven in tabel 9.3.

De verklaring van de totale input is zeer beperkt, namelijk minder dan de helft van de genen in de totale inputset wordt verklaard. Indien we echter rekening houden met het aantal genen dat PheNetic maximaal kan verklaren op basis van het gebruikte interactienetwerk en de opgegeven paddefinitie (bi-regulatorisch versus simpel) is het resultaat gunstiger.

Tabel 9.3. Netwerkanalyse van de experimenten in de gevalstudie

experiment	# KP	# VL	% verklaarde input	% verklaarde startdata	# regulatoren	# genen* (gemiddeld)	# verklaard* (gemiddeld)
Controle versus imidazoolbehandeling							
imidazool_1	105	143	46	81	37	2,76	1,89
imidazool_2	96	159	41	75	30	4,80	3,53
imidazool_2–simpel	211	327	51	69	6	2,17	0,83
imidazool_2–totaal	258	477	63	90	32	4,72	3,31
imidazoline	42	50	22	49	20	2,15	1,70
Gevoelig versus ongevoelig voor imidazoolbehandeling							
sensitiviteit_1	124	266	28	98	47	4,96	3,49
sensitiviteit_2	170	385	68	100	59	5,71	4,31
sensitiviteit_2–simpel	245	414	64	64	13	1,92	1,15
sensitiviteit_2–totaal	346	774	90	100	62	5,58	4,10

Legende: # aantal, KP knooppunten, VL verbindingslijnen, *per regulator, –simpel gezocht op simpele paden, –totaal combinatie van regulatorisch en simpel netwerk.

Voor puur regulatorische netwerken – de experimenten zonder extra achtervoegsel – geldt dat meer geselecteerde knooppunten resulteert in een hoger aantal regulatoren. De verhouding van regulatoren en geselecteerde knooppunten is bovendien bij benadering één derde. Zoekacties op simpele paden zijn in staat meer van de ingevoerde genen te verklaren, maar ze selecteren – zoals te verwachten – een lagere fractie regulatoren. Om een totaalbeeld te krijgen van enerzijds het regulatorisch netwerk en anderzijds het netwerk dat de functie weerspiegelt (simpele paden) is het mogelijk beide samen te voegen tot een netwerk dat zowel veel input verklaart als veel regulatoren levert.

Er is niet meteen een verband tussen de netwerkgrootte en het gemiddeld aantal doelwitten per regulator. Wel valt op dat experimenten die gebruik maken van de exponentiële herwegingsmethode een lager gemiddeld aantal doelwitten bevatten per regulator. De doorzoeking van interactienetwerken herwogen op basis van de sigmoïdale graadverdeling van de eindknooppunten, op regulatorische paden met een maximale padlengte van zes verbindingslijnen resulteert in gemiddeld vijf doelwitten per regulator.

9.2.2 Biologische analyse

De netwerkanalyse toonde reeds aan dat er effectief regulatoren teruggevonden kunnen worden. Deze sectie bestudeert de biologische relevantie van de teruggevonden regulatoren, door te kijken of ze (i) verband houden met de functionele groepen die in de startdata voorkomen en (ii) verrijkt zijn in de bekomen oplossingen.

9.2.2.1 Exploratie startdata

Om een idee te krijgen van de invloed van imidazool op *Salmonella* LT2 worden de ingevoerde genen gevisualiseerd op het gebruikte interactienetwerk. Op basis van verrijkingsanalyses worden de aanwezige genen manueel ingedeeld in negen functionele groepen:

- sulfaat-geassocieerde processen (S): genen betrokken in het metabolisme van sulfaat
- nitraat-geassocieerde processen (N): genen betrokken in het metabolisme van nitraat
- koolstofmetabolisme (C): genen rechtstreeks in de energiecyclus en de productie van koolhydraten
- lipide-geassocieerde processen (L): genen betrokken in het lipidemetabolisme, inclusief stressrespons en pathogenese
- aminozuurmetabolisme (AZ): genen die een rol spelen in het metabolisme van aminozuren

- transport (T): genen betrokken in transport van onder andere aminozuren, koolhydraten, eiwitten en ionen
- (ribo-)nucleotide metabolisme ((r)Nt): genen betrokken in het metabolisme van purine en pyrimidine (ribo-)nucleotiden
- celmetabolisme (CM): genen die een rol spelen in transcriptie, translatie, RNA-processing en celdeling.
- celbeweging (CB): alle genen die te maken hebben met flagelbeweging en cilia-vorming

Indien genen in meerdere functionele groepen ondergebracht kunnen worden, gebeurt de indeling op basis van de functies van de interagerende genen. Tabel 9.4 geeft per experiment de bijhorende figuren in bijlage B weer met de verrijkte functionele groepen.

Tabel 9.4. Biologische exploratie van de startdata gebruikt in de gevalstudie

experiment	figuur	KP	VL	S	N	C	L	AZ	T	(r)Nt	CM	CB
Controle versus imidazoolbehandeling												
imidazool_1	B.1	83	48	↑↑	/	/	↑	↑	↑	↓↓	↑	↓↓
imidazool_2	B.2	112	67	↑↑	/	/	↑	↑	↑	↓↓	↑	↓↓
imidazoline	B.3	52	22	/	/	/	↑	↑↓	↑↓	↓↓	↑↑	↓↓
Gevoelig versus ongevoelig voor imidazoolbehandeling												
sensitiviteit_1	B.4	99	53	/	↑↓	↑↓	↑↓	↑↑	↑↓	↑↑	↑↑	↓↓
sensitiviteit_2	B.5	136	75	/	↑↓	↑↓	↑↓	↓	↑↓	↑↑	↑	↓↓

Legende: KP aantal knooppunten (genen), VL aantal verbindingslijnen (interacties), S sulfaat-geassocieerde processen, N nitraat-geassocieerde processen, C koolstofmetabolisme, L lipide-geassocieerde processen, AZ aminozuurmetabolisme, T transport, (r)Nt (ribo)nucleotide-metabolisme, CM celmetabolisme, CB celbeweging, ↑↑ alle betrokken genen zijn opgereguleerd, ↑ de meeste betrokken genen zijn opgereguleerd, ↑↓ de betrokken genen zijn zowel op- als neergereguleerd, ↓ de meeste genen zijn neergereguleerd, ↓↓ alle genen zijn neergerguleerd.

In het algemeen zijn het (ribo)nucleotide metabolisme en celbeweging minder aanwezig in de behandelde stammen. Sulfaat-, nitraat-, koolstof- en lipide-geassocieerde processen (inclusief de antibioticumrespons), transport, celmetabolisme en aminozuurmetabolisme komen anderzijds meer tot expressie.

9.2.2.2 Regulatoren

Het is de bedoeling om nu met behulp van een netwerk-gebaseerde analyse de regulatoren waarop imidazool inwerkt en die dus waarschijnlijk een rol spelen in biofilmvorming, te identificeren. De regulatoren bekomen uit experimenten imidazool_1, imidazool_2, imidazoline, sensitiviteit_1 en sen-

sitiviteit_2 worden onderworpen aan een verrijkingsanalyse (hoofdstuk 6). Tabel 9.5 geeft een overzicht van enkele teruggevonden regulatoren.

Hoewel de startdata niet altijd overeenkomen, komen wel ongeveer dezelfde regulatoren voor in beide soorten experimenten. Zowel algemene als specifieke regulatoren, voornamelijk betrokken in celbeweging of lipide-geassocieerde processen worden teruggevonden. Bijna de helft van de teruggevonden regulatoren is echter nog niet geclassificeerd in een functionele groep, ze hebben wel allemaal een rol in de regulatie van transcriptie.

Tabel 9.5. Overzicht van een aantal interessante regulatoren in de gevalstudie

regulator	proces	imidazool_1	imidazool_2	imidazoline	sensitiviteit_1	sensitiviteit_2
*cys*B	sulfaat-geassocieerd	++	++	−−	−−	−−
*csg*D	celbeweging	+−	+	−−	+−	++
*dsr*A	ongeclassificeerd	++	−−	−−	++	++
*fli*Z	celbeweging	−−	+	−−	++	++
hns	globaal	+	+	+−	++	++
mig-14	ongeclassificeerd	++	++	++	+	+
*pho*P	lipide-geassocieerd	+	++	+−	+−	+−
*rpo*S	stress-respons	−−	−−	−−	++	++
*rst*A	ongeclassificeerd	+	++	++	++	++
*sly*A	lipide-geassocieerd	+	++	+	−−	−−
ycfQ	ongeclassificeerd	−−	++	−−	++	++
*yif*A	celbeweging	++	++	−−	++	++

Legende: ++ p-waarde < 0,0001; + 0,0001 < p-waarde < 0,01; +− 0,01 < p-waarde < 0,50; − p-waarde > 0,50; −− niet in oplossing aanwezig.

9.3 Discussie

Deze discussie probeert logische verklaringen te vinden voor de resulterende subnetwerken (sectie 9.3.1) en bespreekt beknopt de bevindingen voor de gegeven probleemstelling (sectie 9.3.2).

9.3.1 Netwerkanalyse

De oorzaak voor het (relatief) lage percentage ingevoerde genen dat verklaard kan worden, moet waarschijnlijk bij de gebruikte interactienetwerken gezocht worden. Als deze niet de juiste info bevatten, kan PheNetic deze uiteraard ook niet terugvinden. Merk wel op dat de selectie van een betrouwbare fractie van het netwerk leidt tot de verwijdering van interacties en genen die wel in

de oorspronkelijke subnetwerkoplossingen zaten en misschien wel in staat waren meer ingevoerde genen te verklaren.

Netwerken die resulteren uit regulatorische subnetwerkselectiemethoden lijken goed in staat een groot aantal van de ingevoerde genen te verklaren met behulp van regulatoren. Simpele subnetwerkselectiemethoden zijn duidelijk niet bruikbaar voor het zoeken naar regulatoren. De verklaring hiervoor ligt in (i) de gebruikte paddefinitie die niet afdwingt dat er naar regulatoren wordt gezocht en (ii) de samenstelling van het netwerk, dat verhoudingsgewijs minder regulatorische interacties bevat (figuur 7.2). Zoektochten naar simpele paden zijn in deze context dus enkel nuttig als ze samengevoegd worden met regulatorische netwerken in een poging om meer ingevoerde genen te verklaren.

Het is bovendien logisch dat interactienetwerken herwogen op basis van de exponentiële graadverdeling van de eindknooppunten minder doelwitten per regulator bevatten, omdat in deze proefopzet knooppunten die relatief weinig interacties aangaan reeds een zeer lage probabiliteit krijgen. De interacties die wel aanwezig zijn in de selectiemethoden die een sigmoïdale graadverdeling bevatten, kunnen daardoor waarschijnlijk gewoon niet geselecteerd worden.

9.3.2 Biologische analyse

Deze sectie bespreekt enkele teruggevonden regulatoren door (i) te kijken of het logisch is dat ze geselecteerd worden volgens het subnetwerkselectieprincipe en (ii) beknopt de biologische relevantie te duiden. Het feit dat PheNetic dezelfde regulatoren terugvindt voor startdata die elkaar tegenspreekt (op- versus neergereguleerd) is inherent aan de methode, omdat het algoritme enkel rekening houdt met de absolute DE-waarden. Als doelwitten van bepaalde regulatoren niet voorkomen in de startdata kunnen deze regulatoren uiteraard niet teruggevonden worden.

9.3.2.1 *cys*B

*cys*B is een transcriptionele regulator centraal in sulfaat-geassocieerde processen zoals cysteïne biosynthese en zwavelmetabolisme (Ren *et al.*, 2005; Lee *et al.*, 2007; Caspi *et al.*, 2012). Deze regulator wordt enkel teruggevonden in experimenten imidazool_1 en imidazool_2 omdat ook enkel deze experimenten andere sulfaat-geassocieerde genen – en minstens de helft van alle doelwitten voor *cys*B – bevatten in hun startdata (figuur B.1 en B.2). Ren *et al.* (2005) en Lee *et al.* (2007) opperden reeds dat *cys*B een invloed heeft op biofilmvorming in *E. coli*-stammen. Een mogelijke verklaring voor de afwezigheid van deze genen in de gevoeligheidsexperimenten is het tijdstip van de staalname, daar de genexpressie ervan varieert gedurende biofilmvorming (Domka *et al.*, 2007;

Lee *et al.*, 2007).

9.3.2.2 *ycf*Q

*ycf*Q codeert voor een transcriptionele repressor van *ycf*R (Deng *et al.*, 2011; Caspi *et al.*, 2012), een gen dat voorkomt in de startdata van experimenten imidazool_2, sensitiviteit_1 en sensitiviteit_2 (figuur B.2, B.4 en B.3). De DE-waarde van *ycf*R hoort echter niet bij de hoogste vijf procent voor de overige experimenten, waardoor *ycf*Q hier niet als regulator teruggevonden wordt. Het gebruikte interactienetwerk bevat enkel *ycf*R als doelwit voor *ycf*Q, hetgeen de zeer lage p-waarden verklaart. Bovendien kan deze vondst interessant zijn, daar *ycf*R mogelijks een regulator is voor biofilmvorming (Lee *et al.*, 2007; Zhang *et al.*, 2007; Deng *et al.*, 2011).

9.3.2.3 Celbeweging

Een aantal teruggevonden regulatoren zijn geassocieerd met celbeweging (Caspi *et al.*, 2012) zoals (i) *csg*D voor de biosynthese van curli, (ii) *fli*Z voor de biosynthese van flagellen en (iii) *yif*A coderend voor HdfR, een transcriptionele regulator voor de biosynthese van flagellen (Ko & Park, 2000). Deze worden in het algemeen teruggevonden omdat een aantal genen betrokken bij flagel- en curlibiosythese sterk differentieel geëxpresseerd zijn in de gebruikte datasets. Bovendien spelen deze regulatoren een rol bij de keuze tussen een flagel-gebaseerde vrijlevende levensstijl of curli-gebaseerde adhesie en biofilmvorming (Prigent-Combaret *et al.*, 2000; Pesavento & Hengge, 2012).

9.3.2.4 *pho*P

*pho*P is de cytoplasmatische responsregulator van het PhoPQ twee-componentsysteem in *Salmonella* (Caspi *et al.*, 2012; Steenackers *et al.*, 2012). Deze regulator wordt in alle experimenten teruggevonden omdat de overeenkomstige startdata veel doelwitten van *pho*P bevat (figuren B.1 – B.5). Dat is goed, want deze regulator speelt inderdaad een rol in biofilmvorming (Steenackers *et al.*, 2012). De regulatoren *mig*–14 en *sly*A staan hiermee in verband, daar ze interacties aangaan met *pho*P en *pho*P-gereguleerde genen (Brodsky *et al.*, 2002; Spory *et al.*, 2002). *sly*A wordt echter niet teruggevonden in de gevoeligheidsexperimenten.

9.3.2.5 *rpo*S

*rpo*S codeert voor de sigma-factor σ^S verantwoordelijk voor de regulatie van transcriptie tijdens de algemene stress-respons en de stationaire fase (Steenackers *et al.*, 2012). Deze regulator speelt ook een belangrijke rol bij *Salmonella*-biofilmvorming, maar wordt enkel in de gevoeligheidsexperimenten teruggevonden. Een mogelijke verklaring hiervoor is het tijdstip van de staalname dat

verschilt tussen beide soorten experimenten: vroeg versus laat in de exponentiële fase. De "partner-in-crime" *crl* wordt echter nergens teruggevonden omdat de gebruikte interactienetwerken enkel de interactie *crl*–*rpo*N bevatten en *rpo*N niet in de startdata voorkomt. PheNetic zal dus nooit *crl* opgeven voor het geformuleerde probleem.

9.3.2.6 Connecties

De hierboven besproken genen (of hun genproducten) gaan ook interacties aan met andere regulatoren – waarvan sommige reeds aangehaald werden in de vorige secties – en met elkaar. Deze sectie bespreekt nog andere teruggevonden regulatoren die interacties aangaan met meerdere eerder besproken genen.

Zo heeft *dsr*A, een klein RNA-molecule, invloed op *rpo*S en *csg*D (Beisel & Storz, 2010; Mika & Hengge, 2014). *rst*A, de responsregulator van het RstAB twee-compnonentsysteem, wordt gereguleerd door *pho*P en heeft invloed op de expressie van *rpo*S en *csg*D (Steenackers *et al.*, 2012). Zelfs de zeer globale regulator *hns* wordt teruggevonden. Dit gen codeert voor een histon-achtig nucleoïde-structurerend eiwit (H-NS) betrokken in de directe en indirecte regulatie van veel niet-verwante genen (Steenackers *et al.*, 2012).

PheNetic vindt deze regulatoren terug omdat ze verbindingen maken tussen de verschillende reactiewegen betrokken in *Salmonella* biofilmvorming en dus indirect heel veel ingevoerde genen kunnen verklaren. Of deze ook een rol spelen in de inwerking van imidazolen is nog niet duidelijk, maar is eventueel stof voor verder onderzoek.

9.4 Conclusie

PheNetic kan regulatoren die mogelijk een rol spelen in *Salmonella* biofilmvorming identificeren. Deze gevalstudie bevestigt de rol van regulatoren als *csg*D, *pho*P en *rpo*S in *Salmonella* biofilmvorming. Daarnaast tonen de bekomen subnetwerken de mogelijke invloed van een imidazoolbehandeling op biologische functies als celbeweging en lipidenmetabolisme. Bijkomend suggereren ze ook de potentiële rol van de regulatoren *cys*B en *ycf*Q in *Salmonella* biofilmvorming. De bekomen resultaten waren een rechtstreekse aanleiding voor bijkomend 'wet-lab' onderzoek wat de toepasbaarheid van PheNetic onderstreept.

Algemene discussie en toekomstvisie

Op het einde van hoofdstukken 7, 8 en 9 worden de bekomen resultaten per experiment reeds geïnterpreteerd en bediscussieerd. Dit leidt tot de conclusie dat PheNetic in staat is om, op basis van genenlijsten, beknopte en biologisch relevante subnetwerken te selecteren uit biologische interactienetwerken voor de bestudeerde problemen. Deze algemene discussie haalt enkele interessante ontdekkingen nogmaals aan en suggereert verbeteringen naar de toekomst toe.

Om te beginnen kunnen we de gebruikte interactienetwerken in vraag stellen. Deze kunnen duidelijk uitgebreid worden, omdat niet alle genen uit de gebruikte datasets erin voorkomen. Of dit moet gebeuren door specifieker in te gaan op *Salmonella* LT2 of door sterk bestudeerde orthologe organismen in rekening te brengen, is niet duidelijk. Hiervoor kan een concrete vergelijking van de twee gebruikte eiwitinteractienetwerken interessant zijn.

Een tweede punt van kritiek is de verwijdering van de 'hubs' *gro*EL en *rpo*D uit de gebruikte interactienetwerken, waardoor weliswaar veel irrelevante maar eventueel ook relevante kennis verloren gaat. Het is misschien interessant om eens te testen of de sigmoïdale wegingsmethode deze ook voldoende afstraft en – indien dit niet het geval is – te checken of andere 'hubs' ook in aanmerking komen voor verwijdering ter verbetering van de resultaten. Bovendien selecteren zoektochten zonder opgegeven paddefinitie ("simpele paden") te grote subnetwerken om specifiek te zijn, zodat de gebruikte wegingsmethoden ook op dit vlak nog aanpassingen vereisen. Zoektochten op (bi-)regulatorische paden lossen dit probleem gedeeltelijk op, maar sluiten minder aan bij de realiteit omdat een organisme niet enkel uit regulatorische interacties voorkomt. Ze zijn echter wel een goede benadering om regulatorische mechanismen te ontrafelen.

Ten derde enkele opmerkingen met betrekking tot de parameterinstellingen voor PheNetic. Enerzijds bevestigen de padlengten gebruikt in deze studie, de 'small world property' van biologische netwerken (Alm & Arkin, 2003; Mason & Verwoerd, 2008): padlengten van vier tot zes verbindingslijnen zijn voldoende om relevante informatie te vinden. Anderzijds is dit tegenstrijdig met de waarneming dat langere padlengten meer input kunnen verklaren (parameteranalyse, hoofdstuk 7). Let wel dat de resulterende netwerken sterk verschillen, hetgeen misschien al een verklaring is voor deze tegenstrijdigheid. Vervolgens kan de gebruikte expressiefractie verhoogd worden om meer informatie terug te vinden. Dit heeft echter enkel nut als het gebruikte netwerk en de opgegeven paddefinitie de extra ingevoerde genen ook kunnen verklaren. Bovendien is het aangeraden

meerdere optimalisatieronden te lopen, daar de huidige optimalisatieprocedure begint van een willekeurig startpunt. Door vervolgens enkel rekening te houden met frequent geselecteerde interacties, heeft dit artefact van de methode geen invloed op de resultaten. Tenslotte lijkt in deze studie de selectie van meer beste paden in de genenparennetwerken geen invloed te hebben op het resultaat. Omwille van computationele redenen kon deze stelling slechts tot twintig beste paden getest worden, maar met de komst van snellere technieken lijkt deze hypothese zeker het testen waard.

Merk ook op dat deze studie enkel de netwerken resulterend uit een combinatie van "paden zoeken" en optimalisatie bestudeert. Om meer inzicht te krijgen in de optimalisatieprocedure, wordt beter vertrokken vanuit de gevonden paden. Aangezien de padprobabiliteiten het product zijn van de probabiliteiten van de opbouwende verbindingslijnen, is de kans immers reëel dat kortere paden hogere probabiliteiten bevatten. De hypothese is echter dat PheNetic deze niet in de uiteindelijke oplossing selecteert omdat ze te weinig input verklaren of te weinig overlap met ander paden kennen. In deze context kan het zijn dat een optimalisatie gebaseerd op meer dan twintig beste paden per genenpaar meer kan verklaren, omdat sterker overlappende paden die lagere probabiliteiten bevatten nu toch geselecteerd kunnen worden.

Tot slot moet de gebruiker altijd in het achterhoofd houden dat PheNetic slechts een mogelijke oplossing voorspelt. Het is in geen geval de enige mogelijke of werkelijke oplossing, omdat er te veel niveaus van onzekerheid aanwezig zijn. Ten eerste is de opstelling van de gebruikte interactienetwerken gebaseerd op wereldwijd beschikbare data, die weliswaar opgezuiverd werd maar eigenlijk reeds een eerste voorspelling maakt van de werkelijkheid. Ten tweede wordt het hieruit bekomen netwerk doorzocht met data uit 'wet-lab' experimenten, zodat ook altijd omgevingsfactoren een rol spelen en de resultaten kunnen beïnvloeden. En ten derde maakt PheNetic zelf assumpties, al dan niet opgegeven door de gebruiker, om een mogelijke oplossing voor de gegeven probleemstelling te bekomen. Het is dus met andere woorden een tool om de eerste inzichten te verwerven in een bepaalde probleemstelling.

Algemeen besluit

Deze studie toont aan dat PheNetic toepasbaar is voor de analyse van expressiedata bekomen uit hoge-doorvoerexperimenten voor *Salmonella* LT2. Dit gebeurt door (i) een karakterisering van het gedrag van het algoritme (hoofdstuk 7), (ii) een biologische validatie van de bekomen resultaten (hoofdstuk 8) en (iii) een praktische toepassing (hoofdstuk 9). Uit de bekomen resultaten kunnen we besluiten dat PheNetic in staat is om, op basis van genenlijsten, beknopte en biologisch relevante subnetwerken te selecteren uit biologische interactienetwerken.

Tijdens de studie kwamen echter ook enkele belangrijke punten voor verbetering naar boven. Ten eerste is de methode computationeel zeer intensief, wat drastische limieten stelt aan de te doorzoeken data. Ten tweede moet er verder inzicht worden verworven in de wegingsmethode van de netwerken. De biologische validatieset, voorgesteld in deze studie, geeft wel de mogelijkheid om een objectieve analyse van deze netwerkwegingen te doen.

Verder zijn er geen algemene parameterinstellingen van het algoritme en moet de gebruiker van PheNetic deze aanpassen aan zijn specifieke probleemstelling. Deze studie geeft wel aan dat paden met een padlengte van vier tot zes verbindingslijnen voldoende zijn om biologisch relevante subnetwerken te selecteren wat overeenkomt met de 'small world property' van biologische netwerken (Alm & Arkin, 2003; Mason & Verwoerd, 2008).

Bijkomend moet benadrukt worden dat PheNetic een subnetwerk selecteert vanuit publieke data in combinatie met genenlijsten bekomen uit 'wet-lab' experimenten. Hierdoor zijn er vele onzekerheden en is het belangrijk dat de bekomen resultaten niet als absoluut juist worden aangenomen, maar als een voorstel van het biologisch mechanisme achter het fenotype dat onderzocht wordt.

Bibliografie

Akramifar, S. & Ghassem-Sani, G. (2010). Fast forward planning by guided enforced hill climbing. *Engineering Applications of Artificial Intelligence*, 23(8): 1327–1339.

Albayrak, M. & Allahverdi, N. (2011). Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. *Expert Systems with Applications*, 38(3): 1313–1320.

Alm, E. & Arkin, A. P. (2003). Biological networks. *Current Opinion in Structural Biology*, 13: 193–202.

Alon, U. (2003). Biological networks: the tinkerer as an engineer. *Science*, 301(5641): 1866–1867.

Babu, M. M., Luscombe, N. M., Aravind, L., Gerstein, M. & Teichmann, S. A. (2004). Structure and evolution of transcriptional regulatory networks. *Current opinion in structural biology*, 14(3): 283–291.

Bader, D. & Madduri, K. (2006). Designing multithreaded algorithms for breadth-first search and st-connectivity on the Cray MTA-2. *Proceedings of the 35th International Conference on Parallel Processing (ICPP-2006)*, Washington DC, Washington, USA: 523–530.

Bailly-Bechet, M., Braunstein, A. & Zecchina, R. (2009). Computational methods in systems biology. chapter A Prize-Collecting Steiner Tree Approach for Transduction Network Inference, pp. 83–95. Springer, Berlin, Germany.

Barabási, A. L. & Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2): 101–113.

Basha, O., Tirman, S., Eluk, A. & Yeger-Lotem, E. (2013). ResponseNet2.0: revealing signaling and regulatory pathways connecting your proteins and genes – now with human data. *Nucleic Acids Research*, 41(W1): W198–W203.

Beisel, C. L. & Storz, G. (2010). Base pairing small RNAs and their roles in global regulatory networks. *Federation of European Microbiological Societies Microbiology Reviews*, 34(5): 866–882.

Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B (Methodological)*, 57(1): 289–300.

Benson, N. R., Wong, R. M. Y. & McClelland, M. (2000). Analysis of the SOS response in Salmonella enterica serovar Typhimurium using RNA fingerprinting by arbitrarily primed PCR. *Journal of bacteriology*, 182(12): 3490–3497.

Bhalla, U. S. & Iyengar, R. (1999). Emergent properties of networks of biological signaling pathways. *Science*, 283(5400): 381–387.

Bonetta, L. (2010). Interactome under construction. *Nature*, 468(7325): 851–854.

Boyan, J. & Moore, A. W. (2001). Learning evaluation functions to improve optimization by local search. *Journal of Machine Learning Research*, 1: 77–112.

Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Lara, G. G., Oezcimen, A., Rocca-Serra, P. & Sansone, S. A. (2003). ArrayExpress – a public repository for microarray gene expression data at the EBI. *Nucleic acids research*, 31(1): 68–71.

Brem, R. B., Yvert, G., Clinton, R. & Kruglyak, L. (2002). Genetic dissection of transcriptional regulation in budding yeast. *Science*, 296(5568): 752–755.

Brodsky, I. E., Ernst, R. K., Miller, S. I. & Falkow, S. (2002). mig-14 is a Salmonella gene that plays a role in bacterial resistance to antimicrobial peptides. *Journal of bacteriology*, 184(12): 3203–3213.

Burke, E. K. & Kendall, G. (2014). Search methodologies. chapter Introduction, pp. 1–17. Springer, New York City, New York, USA.

Cadoli, M. & Donini, F. M. (1997). A survey on knowledge compilation. *Artificial Intelligence Communications*, 10: 137–150.

Caspi, R., Altman, T., Dreher, K., Fulcher, C. A., Subhraveti, P., Keseler, I. M., Kothari, A., Krummenacker, M., Latendresse, M., Mueller, L. A., Ong, Q., Paley, S., Pujar, A., Shearer, A. G., Travers, M., Weerasinghe, D., Zhang, P. & Karp, P. D. (2012). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic acids research*, 40(D1): D742–D753.

Cherry, J. M., Hong, E. L., Amundsen, C., Balakrishnan, R., Binkley, G., Chan, E. T., Christie, K. R., Costanzo, M. C., Dwight, S. S., Engel, S. R., Fisk, D. G., Hirschman, J. E., Hitz, B. C., Karra, K., Krieger, C. J., Miyasato, S. R., Nash, R. S., Park, J., Skrzypek, M. S., Simison, M., Weng, S. & Wong, E. D. (2012). Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic Acids Research*, 40(D1): D700–D705.

Clark, M., Kim, Y., Kruschwitz, U., Song, D., Albakour, D., Dignum, S., Beresi, U. C., Fasli, M. & Roeck, A. D. (2012). Automatically structuring domain knowledge from text: an overview of current research. *Information Processing & Management*, 48(3): 552–568.

Cloots, L., De Maeyer, D. & Marchal, K. (2014). Springer handbook of bio-/neuroinformatics. chapter Path Finding in Biological Networks to Interpret Functional Data, pp. 289–309. Springer, Berlin, Germany.

Cloots, L. & Marchal, K. (2011). Network-based functional modeling of genomics, transcriptomics and metabolism in bacteria. *Current Opinion in Microbiolgy*, 14: 599–607.

Cohen, K. B. & Hunter, L. (2008). Getting started in text mining. *Public Library of Science Computational Biology*, 4(1): e20.

Cormen, T., Leiserson, C., Rivest, R. & Stein, C. (2001). Introduction to algorithms. The Massachusetts Institute of Technology Press, Cambridge, Massachusetts, USA. 1191 p.

Darwiche, A. (2009). Modelling and reasoning with Bayesian networks. chapter Compiling Bayesian Networks, pp. 287–312. Cambridge University Press, Cambridge, New York, USA.

Darwiche, A. & Marquis, P. (2002). A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17: 229–264.

Daudin, J. J., Picard, F. & Robin, S. (2008). A mixture model for random graphs. *Statistics and Computing*, 18(2): 173–183.

De Maeyer, D., Renkens, J., Cloots, L., De Raedt, L. & Marchal, K. (2013). PheNetic: network-based interpretation of unstructured gene lists in E. coli. *Molecular BioSystems*, 9: 1594–1603.

De Smet, R. & Marchal, K. (2010). Advantages and limitations of current network inference methods. *Nature Reviews Microbiology*, 8: 717–729.

DeLuca, T. F., Wu, I.-H., Pu, J., Monaghan, T., Peshkin, L., Singh, S. & Wall, D. P. (2006). Roundup: a multi-genome repository of orthologs and evolutionary distances. *Bioinformatics*, 22(16): 2044–2046.

Deng, K., Wang, S., Rui, X., Zhang, W. & Tortorello, M. L. (2011). Functional analysis of ycfR and ycfQ in Escherichia coli O157: H7 linked to outbreaks of illness associated with fresh produce. *Applied and environmental microbiology*, 77(12): 3952–3959.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1): 269–271.

Domka, J., Lee, J., Bansal, T. & Wood, T. K. (2007). Temporal gene-expression in Escherichia coli K-12 biofilms. *Environmental microbiology*, 9(2): 332–346.

Donaldson, I., Martin, J., De Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G. D., Michalickova, K., Pawson, T. & Hogue, C. W. (2003). PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BioMed Central Bioinformatics*, 4(1): 11.

Downsland, K. A. (2014). Search methodologies. chapter Classical techniques, pp. 19–65. Springer, New York City, New York, USA.

Doyle, P. G. & Snell, J. L. (2000). Random walks and electric networks. Free Software Foundation, Boston, Massachusetts, USA. 120 p.

Dupont, P., Callut, J., Dooms, G., Monette, J. & Deville, Y. (2006). Relevant subgraph extraction from random walks in a graph. *Université catholique de Louvain Research reports*, 7.

Edgar, R., Domrachev, M. & Lash, A. E. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1): 207–210.

Emmert-Streib, F. & Dehmer, M. (2011). Networks for systems biology: conceptual connection of data and function. *Systems Biology, Institution of Engineering and Technology*, 5(3): 185–207.

Faust, K., Dupont, P., Callut, J. & Van Helden, J. (2010). Pathway discovery in metabolic networks by subgraph extraction. *Bioinformatics*, 26(9): 1211–1218.

Felner, A., Modenhauer, C., Sturtevant, N. & Schaeffer, J. (2010). Single-frontier bidirectional search. *Proceedings of the Third Annual Symposium on Combinatorial Search (SOCS-10)*, Atlanta, Georgia, USA: 59–64.

Fitch, W. M. (1970). Distinguishing homologous from analogous proteins. *Systematic Biology*, 19(2): 99–113.

Franceschini, A., Szklarczyk, D., Frankild, S., Kuhn, M., Simonovic, M., Roth, A., Lin, J., Minguez, P., Bork, P., von Mering, C. & Jensen, L. J. (2013). STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(D1): D808–D815.

Frye, J. G., Porwollik, S., Blackmer, F., Cheng, P. & McClelland, M. (2005). Host gene expression changes and DNA amplification during temperate phage induction. *Journal of bacteriology*, 187(4): 1485–1492.

Gibbs, D., Baratt, A., Baric, R., Kawaoka, Y., Smith, R., Orwoll, E., Katze, M. & McWeeney, S. (2013). Protein co-expression network analysis (ProCoNA). *Journal of Clinical Bioinformatics*, 3(1): 11.

Gilad, Y., Rifkin, S. A. & Pritchard, J. K. (2008). Revealing the architecture of gene regulation: the promise of eQTL studies. *Trends in genetics*, 24(8): 408–415.

Glover, F. (1989). Tabu search – part I. *Operations Research Society of America's Journal on Computing*, 1(3): 190–206.

Gnad, F., Gunawardena, J. & Mann, M. (2011). PHOSIDA 2011: the posttranslational modification database. *Nucleic Acids Research*, 39(suppl 1): D253–D260.

Goldberg, D. E. & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2): 95–99.

Golomb, S. W. & Baumert, L. D. (1965). Backtrack programming. *Journal of the Association for Computing Machinery*, 12(4): 516–524.

Granville, V., Krivanek, M. & Rasson, J. P. (1994). Simulated annealing: a proof of convergence. *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence*, 16(6): 652–656.

Guimerà, R. & Amaral, L. A. N. (2005). Functional cartography of complex metabolic networks. *Nature*, 433(7028): 895–900.

Hirschhorn, J. N. & Daly, M. J. (2005). Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(2): 95–108.

Hopcroft, J. & Tarjan, R. (1973). Algorithm 447: efficient algorithms for graph manipulation. *Commununications of the Association for Computing Machinery*, 16(6): 372–378.

Huang, C. Y., Sun, C. T. & Lin, H. C. (2005). Influence of local information on social simulations in small-world network models. *Journal of Artificial Societies and Social Simulation*, 8(4): 8.

Huang, J., Liu, Y., Zhang, W., Yu, H. & Han, J. D. J. (2011). eResponseNet: a package prioritizing candidate disease genes through cellular pathways. *Bioinformatics*, 27(16): 2319–2320.

Huang, S. C. & Fraenkel, E. (2009). Integrating proteomic, transcriptional, and interactome data reveals hidden components of signaling and regulatory networks. *Science Signaling*, 2(81): ra40.

Huerta, A. M., Salgado, H., Thieffry, D. & Collado-Vides, J. (1998). RegulonDB: A database on transcriptional regulation in Escherichia coli. *Nucleic Acids Research*, 26(1): 55–59.

Hwang, F., Richards, D. & Winter, P. (1992). The steiner tree problem. Elsevier Science Publischers B.V., Amsterdam, The Netherlands. 340 p.

Ideker, T., Ozier, O., Schwikowski, B. & Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18(suppl 1): S233–S240.

Irwin, J. D. & Nelms, R. M. (2008). Basic engineering circuit analysis. Wiley Publishing, Hoboken, New Jersey, USA. 864 p.

Jensen, L. J., Julien, P., Kuhn, M., von Mering, C., Muller, J., Doerks, T. & Bork, P. (2008). eggNOG: automated construction and annotation of orthologous groups of genes. *Nucleic acids research*, 36(suppl 1): D250–D254.

Jensen, L. J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., Bork, P. & von Mering, C. (2009). STRING 8 - a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Research*, 37(suppl 1): D412–D416.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. & Barabási, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, 407(6804): 651–654.

Jiménez, V. M. & Marzal, A. (1999). Computing the k shortest paths: a new algorithm and an experimental comparison. *Proceedings of the Third International Workshop on Algorithm Engineering (WAE 1999)*, London, UK: 15–29.

Joyce, A. R. & Palsson, B. O. (2006). The model organism as a system: integrating 'omics' data sets. *Nature Reviews Molecular Cell Bioliogy*, 7(3): 198–210.

Kanehisa, M. & Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1): 27–30.

Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y. & Hattori, M. (2004). The KEGG resource for deciphering the genome. *Nucleic acids research*, 32(suppl 1): D277–D280.

Kanehisa, M., Goto, S., Sato, Y., Kawashima, M., Furumichi, M. & Tanabe, M. (2014). Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Research*, 42(D1): D199–D205.

Karp, P. D., Ouzounis, C. A., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahrén, D., Tsoka, S., Darzentas, N., Kunin, V. & López-Bigas, N. (2005). Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Research*, 33(19): 6083–6089.

Karp, P. D., Riley, M., Paley, S. M., Pellegrini-Toole, A. & Krummenacker, M. (1997). EcoCyc: Enyclopedia of Escherichia coli genes and metabolism. *Nucleic Acids Research*, 25(1): 43–50.

Karp, P. D., Riley, M., Saier, M., Paulsen, I. T., Collado-Vides, J., Paley, S. M., Pellegrini-Toole, A., Bonavides, C. & Gama-Castro, S. (2002). The EcoCyc database. *Nucleic Acids Research*, 30(1): 56–58.

Khanin, R. & Wit, E. (2006). How scale-free are biological networks. *Journal of Computational Biology*, 13(3): 810–818.

Kirkpatrick, S. (1984). Optimization by simulated annealing: quantitative studies. *Journal of Statistical Physics*, 34(5-6): 975–986.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598): 671–680.

Klein, P. & Ravi, R. (1995). A nearly best-possible approximation algorithm for node-weighted Steiner trees. *Journal of Algorithms*, 19(1): 104–115.

Ko, M. & Park, C. (2000). H-NS-dependent regulation of flagellar synthesis is mediated by a LysR family protein. *Journal of Bacteriology*, 182(16): 4670–4672.

Korf, R. E. (1985). Depth-first iterative-deepening: an optimal admissible tree search. *Artificial Intelligence*, 27(1): 97–109.

Korf, R. E. & Schultze, P. (2005). Large-scale parallel breadth-first search. *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-2005)*, Pittsburgh, Pennsylvania, USA: 1380–1385.

Kou, L., Markowsky, G. & Berman, L. (1981). A fast algorithm for Steiner trees. *Acta informatica*, 15(2): 141–145.

Kwa, J. B. (1989). BS*: An admissible bidirectional staged heuristic search algorithm. *Artificial Intelligence*, 38(1): 95–109.

Lan, A., Smoly, I. Y., Rapaport, G., Lindquist, S., Fraenkel, E. & Yeger-Lotem, E. (2011). ResponseNet: revealing signaling and regulatory networks linking genetic and transcriptomic screening data. *Nucleic Acids Research*, 39(suppl 2): W424–W429.

Latapy, M. & Pons, P. (2005). Computing communities in large networks using random walks. *Proceedings of the 20th International Symposium on Computer and Information Sciences (ISCIS'05)*, Paris, France: 284–293.

Lee, I., Ambaru, B., Thakkar, P., Marcotte, E. M. & Rhee, S. Y. (2010). Rational association of genes with traits using a genome-scale gene network for Arabidopsis thaliana. *Nature Biotechnology*, 28: 149–156.

Lee, I., Date, S. V., Adai, A. T. & Marcotte, E. M. (2004). A probabilistic functional network of yeast genes. *Science*, 306(5701): 1555–1558.

Lee, J., Bansal, T., Jayaraman, A., Bentley, W. E. & Wood, T. K. (2007). Enterohemorrhagic Escherichia coli biofilms are inhibited by 7-hydroxyindole and stimulated by isatin. *Applied and environmental microbiology*, 73(13): 4100–4109.

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J.-B., Volkert, T. L., Fraenkel, E., Gifford, D. K. & Young, R. A. (2002). Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, 298(5594): 799–804.

Licatalosi, D. D. & Darnell, R. B. (2010). RNA processing and its regulation: global insights into biological networks. *Nature Reviews Genetics*, 11(1): 75–87.

Lima-Mendez, G. & van Helden, J. (2009). The powerful law of the power law and other myths in network biology. *Molecular BioSystems*, 5: 1482–1493.

Lipowski, A. & Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6): 2193–2196.

Lovász, L. (1993). Random walks on graphs: a survey. *Bolyai Society Mathematical Studies*, 2: 353–397.

Lowerre, B. (1990). Readings in speech recognition. chapter The Harpy speech understanding system, pp. 576–586. Morgan Kaufmann Publishers Incorporation, San Francisco, California, USA.

Macek, B., Gnad, F., Soufi, B., Kumar, C., Olsen, J. V., Mijakovic, I. & Mann, M. (2008). Phosphoproteome analysis of E. coli reveals evolutionary conservation of bacterial Ser/Thr/Tyr phosphorylation. *Molecular & Cellular Proteomics*, 7(2): 299–307.

Maere, S., Heymans, K. & Kuiper, M. (2005). BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. *Bioinformatics*, 21(16): 3448–3449.

Mason, O. & Verwoerd, M. (2008). Graph theory and networks in biology. Science Foundation, Ireland, UK. 52 p.

Meysman, P., Sonego, P., Bianco, L., Fu, Q., Ledezma-Tejeida, D., Gama-Castro, S., Liebens, V., Michiels, J., Laukens, K., Marchal, K., Collado-Vides, J. & Engelen, K. (2014). COLOMBOS v2.0: an ever expanding collection of bacterial expression compendia. *Nucleic Acids Research*, 42(D1): D649–D653.

Mika, F. & Hengge, R. (2014). Small RNAs in the control of RpoS, CsgD, and biofilm architecture of Escherichia coli. *RiboNucleid Acid Biology*, 11(5): in press.

Miljkovic, D., Stare, T., Mozetič, I., Podpečan, V., Petek, M., Witek, K., Dermastia, M., Lavrač, N. & Gruden, K. (2012). Signalling network construction for modelling plant defence response. *Public Library of Science ONE*, 7(12): e51822.

Moore, E. (1959). The shortest path through a maze. *Proceedings of an International Symposium on the Theory of Switching, Part II*, Cambridge, Massachusetts, USA: 285–292.

Muise, C., McIlraith, S. A., Beck, J. C. & Hsu, E. I. (2012). Advances in artificial intelligence. chapter DSHARP: Fast d-DNNF compilation with sharpSAT, pp. 356–361. Springer, Berlin, Germany.

Nelson, P. C. & Toptsis, A. A. (1992). Unidirectional and bidirectional search algorithms. *Software, Institute of Electrical and Electronics Engineers*, 9(2): 77–83.

Nicholson, T. A. J. (1966). Finding the shortest route between two points in a network. *The Computer Journal*, 9(3): 275–280.

Oliveira, C. A. & Pardalos, P. M. (2011). Mathematical aspects of network routing optimization. chapter Steiner Trees and Multicast, pp. 29–45. Springer, Berlin, Germany.

Oliver, S. (2000). Proteomics: guilt-by-association goes global. *Nature*, 403(6770): 601–603.

Olsen, J. V., Blagoev, B., Gnad, F., Macek, B., Kumar, C., Mortensen, P. & Mann, M. (2006). Global, in vivo, and site-specific phosphorylation dynamics in signaling networks. *Cell*, 127(3): 635–648.

Ourfali, O., Shlomi, T., Ideker, T., Ruppin, E. & Sharan, R. (2007). SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics*, 23(13): i359–i366.

Pavlopoulos, G. A., Secrier, M., Moschopoulos, C. N., Soldatos, T. G., Kossida, S., Aerts, J., Schneider, R. & Bagos, P. G. (2011). Using graph theory to analyze biological networks. *BioData Mining*, 4(1): 10.

Pearl, J. (1984). Heuristics: Intelligent search strategies for computer problem solving. Addison-Wesley Reading, Boston, Massachusetts, USA.

Peregrín-Alvarez, J. M., Xuejian, X., Chong, S. & Parkinson, J. (2009). The modular organization of protein interactions in Escherichia coli. *Public Library of Science Computational Biology*, 5(10): 1–16.

Pesavento, C. & Hengge, R. (2012). The global repressor FliZ antagonizes gene expression by σs-containing rna polymerase due to overlapping DNA binding specificity. *Nucleic acids research*, 40(11): 4783–4793.

Pijls, W. & Post, H. (2009). A new bidirectional search algorithm with shortened postprocessing. *European Journal of Operational Research*, 198(2): 363–369.

Pohl, I. (1970). Bi-directional search. *Machine Intelligence*, 6: 124–140.

Prigent-Combaret, C., Prensier, G., Le Thi, T. T., Vidal, O., Lejeune, P. & Dorel, C. (2000). Developmental pathway for biofilm formation in curli-producing Escherichia coli strains: role of flagella, curli and colanic acid. *Environmental microbiology*, 2(4): 450–464.

Rao, V. & Kumar, V. (1987). Parallel depth first search. part I. Implementation. *International Journal of Parallel Programming*, 16(6): 479–499.

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N. & Barabási, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586): 1551–1555.

Raza, S., McDerment, N., Lacaze, P., Robertson, K., Watterson, S., Chen, Y., Chisholm, M., Eleftheriadis, G., Monk, S., O'Sullivan, M., Turnbull, A., Roy, D., Theocharidis, A., Ghazal, P. & Freeman, T. (2010). Construction of a large scale integrated map of macrophage pathogen recognition and effector systems. *BioMed Central Systems Biology*, 4(1): 63.

Ren, D., Zuo, R., Barrios, A. F. G., Bedzyk, L. A., Eldridge, G. R., Pasmore, M. E. & Wood, T. K. (2005). Differential gene expression for investigation of Escherichia coli biofilm inhibition by plant extract ursolic acid. *Applied and environmental microbiology*, 71(7): 4022–4034.

Rockman, M. V. & Kruglyak, L. (2006). Genetics of global gene expression. *Nature Reviews Genetics*, 7(11): 862–872.

Rosete-Suarez, A. & Ochao-Rodriguez, A. (1999). Automatic graph drawing and stochastic hill climbing. *Proceedings of the Genetic and Evolutionary Computation Conference (Gecco'99)*, Orlando, Florida, USA: 1699–1706.

Russel, S. & Norvig, P. (2003). Artificial intellegence: a modern approach. Pearson Education International, Upper Saddle River, New Jersey, USA. 1081 p.

Salgado, H., Peralta-Gil, M., Gama-Castro, S., Santos-Zavaleta, A., Muñiz Rascado, L., García-Sotelo, J. S., Weiss, V., Solano-Lira, H., Martínez-Flores, I., Medina-Rivera, A., Salgado-Osorio, G., Alquicira-Hernández, S., Alquicira-Hernández, K., López-Fuentes, A., Porrón-Sotelo, L., Huerta, A. M., Bonavides-Martínez, C., Balderas-Martínez, Y. I., Pannier, L., Olvera, M., Labastida, A., Jiménez-Jacinto, V., Vega-Alvarado, L., del Moral-Chávez, V., Hernández-Alvarez, A., Morett, E. & Collado-Vides, J. (2013). RegulonDB v8.0: omics data sets, evolutionary conservation, regulatory phrases, cross-validated gold standards and more. *Nucleic Acids Research*, 41(D1): D203–D213.

Sanchez, C., Lachaize, C., Janody, F., Bellon, B., Röder, L., Euzenat, J., Rechenmann, F. & Jacq, B. (1999). Grasping at molecular interactions and genetic networks in Drosophila melanogaster using FlyNets, an internet database. *Nucleic Acids Research*, 27(1): 89–94.

Sardiu, M. E. & Washburn, M. P. (2011). Building protein-protein interaction networks with proteomics and informatics tools. *Journal of Biological Chemistry*, 286(27): 23645–23651.

Sastry, K., Goldberg, D. E. & Kendall, G. (2014). Search methodologies. chapter Genetic Algorithms, pp. 93–117. Springer, New York City, New York, USA.

Schadt, E. E. & Lum, P. Y. (2006). Thematic review series: systems biology approaches to metabolic and cardiovascular disorders. Reverse engineering gene networks to identify key drivers of complex disease phenotypes. *Journal of lipid research*, 47(12): 2601–2613.

Schneider, A., Dessimoz, C. & Gonnet, G. H. (2007). OMA Browser-Exploring orthologous relations across 352 complete genomes. *Bioinformatics*, 23(16): 2180–2182.

Schomburg, I., Chang, A. & Schomburg, D. (2002). BRENDA, enzyme data and metabolic information. *Nucleic acids research*, 30(1): 47–49.

Schuster, S., Fell, D. A. & Dandekar, T. (2000). A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature biotechnology*, 18(3): 326–332.

Scott, M. S., Perkins, T., Bunnell, S., Pepin, F., Thomas, D. Y. & Hallett, M. (2005). Identifying regulatory subnetworks for a set of genes. *Molecular & Cellular Proteomics*, 4(5): 683–692.

Sint, L. & de Champeaux, D. (1977). An improved bidirectional heuristic search algorithm. *Journal of the Association for Computing Machinery*, 24(2): 177–191.

Sivaraj, R. & Ravichandran, T. (2011). A review of selection methods in genetic algorithm. *International Journal of Engineering Science & Technology*, 3(5).

Smith, C., Arany, Z., Orrego, C. & Eisenstadt, E. (1991). DNA damage-inducible loci in Salmonella typhimurium. *Journal of bacteriology*, 173(11): 3587–3590.

Snel, B., Lehmann, G., Bork, P. & Huynen, M. A. (2000). STRING: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucleic Acids Research*, 28(18): 3442–3444.

Snel, B., Van Noort, V. & Huynen, M. A. (2004). Gene co-regulation is highly conserved in the evolution of eukaryotes and prokaryotes. *Nucleic acids research*, 32(16): 4725–4731.

Spory, A., Bosserhoff, A., von Rhein, C., Goebel, W. & Ludwig, A. (2002). Differential regulation of multiple proteins of Escherichia coli and Salmonella enterica serovar Typhimurium by the transcriptional regulator SlyA. *Journal of bacteriology*, 184(13): 3549–3559.

Stark, C., Breitkreutz, B. J., Reguly, T., Boucher, L., Breitkreutz, A. & Tyers, M. (2006). BioGRID: a general repository for interaction datasets. *Nucleic Acids Research*, 34(1): D535–D539.

Steenackers, H., Hermans, K., Vanderleyden, J. & De Keersmaecker, S. C. (2012). *Salmonella* biofilms: an overview on occurrence, structure, regulation and eradication. *Food Research International*, 45(2): 502–531.

Suthram, S., Beyer, A., Karp, R. M., Eldar, Y. & Ideker, T. (2008). eQED: an efficient method for interpreting eQTL associations using protein networks. *Molecular Systems Biology*, 4(162).

Takahashi, H. & Matsuyama, A. (1980). An approximate solution for the Steiner problem in graphs. *Mathematica Japonica*, 24(6): 573–577.

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *Society for Industrial and Applied Mathematics, Journal on Computing*, 1(2): 146–160.

Tatusov, R. L., Koonin, E. V. & Lipman, D. J. (1997). A genomic perspective on protein families. *Science*, 278(5338): 631–637.

Thede, S. M. (2004). An introduction to genetic algorithms. *Journal of computing sciences in colleges*, 20(1): 115–123.

Tian, F., Shah, P. K., Liu, X., Negre, N., Chen, J., Karpenko, O., White, K. P. & Grossman, R. L. (2009). Flynet: a genomic resource for Drosophila melanogaster transcriptional regulatory networks. *Bioinformatics*, 25(22): 3001–3004.

Tirosh, I., Bilu, Y. & Barkai, N. (2007). Comparative biology: beyond sequence analysis. *Current opinion in biotechnology*, 18(4): 371–377.

Tomasz, M. (1995). Mitomycin C: small, fast and deadly (but very selective). *Chemistry & Biology*, 2(9): 575–579.

Tong, A. H. Y., Evangelista, M., Parsons, A. B., Xu, H., Bader, G. D., Pagé, N., Robinson, M., Raghibizadeh, S., Hogue, C. W. V., Bussey, H., Andrews, B., Tyers, M. & Boone, C. (2001). Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294(5550): 2364–2368.

Tu, Z., Wang, L., Arbeitman, M. N., Chen, T. & Sun, F. (2006). An integrative approach for causal gene identification and gene regulatory pathway inference. *Bioinformatics*, 22(14): e489–e496.

Tuncbag, N., McCallum, S., Huang, S. C. & Fraenkel, E. (2012). SteinerNet: a web server for integrating omic data to discover hidden components of response pathways. *Nucleic Acids Research*, 40(W1): W505–W509.

Van Helden, J., Naim, A., Mancuso, R., Eldridge, M., Wernisch, L., Gilbert, D. & Wodak, S. J. (2000). Representing and analysing molecular and cellular function using the computer. *Biological chemistry*, 381: 921–935.

Van Landeghem, S., Ginter, F., Van de Peer, Y. & Salakoski, T. (2011). EVEX: a pubmed-scale resource for homology-based generalization of text mining predictions. *Proceedings of Biomedical Natural Language Processing 2011 WorkshopAssociation for Computational Linguistics (BioNLP11)*, Portland, Oregon, USA: 28–37.

von Mering, C., Huynen, M., Jaeggi, D., Schmidt, S., Bork, P. & Snel, B. (2003). STRING: a database of predicted functional associations between proteins. *Nucleic Acids Research*, 31(1): 258–261.

von Mering, C., Jensen, L. J., Kuhn, M., Chaffron, S., Doerks, T., Krüger, B., Snel, B. & Bork, P. (2007). STRING 7 – recent developments in the integration and prediction of protein interactions. *Nucleic acids research*, 35(suppl 1): D358–D362.

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S. & Bork, P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417: 399–403.

Wang, F. & Lim, A. (2007). A stochastic beam search for the berth allocation problem. *Decision Support Systems*, 42(4): 2186–2196.

Wessels, L., van Someren, E. & Reinders, M. (2001). A comparison of genetic network models. *Pacific Symposium on Biocomputing*, 6: 508–519.

Wilson, N. (2004). Human protein reference database. *Nature Reviews Genetics*, 5(1): 8.

Xenarios, I., Rice, D. W., Salwinski, L., Baron, M. K., Marcotte, E. M. & Eisenberg, D. (2000). DIP: the database of interacting proteins. *Nucleic Acids Research*, 28(1): 289–291.

Yeang, C. H., Ideker, T. & Jaakkola, T. (2004). Physical network models. *Journal of computational biology*, 11(2-3): 243–262.

Yeang, C. H. & Jaakkola, T. (2003). Physical network models and multi-source data integration. *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB '03)*, Berlin, Germany: 312–321.

Yeger-Lotem, E. & Margalit, H. (2003). Detection of regulatory circuits by integrating the cellular networks of protein-protein interactions and transcription regulation. *Nucleic acids research*, 31(20): 6053–6061.

Yeger-Lotem, E., Riva, L., Su, L. J., Gitler, A. D., Cashikar, A. G., King, O. D., Auluck, P. K., Geddie, M. L., Valastyan, J. S., Karger, D. R., Lindquist, S. & Fraenkel, E. (2009). Bridging high-throughput genetic and transcriptional data reveals cellular responses to alpha-synuclein toxicity. *Nature Genetics*, 41(3): 316–323.

Yeger-Lotem, E., Sattath, S., Kashtan, N., Itzkovitz, S., Milo, R., Pinter, R. Y., Alon, U. & Margalit, H. (2004). Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proceedings of the National Academy of Sciences of the United States of America*, 101(16): 5934–5939.

Yoo, A., Chow, E., Henderson, K., McLendon, W., Hendrickson, B. & Catalyurek, U. (2005). A scalable distributed parallel breadth-first search algorithm on BlueGene/L. *Proceedings of the Association for Computing Machinery/Institute of Electrical and Electronics Engineers Super Computing 2005 Conference (SC'05)*, Washington DC, Washington, USA: 19.

Yvert, G., Brem, R. B., Whittle, J., Akey, J. M., Foss, E., Smith, E. N., Mackelprang, R. & Kruglyak, L. (2003). Trans-acting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors. *Nature genetics*, 35(1): 57–64.

Zanzoni, A., Montecchi-Palazzi, L., Quondam, M., Ausiello, G., Helmer-Citterich, M. & Cesareni, G. (2002). MINT: a Molecular INTeraction database. *Federation of European Biochemical Societies Letters*, 513(1): 135–140.

Zhang, B. & Horvarth, S. (2005). A general framework for weighted gene co-expression network analysis. *Statisticaal Applications in Genetics and Molecular Biology*, 4(1): Artikel 17.

Zhang, J., Lu, K., Xiang, Y., Islam, M., Kotian, S., Kais, Z., Lee, C., Arora, M., Liu, H., Parvin, J. D. & Huang, K. (2012). Weighted frequent gene co-expression network mining to identify genes involved in genome stability. *Public Library of Science Computational Biology*, 8(8): 1–15.

Zhang, X.-S., García-Contreras, R. & Wood, T. K. (2007). YcfR (BhsA) influences Escherichia coli biofilm formation through stress response and surface hydrophobicity. *Journal of Bacteriology*, 189(8): 3051–3062.

Deel IV

Bijlage

Bijlage A

Scala codes

A.1 Code om genenparenbestand aan te maken (1 richting)

```
import scala.io.Source
import java.io.PrintWriter
import scopt.OptionParser

object CreateGenePairFile {
  val parser = new OptionParser[Arguments]("scopt") {
    head("scopt", "3.x")
    opt[String]('i', "input") required () action { (x: String, c) =>
      c.copy(inputFile = x)
    } text ("Full path of the input file.")
    opt[String]('o', "outputFile") required () action { (x, c) =>
      c.copy(outputFile = x)
    } text ("Full path of the outputFile")
    opt[Double]('e', "expressionFraction") action { (x: Double, c) =>
      c.copy(expressionFraction = x)
    } text ("The percentage most differential expressed genes to use as input")
  }
  def main(args: Array[String]) {
    val arguments = parser.parse(args, Arguments()).get
    val outputfile = new PrintWriter(arguments.outputFile)
    val fractionOfGenesAsExpression = arguments.expressionFraction

    val inputFile = Source
        .fromFile(arguments.inputFile)
        .getLines
        .toList
        .tail
        .filter(!_.trim.isEmpty())
        .map( line => {
          val splitted = line.split("\t")
          (splitted(0), splitted(1).replace(",",".").toDouble)
          }).toList
    val visitedDEs = new scala.collection.mutable.HashSet[String]()

    val topBestDEs = inputFile
          .sortBy(x => math.abs(x._2))
          .reverse
          .slice(0, (inputFile.size * fractionOfGenesAsExpression).toInt)
    topBestDEs.map((source) => {
          visitedDEs.add(source._1)
          topBestDEs
            .filter(x => !visitedDEs.contains(x._1))
            .map((x) => {
              val score = math.abs(x._2) + math.abs(source._2)
              outputfile.println(source._1 + "," + x._1 + "," + score)
            })
    })
        outputfile.close
  }
  case class Arguments(outputFile: String = "",
      inputFile: String = "",
      expressionFraction: Double = 1d)
}
```

A.2 Code om genenparenbestand aan te maken (2 richtingen)

```
import scala.io.Source
import java.io.PrintWriter
import scopt.OptionParser

object CreateGenePairFileFullCartesian {
  val parser = new OptionParser[Arguments]("scopt") {
    head("scopt", "3.x")
    opt[String]('i', "input") required () action { (x: String, c) =>
      c.copy(inputFile = x)
    } text ("Full path of the input file.")
    opt[String]('o', "outputFile") required () action { (x, c) =>
      c.copy(outputFile = x)
    } text ("Full path of the outputFile")
    opt[Double]('e', "expressionFraction") action { (x: Double, c) =>
      c.copy(expressionFraction = x)
    } text ("The percentage most differential expressed genes to use as input")
  }
  def main(args: Array[String]) {
    val arguments = parser.parse(args, Arguments()).get
    val outputfile = new PrintWriter(arguments.outputFile)
    val fractionOfGenesAsExpression = arguments.expressionFraction

    val inputFile = Source
        .fromFile(arguments.inputFile)
        .getLines
        .toList
        .tail
        .filter(!_.trim.isEmpty())
        .map( line => {
          val splitted = line.split(",")
          (splitted(0), splitted(1).replace(",",".").toDouble)
          }).toList

    val topBestDEs = inputFile
          .sortBy(x => math.abs(x._2))
          .reverse
          .slice(0, (inputFile.size * fractionOfGenesAsExpression).toInt)
    topBestDEs.map((source) => {
          topBestDEs
            .map((x) => {
              val score = math.abs(x._2) + math.abs(source._2)
              outputfile.println(source._1 + "," + x._1 + "," + score)
            })
    })
        outputfile.close
  }
  case class Arguments(outputFile: String = "",
      inputFile: String = "",
      expressionFraction: Double = 1d)
}
```

A.3 Code extractie specifieke 'hubs'

```
import scopt.OptionParser
import java.io.PrintWriter
import scala.io.Source

object SpecificHubsReweighting {
    val parser = new OptionParser[Arguments]("scopt") {
    head("scopt", "3.x")
    opt[String]('i', "input") required () action { (x: String, c) =>
      c.copy(inputFile = x)
    } text ("Full path of the input file.")
    opt[String]('o', "outputFile") required () action { (x, c) =>
      c.copy(outputFile = x)
    } text ("Full path of the outputFile")
    opt[String]('r', "remove") required () action { (x: String, c) =>
      c.copy(hubs = x)
    } text ("STM-names of the hubs to remove (separated by commas)")
  }
  def main(args: Array[String]) {
    val arguments = parser.parse(args, Arguments()).get
    val outputfile = new PrintWriter(arguments.outputFile)
    val information = Source
      .fromFile(arguments.inputFile)
      .getLines
      .filter(!_.trim.isEmpty())
      .filter(_.contains("%"))
      .foreach(line => {
        outputfile.println(line)
      })
    val inputFile = Source
      .fromFile(arguments.inputFile)
      .getLines
      .filter(!_.trim.isEmpty())
      .filter(!_.contains("%"))
      .map(line => {
        val splitted = line.split("\t")
        (splitted(0), splitted(1), splitted(2), splitted(3))
      }).toList
    val filteredNodes = arguments.hubs
    inputFile
      .filter(x => {
        !filteredNodes.contains(x._1) && !filteredNodes.contains(x._2)
      })
      .foreach(x => {
        outputfile.println(x._1 + "\t" + x._2 + "\t" + x._3 + "\t" + x._4 + "\t")
      })
    outputfile.close
  }
  case class Arguments(outputFile: String = "", inputFile: String = "", hubs: String = """""")
}
```

A.4 Code herweging netwerk met exponentiële graadverdeling

```
import scopt.OptionParser
import java.io.PrintWriter
import scala.io.Source
import org.apache.commons.math3.distribution.ExponentialDistribution

object HubReweightingWithExponentialDegreeDistribution {
  val parser = new OptionParser[Arguments]("scopt") {
    head("scopt", "3.x")
    opt[String]('i', "input") required () action { (x: String, c) =>
      c.copy(inputFile = x)
    } text ("Full path of the input file.")
    opt[String]('o', "outputFile") required () action { (x, c) =>
      c.copy(outputFile = x)
    } text ("Full path of the outputFile")
  }
  def main(args: Array[String]) {
    val arguments = parser.parse(args, Arguments()).get
    val outputfile = new PrintWriter(arguments.outputFile)
    val interactionTypeMap = Source
      .fromFile(arguments.inputFile)
      .getLines
      .filter(!_.trim.isEmpty())
      .filter(_.contains("%"))
      .map(line => {
        outputfile.println(line)
        val splitted = line.split(" ")
        (splitted(1), InteractionType(splitted(1), (if (splitted(3) == "directed") true
            else false)))
      }).toMap
    val graph = new Graph(Source
      .fromFile(arguments.inputFile)
      .getLines
      .filter(!_.trim.isEmpty())
      .filter(!_.contains("%"))
      .map(line => {
        val splitted = line.split("\t")
        Interaction(splitted(0), splitted(1), interactionTypeMap(splitted(2)),
            splitted(3).toDouble)
      }).toSet)
    val distribution = new ExponentialDistribution(graph.getMeanOutDegree)
    outputfile.println( graph.interactions.map(interaction => {
      interaction.reweighProbability((1 - distribution.cumulativeProbability(
          graph.getOutDegreeFor(interaction.to))))
    }).mkString("\n"))

    outputfile.close
  }
  case class Arguments(outputFile: String = "", inputFile: String = "")

  class Graph(val interactions: Set[Interaction]) {
    val nodes = interactions.map(_.getGenes).flatten
    private val outgoingInteractions: Map[String, Int] = {
      nodes.map(node => {
        (node, interactions.filter(interaction => interaction.canStartIn(node)).size)
      }).toMap
    }
    def getOutDegreeFor(node: String) = outgoingInteractions(node)
    val getMeanOutDegree = nodes.map(outgoingInteractions(_)).sum / nodes.size
  }

  case class Interaction(from: String, to: String, typ: InteractionType, probability: Double){
    def getGenes = Set(from, to)
    def canStartIn(node: String): Boolean = {
```

```
      if (typ.isDirected) return from == node
      else from == node || to == node
    }
    def reweighProbability(probability: Double): Interaction = Interaction(from, to,
        typ, this.probability * probability)
    override def toString = from + "\t" + to + "\t" + typ.name + "\t" + probability
  }

  case class InteractionType(name: String, isDirected: Boolean)
}
```

A.5 Code herweging netwerk met sigmoïdale graadverdeling

```
import scopt.OptionParser
import java.io.PrintWriter
import scala.io.Source
import org.apache.commons.math3.distribution.ExponentialDistribution

object HubReweightingWithSigmoidalDegreeDistribution {
  val parser = new OptionParser[Arguments]("scopt") {
    head("scopt", "3.x")
    opt[String]('i', "input") required () action { (x: String, c) =>
      c.copy(inputFile = x)
    } text ("Full path of the input file.")
    opt[String]('o', "outputFile") required () action { (x, c) =>
      c.copy(outputFile = x)
    } text ("Full path of the outputFile")
  }
  def main(args: Array[String]) {
    val arguments = parser.parse(args, Arguments()).get
    val outputfile = new PrintWriter(arguments.outputFile)
    val interactionTypeMap = Source
      .fromFile(arguments.inputFile)
      .getLines
      .filter(!_.trim.isEmpty())
      .filter(_.contains("%"))
      .map(line => {
        outputfile.println(line)
        val splitted = line.split(" ")
        (splitted(1), InteractionType(splitted(1), (if (splitted(3) == "directed") true
            else false)))
      }).toMap
    val graph = new Graph(Source
      .fromFile(arguments.inputFile)
      .getLines
      .filter(!_.trim.isEmpty())
      .filter(!_.contains("%"))
      .map(line => {
        val splitted = line.split("\t")
        Interaction(splitted(0), splitted(1), interactionTypeMap(splitted(2)),
            splitted(3).toDouble)
      }).toSet)
    val distribution = new ExponentialDistribution(graph.getMeanOutDegree)
    val inflectionValue = distribution.inverseCumulativeProbability(0.95)
    val dampingFactor = inflectionValue / 4
    outputfile.println(graph.interactions.map(interaction => {
      interaction.reweighProbability(math.max(0.01,
          math.sqrt(1 / (1 + math.exp((graph.getOutDegreeFor(interaction.to) - inflectionValue)/dampingFactor))
    ))))}).mkString("\n"))

    outputfile.close
  }
  case class Arguments(outputFile: String = "", inputFile: String = "")
```

```
  class Graph(val interactions: Set[Interaction]) {
    val nodes = interactions.map(_.getGenes).flatten
    private val outgoingInteractions: Map[String, Int] = {
      nodes.map(node => {
        (node, interactions.filter(interaction => interaction.canStartIn(node)).size)
      }).toMap
    }
    def getOutDegreeFor(node: String) = outgoingInteractions(node)
    val getMeanOutDegree = nodes.map(outgoingInteractions(_)).sum / nodes.size
  }

  case class Interaction(from: String, to: String, typ: InteractionType, probability: Double){
    def getGenes = Set(from, to)
    def canStartIn(node: String): Boolean = {
      if (typ.isDirected) return from == node
      else from == node || to == node
    }
    def reweighProbability(probability: Double): Interaction = Interaction(from, to,
        typ, this.probability * probability)
    override def toString = from + "\t" + to + "\t" + typ.name + "\t" + probability
  }

  case class InteractionType(name: String, isDirected: Boolean)
}
```

A.6 Code herweging netwerk met differentiële expressiewaarden

```
import scopt.OptionParser
import scala.io.Source
import java.io.PrintWriter
import org.apache.commons.math3.stat.descriptive.DescriptiveStatistics
import org.apache.commons.math3.distribution.NormalDistribution

object DiffExpressionReweighting {
  val parser = new OptionParser[Arguments]("scopt") {
    head("scopt", "3.x")
    opt[String]('n', "network") required () action { (x: String, c) =>
      c.copy(networkFile = x)
    } text ("Full path of the used network file.")
    opt[String]('e', "expression") required () action { (x, c) =>
      c.copy(fullExpressionFile = x)
    } text ("Full path of the file with all the expression values")
    opt[String]('o', "outputFile") required () action { (x, c) =>
      c.copy(outputFile = x)
    } text ("Full path of the outputFile")
  }
  def main(args: Array[String]) {
    val arguments = parser.parse(args, Arguments()).get
    val outputfile = new PrintWriter(arguments.outputFile)
    val information = Source
      .fromFile(arguments.networkFile)
      .getLines
      .filter(!_.trim.isEmpty())
      .filter(_.contains("%"))
      .foreach(line => {
        outputfile.println(line)
      })
    val interactions = Source
      .fromFile(arguments.networkFile)
      .getLines
      .filter(!_.trim.isEmpty())
      .filter(!_.contains("%"))
```

```
    .map(line => {
      val splitted = line.split("\t")
      (splitted(0), splitted(1), splitted(2), splitted(3).toDouble)
    }).toList
  val expressionFile = Source
    .fromFile(arguments.fullExpressionFile)
    .getLines
    .toList
    .tail
    .filter(!_.trim.isEmpty())
    .map(line => {
      val splitted = line.split("\t")
      (splitted(0), splitted(1).replace(",",".").toDouble)
    }).toList

  val statistic = new DescriptiveStatistics()
  expressionFile.foreach((x) => statistic.addValue(x._2))
  val distribution = new NormalDistribution(statistic.getMean(),
      statistic.getStandardDeviation())

  val expressionProbabilities = expressionFile.map((x) => {
    val expressionProbability = math.max(distribution.cumulativeProbability(x._2),
        1 - distribution.cumulativeProbability(x._2))
    val normalisedExpressionProbability = (expressionProbability - 0.5) * 2
    (x._1, normalisedExpressionProbability)
  }).toMap

  interactions.map((x) => {
      val fromExpressionProbability =
        if(expressionProbabilities.contains(x._1)) expressionProbabilities(x._1) else 0.5
      val toExpressionProbability =
        if(expressionProbabilities.contains(x._2)) expressionProbabilities(x._2) else 0.5
  val newprob = x._4 * math.max(fromExpressionProbability,toExpressionProbability)
  outputfile.println(x._1 + "\t" + x._2 + "\t" + x._3 + "\t" + newprob)
  })
  outputfile.close()
 }
case class Arguments(networkFile: String = "",
    fullExpressionFile: String = "",
    outputFile: String = "")}
```

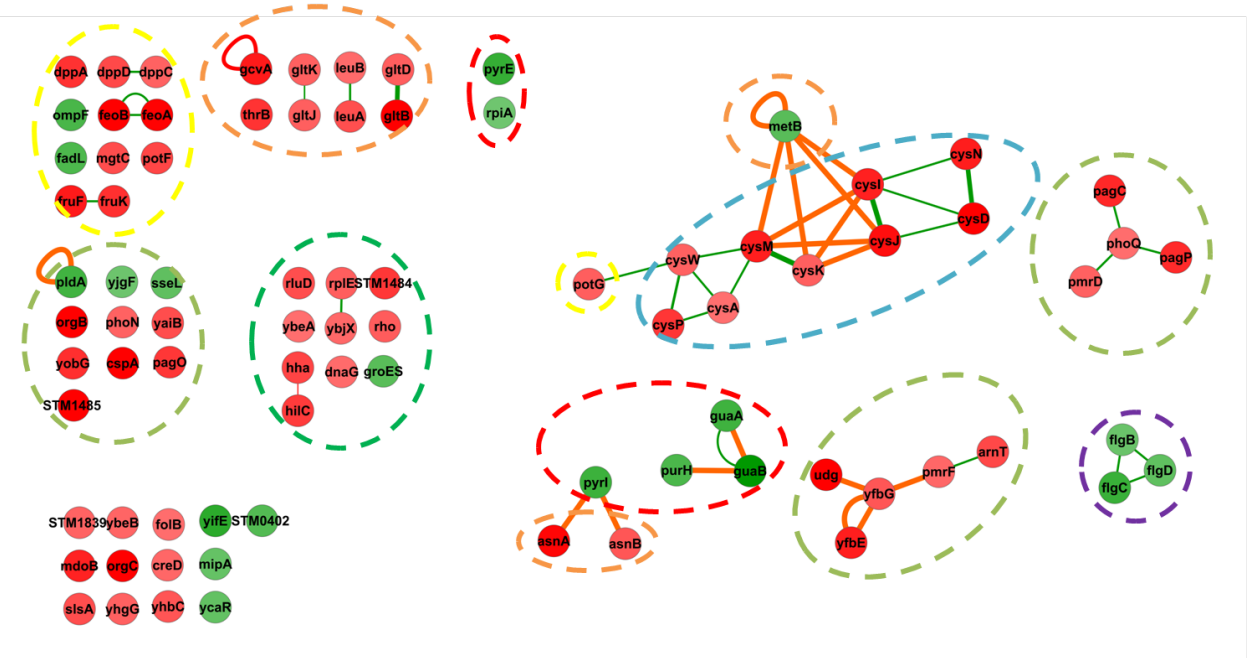
Bijlage B

Visualisatie gevalstudie

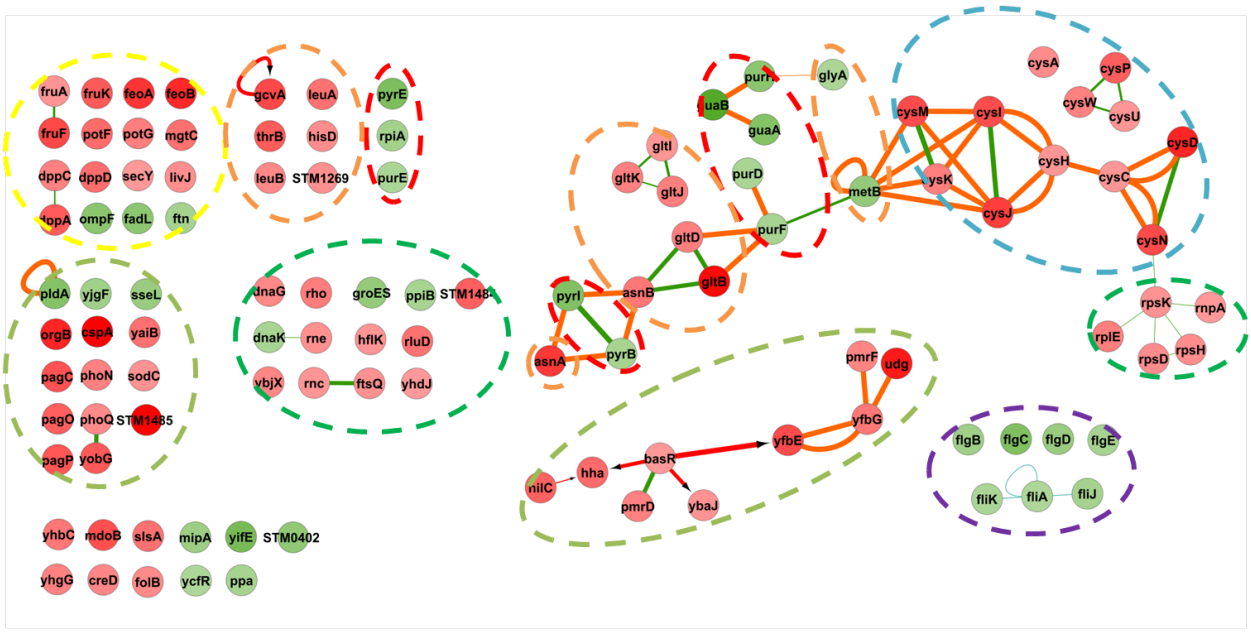
Deze sectie bevat de figuren horende bij de gevalstudie (hoofdstuk 9). Voor alle figuren is de legende als volgt. Knooppunten in de netwerken stellen genen (of hun eiwitproducten) voor met een kleur overeenkomstig de graad van differentiële expressie: rood voor overexpressie, groen voor onderexpressie en grijs indien niet gemeten. Verbindingslijnen in de netwerken stellen interacties voor: eiwit-eiwitinteracties zijn aangegeven in groen, eiwit-DNA-interacties in rood, metabolische interacties in oranje, sigma-interacties in lichtblauw, sRNA-interacties is turkoois en fosforylaties in paars. De voorkomende functionele groepen zijn aangeduid met onderbroken cirkels: blauwe cirkels voor sulfaat- of nitraatmetabolisme, zwarte cirkels voor koolstofmetabolisme, oranje cirkels voor aminozuurmetabolisme, gele cirkels voor transport (aminozuren, koolhydraten, ionen en peptiden), rode cirkels voor het metabolisme van purine en pyrimidine (ribo)nucleotiden, donkergroene cirkels voor celmetabolisme (celdeling, DNA-methylatie, RNA-processing, eiwitsynthese en eiwitvouwing), lichtgroene cirkels voor lipidenmetabolisme en paarse cirkels voor flagelbeweging. De netwerkgrootte wordt per figuur meegegeven.

B.1 Visualisatie startdata

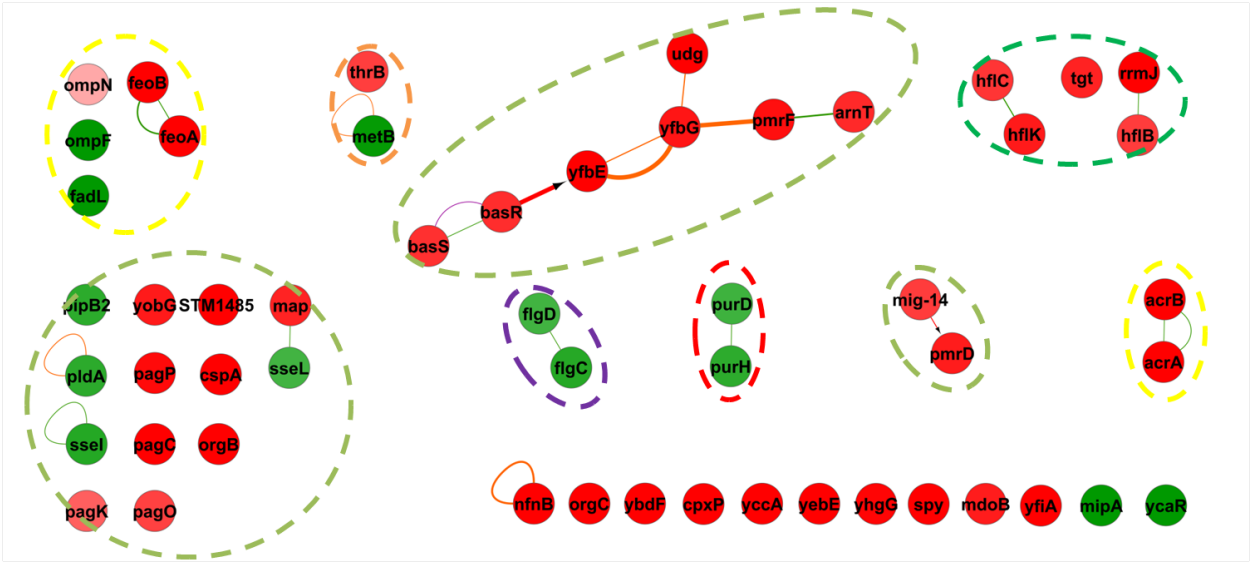
Elke figuur visualiseert de sterk differentieel geëxpresseerde genen na de betreffende imidazoolbehandeling op het fysieke interactienetwerk van *Salmonella* LT2. Tenzij anders vermeld, zijn de netwerken opgebouwd uit de vijf percent meest differentieel geëxpresseerde genen.



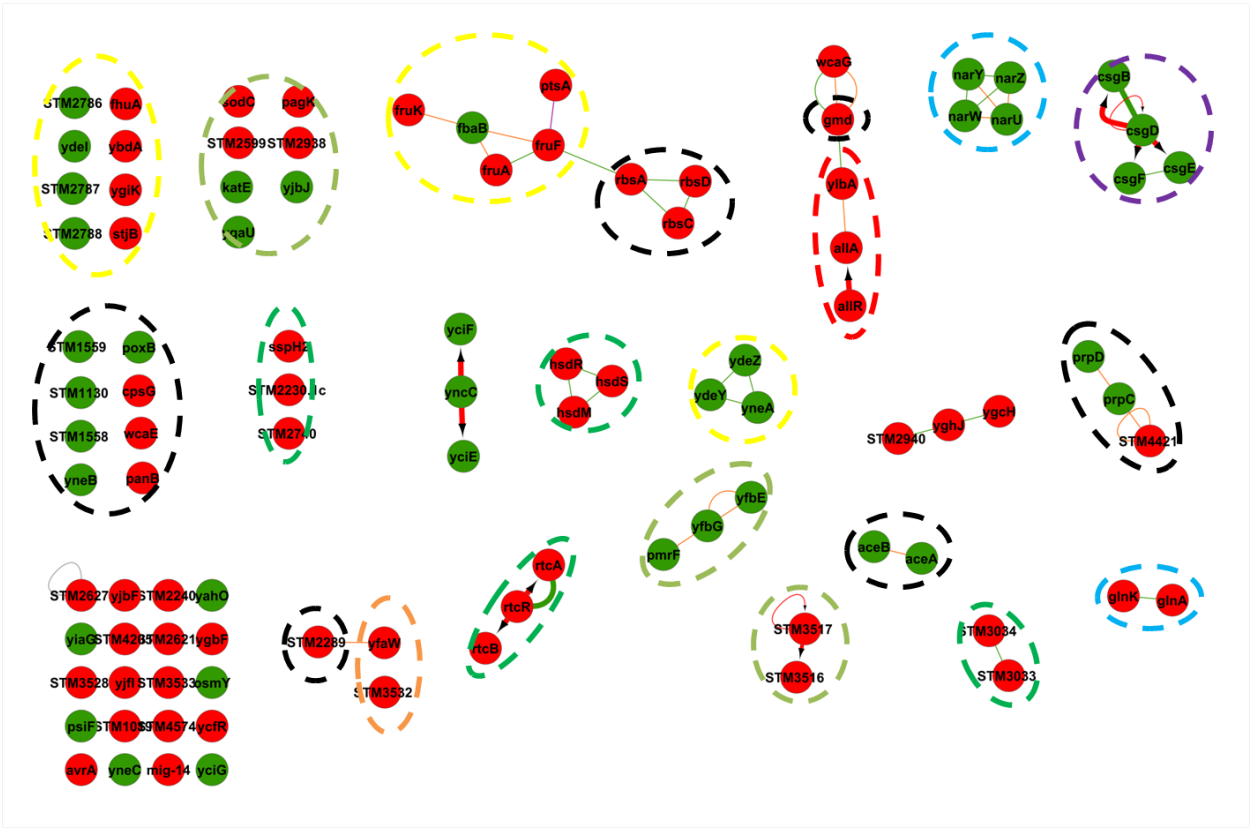
Figuur B.1. Biologische exploratie van de startdata gebruikt voor het experiment imidazool_1 in de gevalstudie. Het netwerk bestaat uit 83 knooppunten en 48 verbindingslijnen.



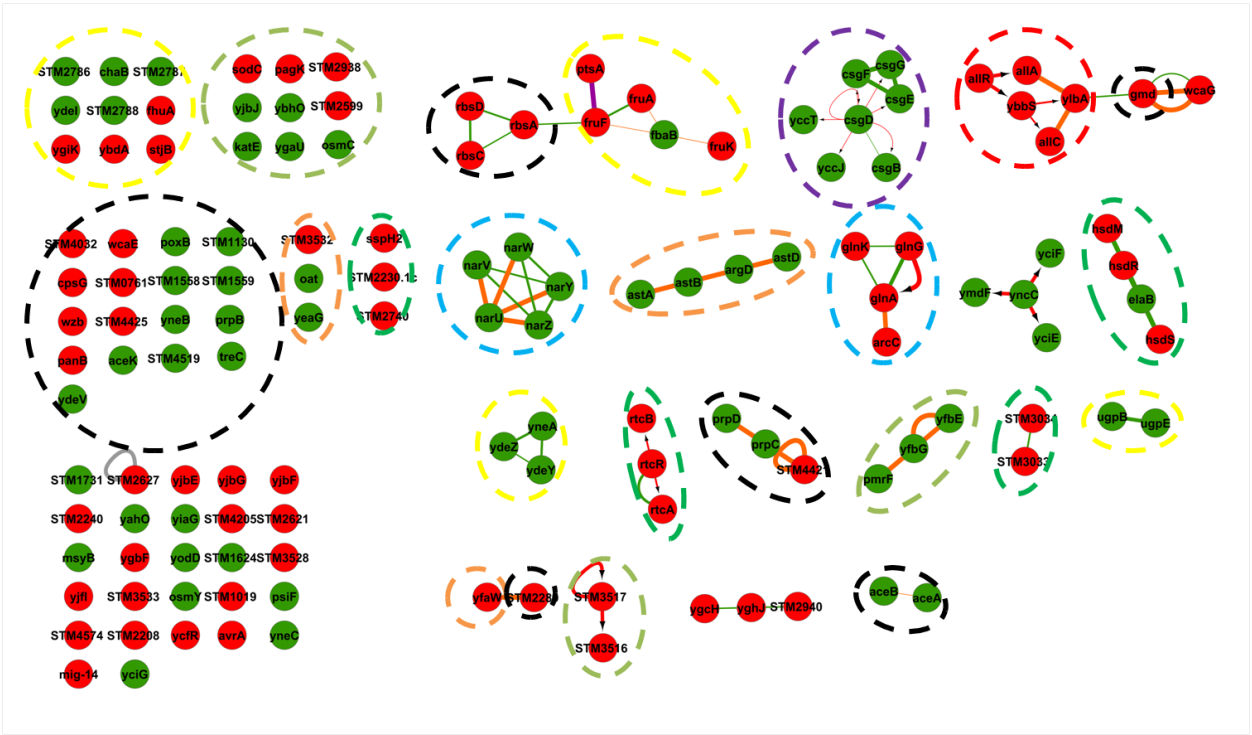
Figuur B.2. Biologische exploratie van de startdata gebruikt voor het experiment imidazool_2 in de gevalstudie. Dit netwerk is opgebouwd op basis van de differentieel geëxpresseerde genen met een foldchange (1/foldchange) hoger dan 2. Het netwerk bestaat uit 112 knooppunten en 67 verbindingslijnen.



Figuur B.3. Biologische exploratie van de startdata gebruikt voor het experiment imidazoline in de gevalstudie. Het netwerk bestaat uit 52 knooppunten en 22 verbindingslijnen.



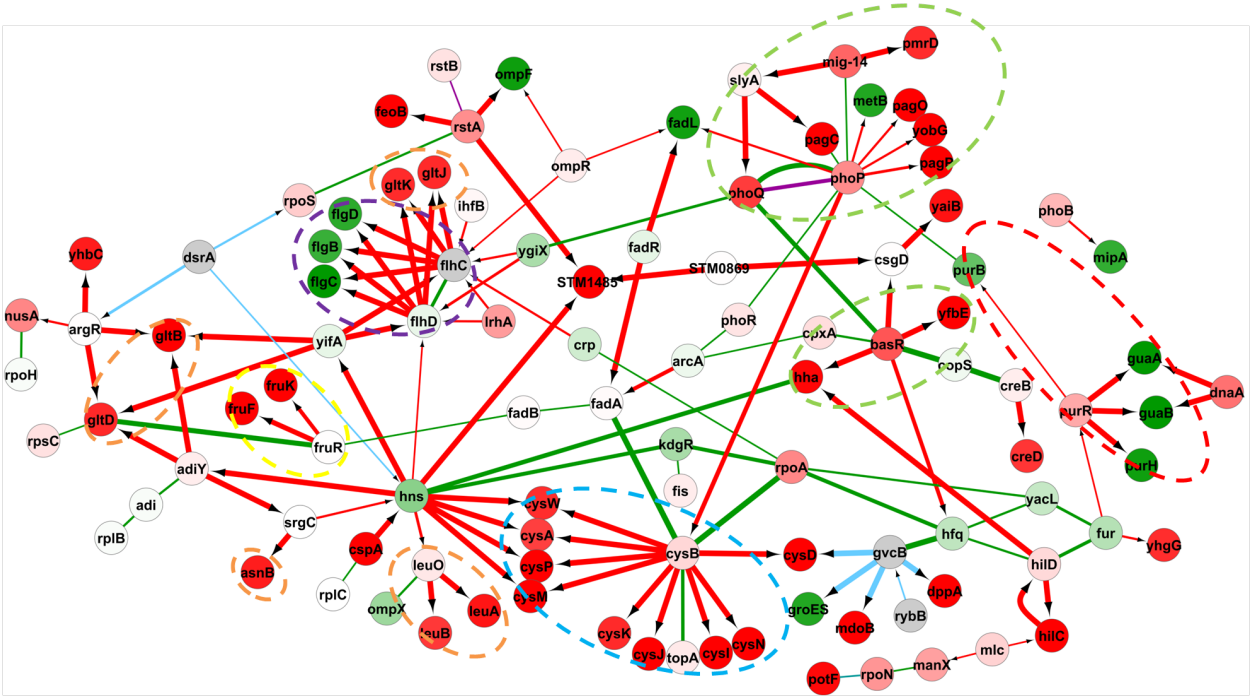
Figuur B.4. Biologische exploratie van de startdata gebruikt voor het experiment sensitiviteit_1 in de gevalstudie. Het netwerk bestaat uit 99 knooppunten en 53 verbindingslijnen.



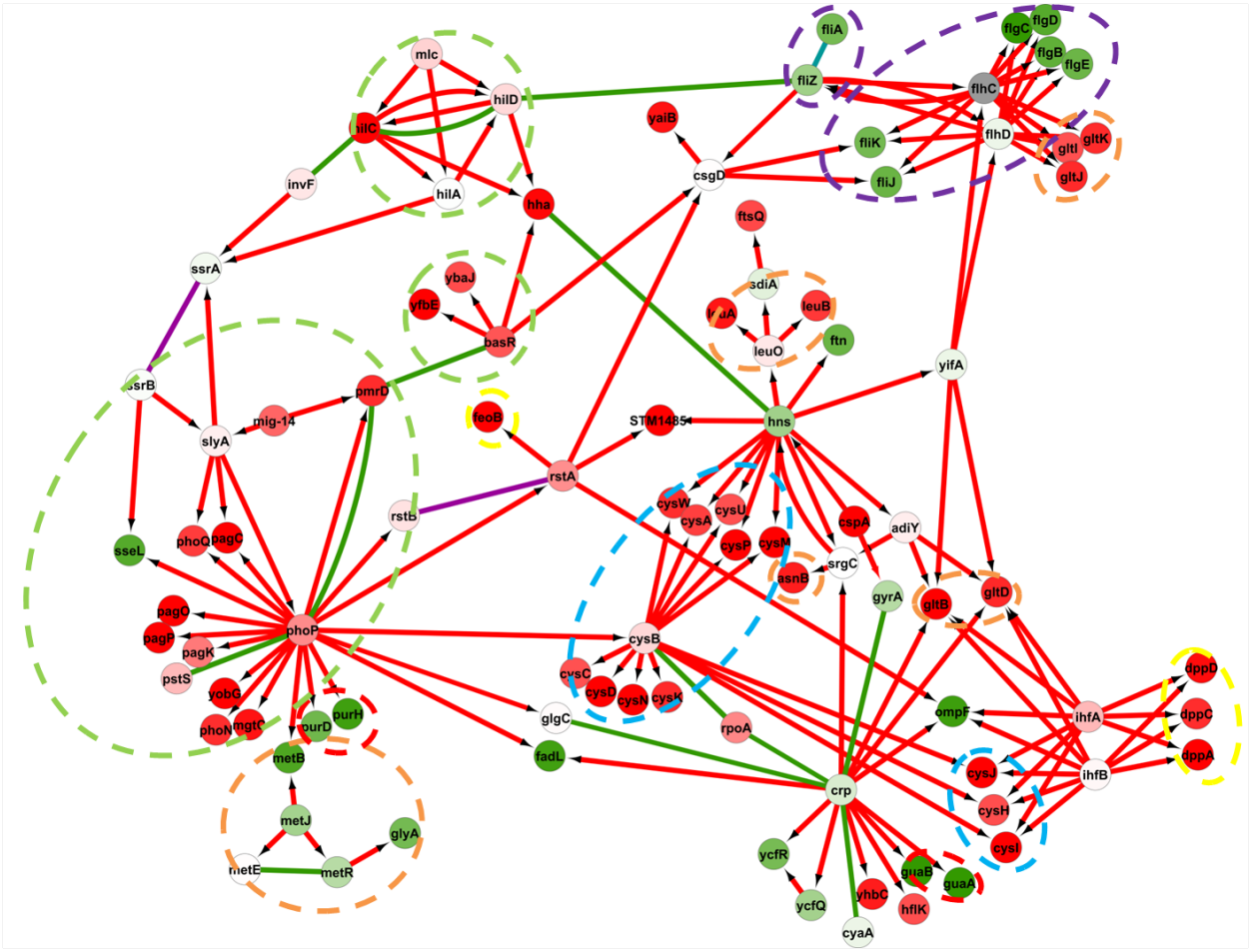
Figuur B.5. Biologische exploratie van de startdata gebruikt voor het experiment sensitiviteit_1 in de gevalstudie. Het netwerk bestaat uit 136 knooppunten en 75 verbindingslijnen.

B.2 Visualisatie resultaten

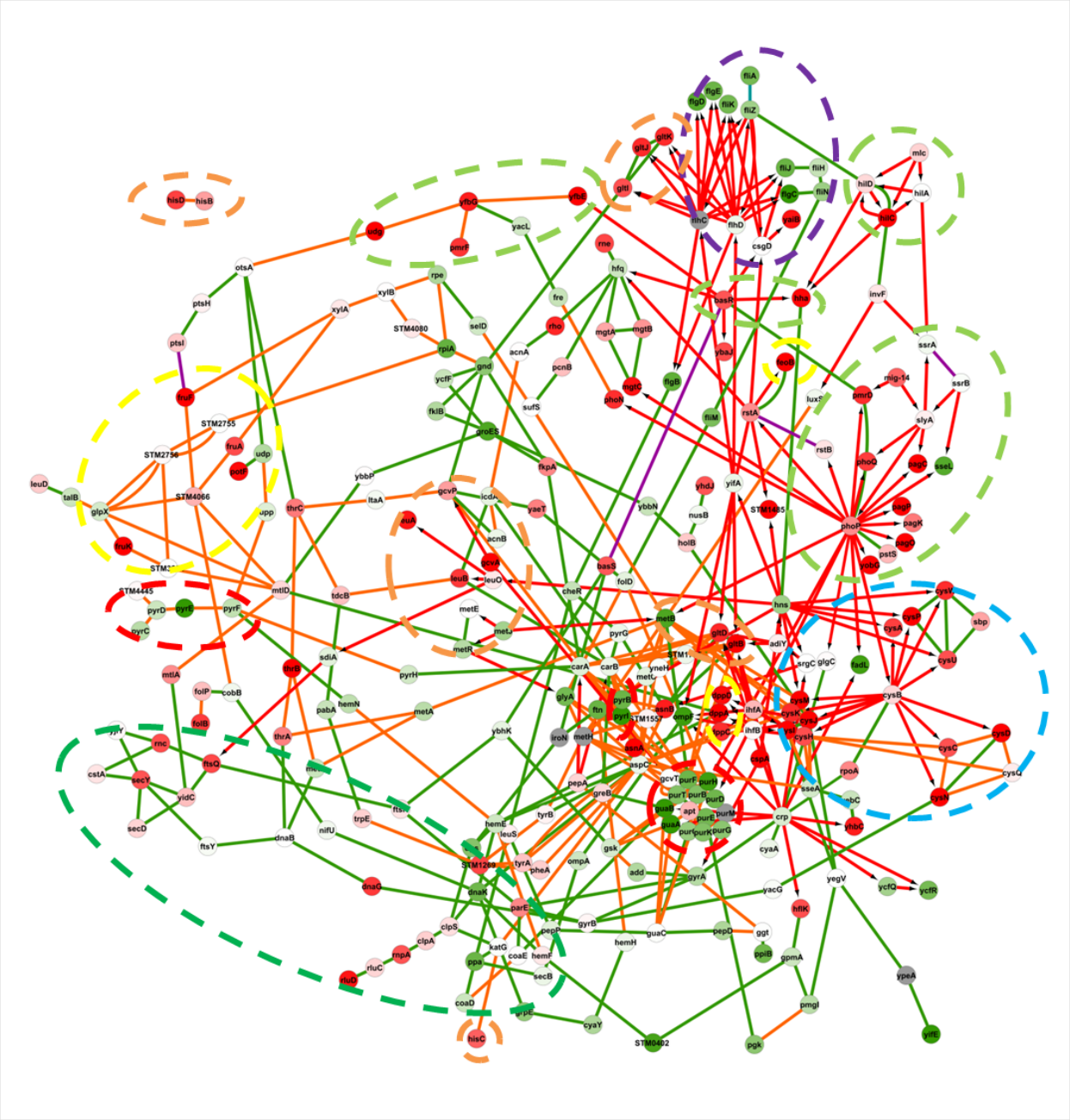
Deze sectie bevat de resulterende subnetwerken gegenereerd door PheNetic ter verklaring van de sterk differentieel geëxpresseerde genen na imidazoolbehandeling van *Salmonella* LT2.



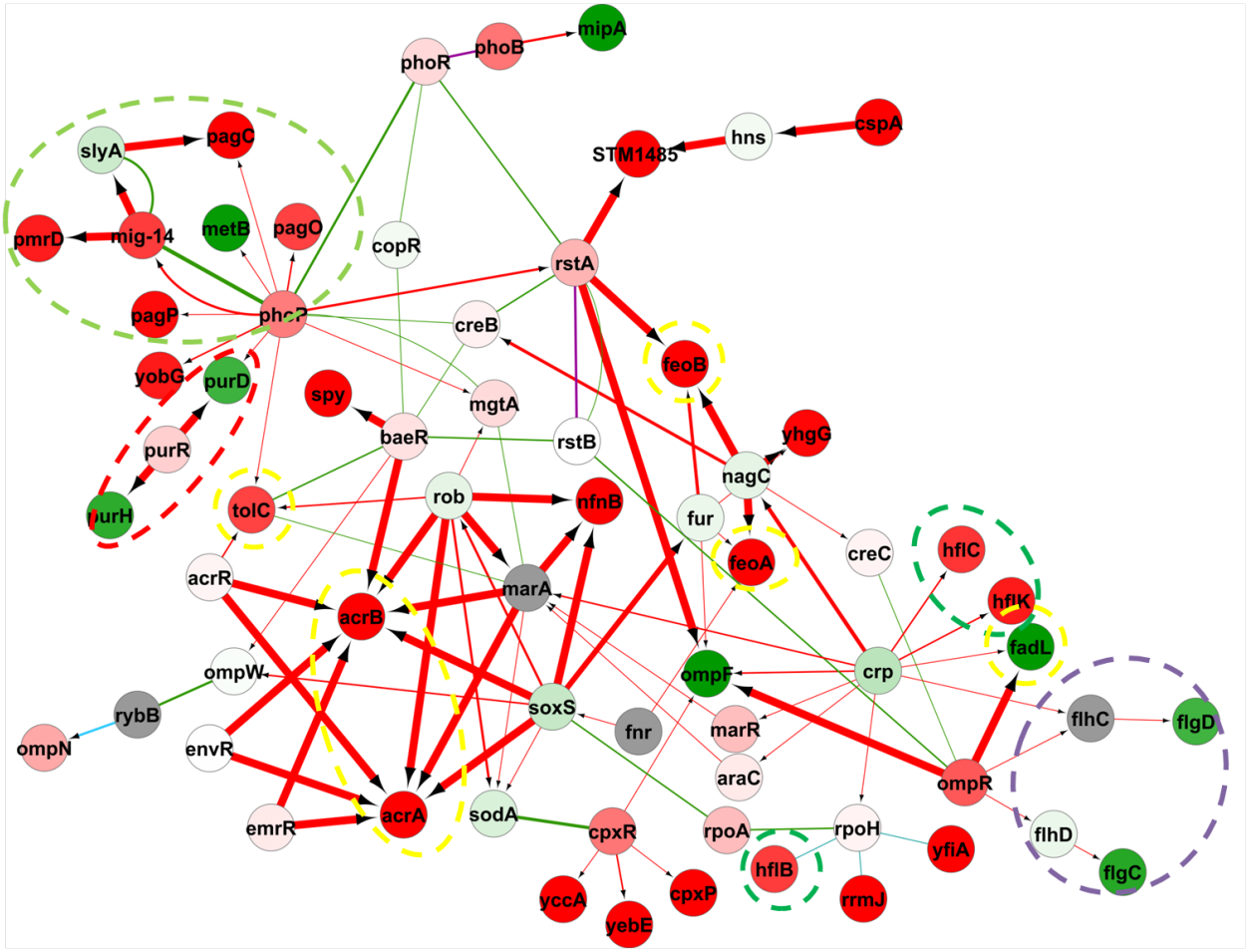
Figuur B.6. Verklaring van PheNetic voor de sterk differentieel geëxpresseerde genen in experiment imidazool_1 van de gevalstudie. Het netwerk bestaat uit 105 knooppunten en 143 verbindingslijnen.



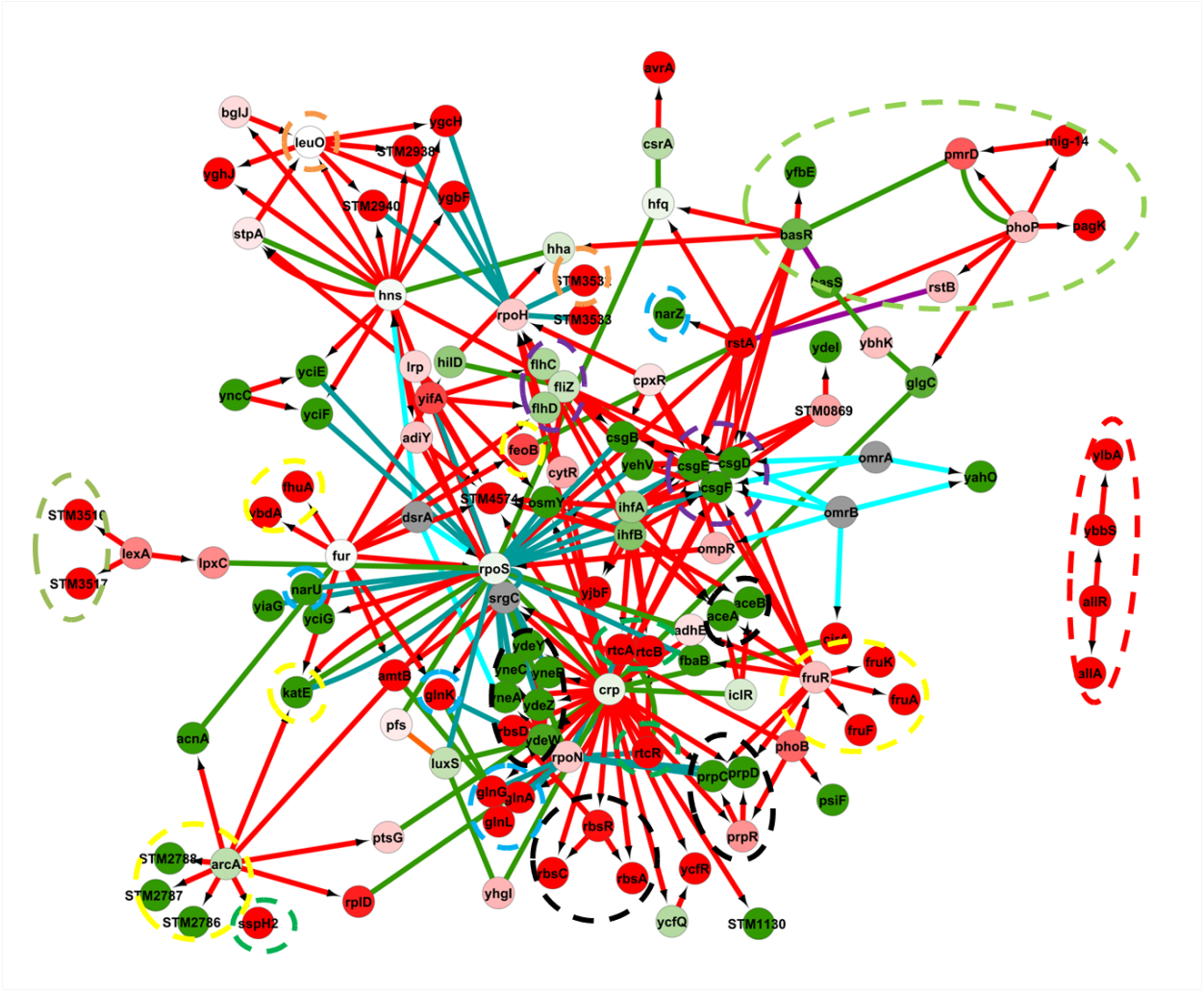
Figuur B.7. Verklaring van PheNetic voor de sterk differentieel geëxpresseerde genen in experiment imidazool_2 van de gevalstudie. Het netwerk bestaat uit 96 knooppunten en 159 verbindingslijnen.



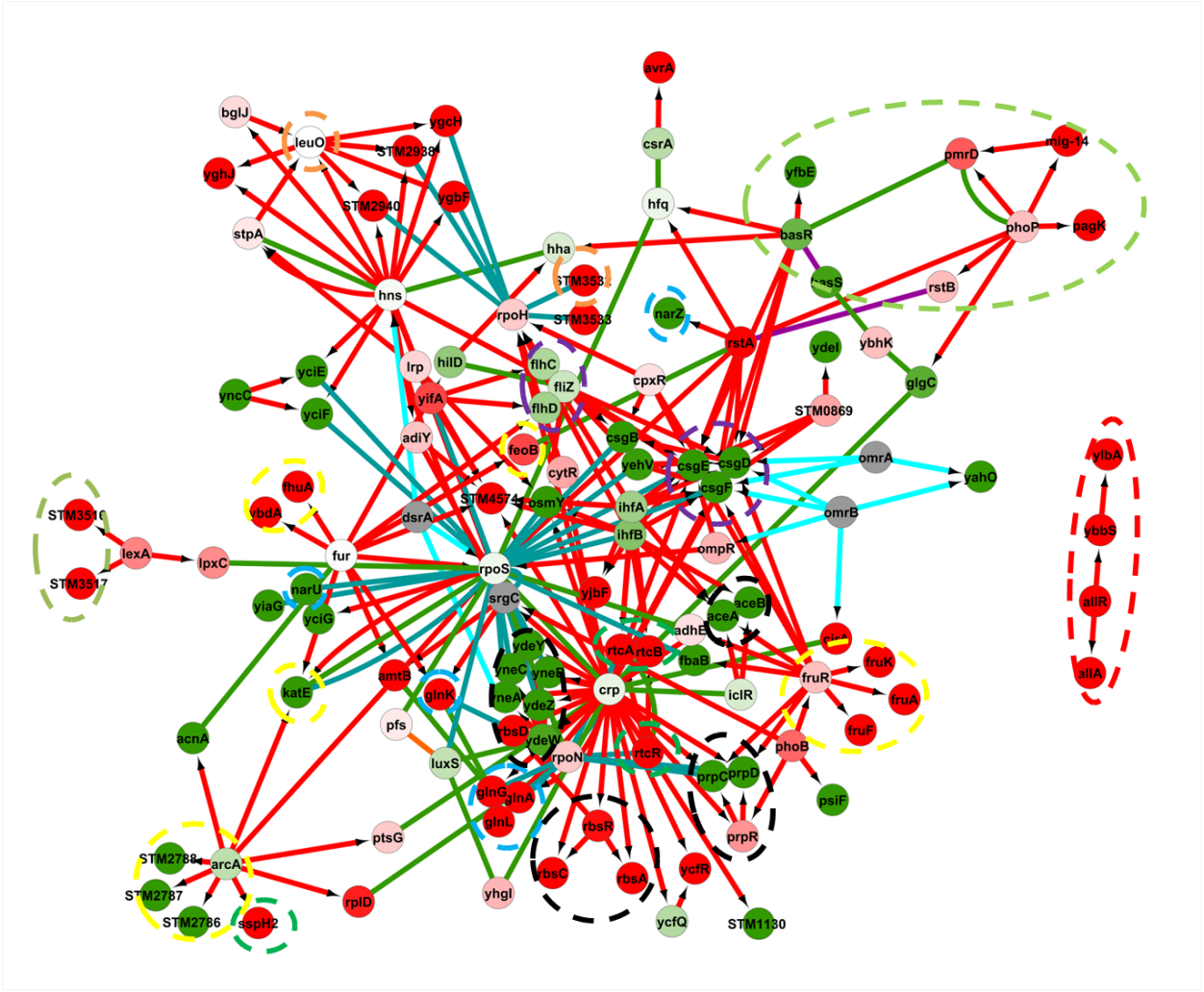
Figuur B.8. Verklaring van PheNetic voor de sterk differentieel geëxpresseerde genen in experiment imidazool_2–totaal van de gevalstudie. Het netwerk bestaat uit 258 knooppunten en 477 verbindingslijnen.



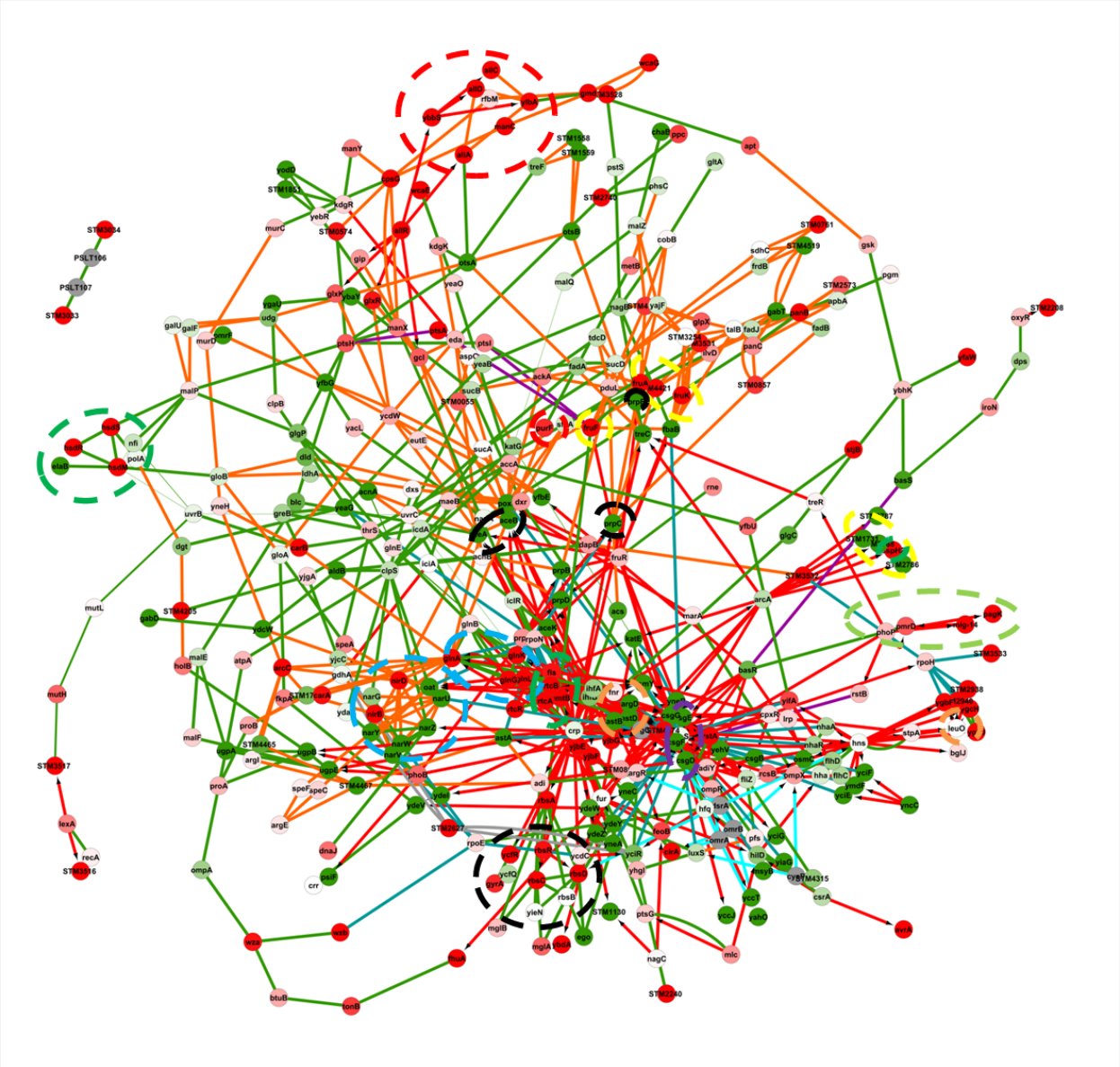
Figuur B.9. Verklaring van PheNetic voor de sterk differentieel geëxpresseerde genen in experiment imidazoline van de gevalstudie. Het netwerk bestaat uit 67 knooppunten en 109 verbindingslijnen.



Figuur B.10. Verklaring van PheNetic voor de sterk differentieel geëxpresseerde genen in experiment sensitiviteit_1 van de gevalstudie. Het netwerk bestaat uit 124 knooppunten en 266 verbindingslijnen.



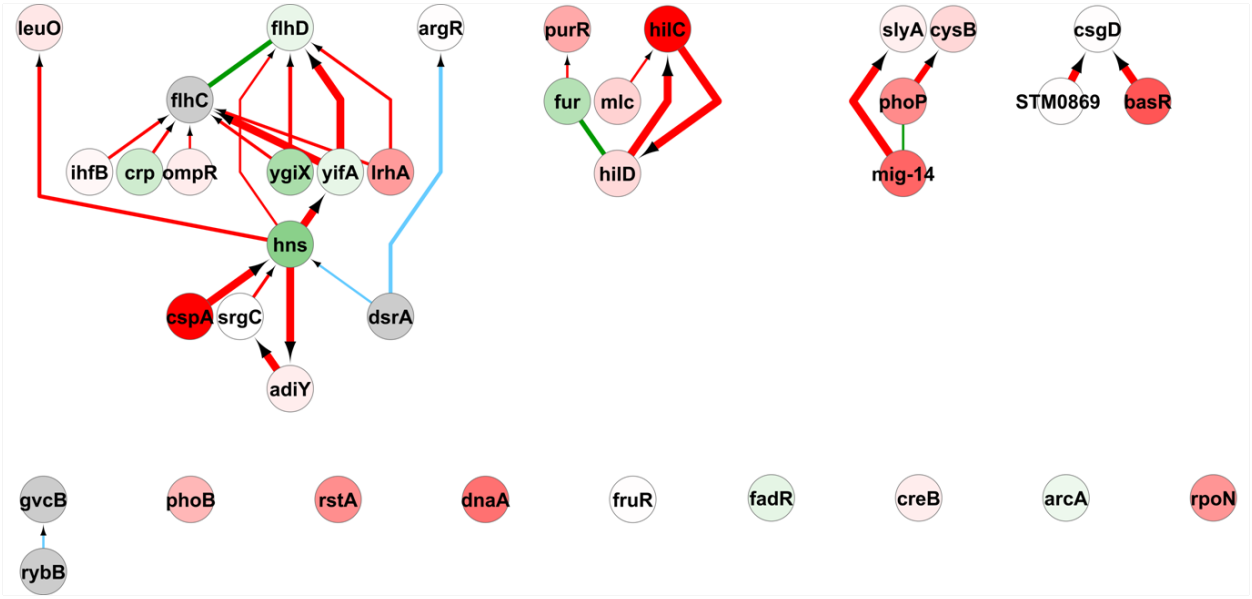
Figuur B.11. Verklaring van PheNetic voor de sterk differentieel geëxpresseerde genen in experiment sensitiviteit_2 van de gevalstudie. Het netwerk bestaat uit 170 knooppunten en 385 verbindingslijnen.



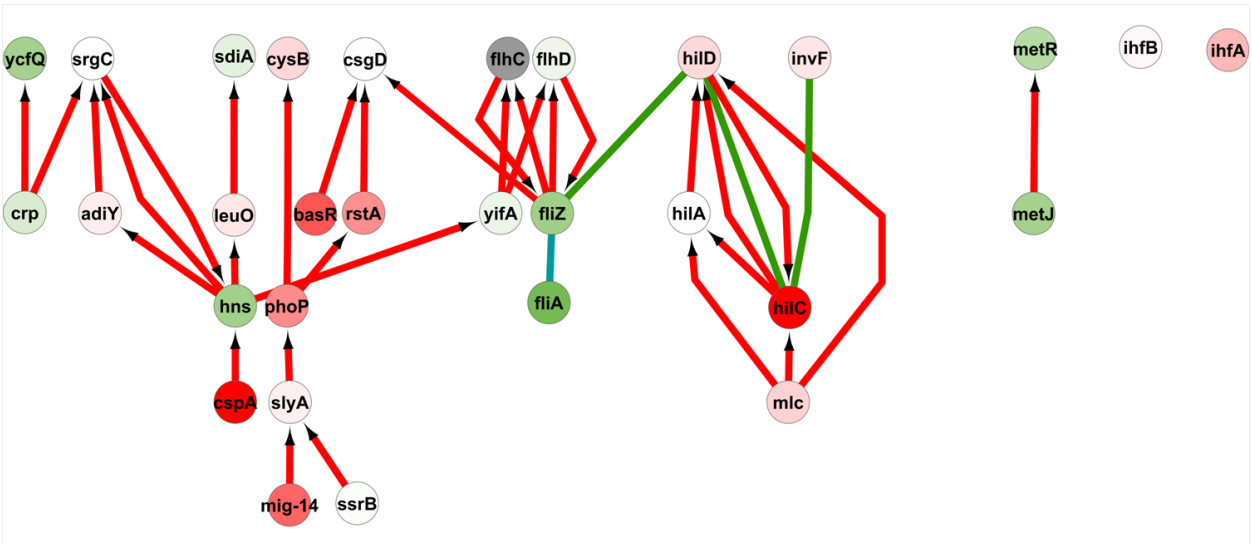
Figuur B.12. Verklaring van PheNetic voor de sterk differentieel geëxpresseerde genen in experiment sensitiviteit_2–totaal van de gevalstudie. Het netwerk bestaat uit 346 knooppunten en 774 verbindingslijnen.

B.3 Visualisatie regulatoren

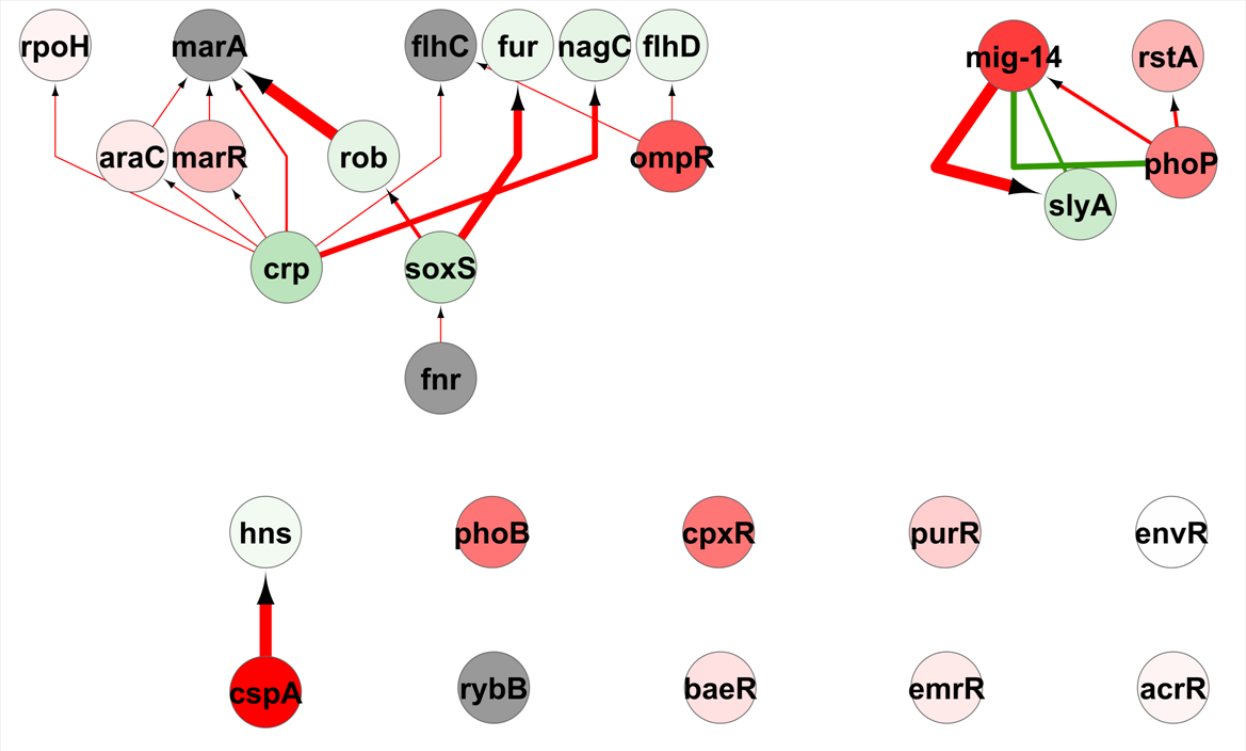
De gevonden regulatoren interageren ook met elkaar. Deze sectie geeft daartoe de regulatie van de reactie van *Salmonella* LT2 op de betreffende imidazoolbehandeling.



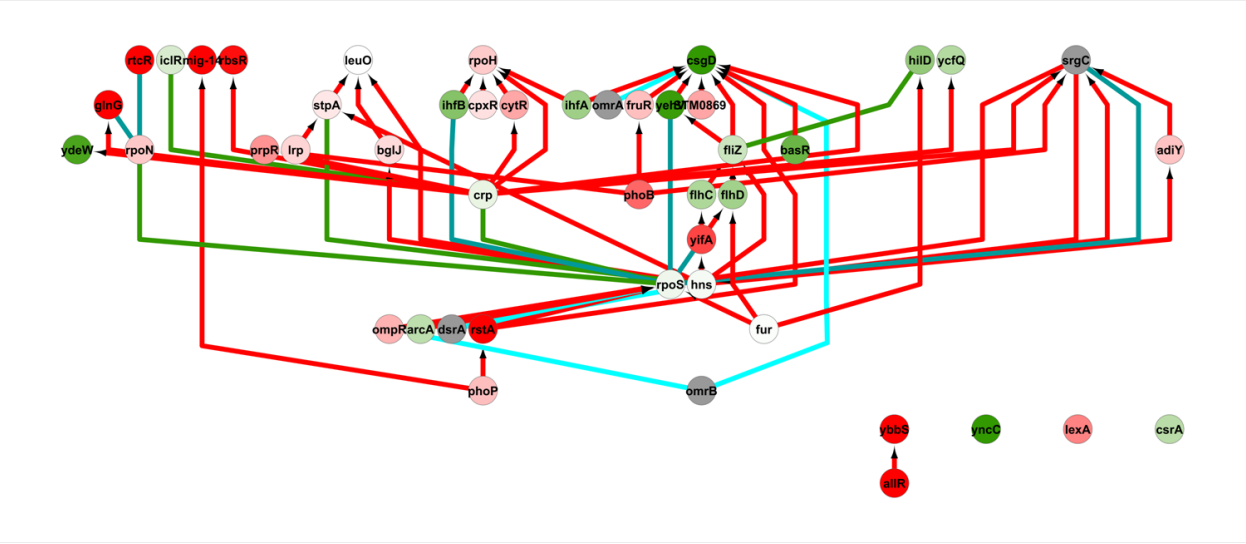
Figuur B.13. Onderlinge samenhang van de regulatoren in experiment imidazool_1 van de gevalstudie. Het netwerk bestaat uit 37 knooppunten en 30 verbindingslijnen.



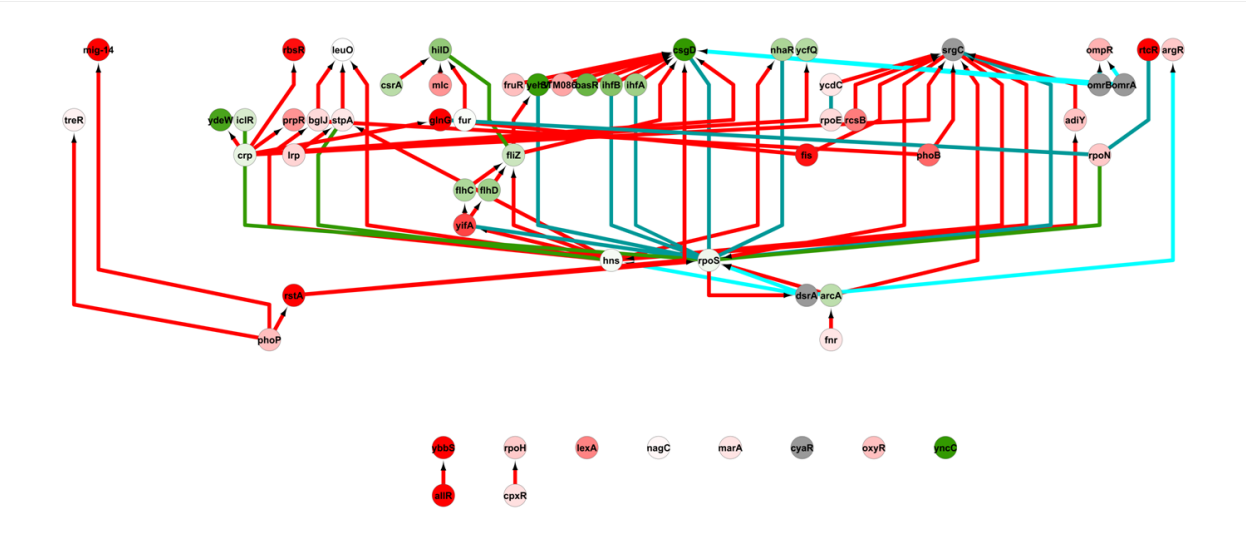
Figuur B.14. Onderlinge samenhang van de regulatoren in experiment imidazool_2 van de gevalstudie. Het netwerk bestaat uit 30 knooppunten en 36 verbindingslijnen.



Figuur B.15. Onderlinge samenhang van de regulatoren in experiment imidazoline van de gevalstudie. Het netwerk bestaat uit 30 knooppunten en 36 verbindingslijnen.



Figuur B.16. Onderlinge samenhang van de regulatoren in experiment sensitiviteit_1 van de gevalstudie. Het netwerk bestaat uit 47 knooppunten en 66 verbindingslijnen.



Figuur B.17. Onderlinge samenhang van de regulatoren in experiment sensitiviteit_2 van de gevalstudie. Het netwerk bestaat uit 59 knooppunten en 75 verbindingslijnen.

Vulgariserende samenvatting

Hedendaagse experimenten in het laboratorium laten toe grote hoeveelheden biologische gegevens te verzamelen en een groot deel hiervan is publiek beschikbaar. Het kan interessant zijn nieuwe bekomen resultaten te bestuderen in het licht van deze publiek beschikbare kennis om zo nieuwe inzichten te verwerven. Al deze gegevens handmatig doorzoeken neemt echter veel tijd in beslag, zodat computertechnieken die al deze informatie bundelen en kunnen doorzoeken, nuttige hulpmiddelen zijn. De ontwikkeling en toepassing van dergelijke computertechnieken om biologische gegevens te doorzoeken, behoren tot het domein van de bio-informatica. Een specifiek onderdeel hiervan, systeembiologie genaamd, bestudeert de interacties die optreden tussen de verschillende componenten die aanwezig zijn in een bepaald(e) cel/organisme.

Deze studie past de computertechniek PheNetic, die publieke interactiegegevens kan doorzoeken, toe om een gemeenschappelijke oorzaak te achterhalen voor effecten geobserveerd na de behandeling van een *Salmonella*-bacterie met een imidazool-antibioticum. Hiertoe worden publieke interactiegegevens voorgesteld als een netwerk waarin elk punt een celcomponent voorstelt en elke verbinding een interactie tussen de verbonden punten. Experimenten kunnen genen – discrete eenheden van erfelijk materiaal – identificeren in *Salmonella* die door de imidazoolbehandeling in verschillende mate – ten opzichte van geen imidazoolbehandeling – vertaald worden in eiwitten. PheNetic probeert deze teruggevonden genen te verbinden op het gedefinieerde netwerk om zo een gemeenschappelijke oorzaak te achterhalen.

Deze techniek bleek enerzijds succesvol in het terugvinden van reeds gekende oorzaken die beïnvloed worden door imidazoolbehandeling. Anderzijds werden ook nieuwe mogelijk interessante ontdekkingen gedaan. Bovendien kon deze studie ook de invloed van verschillende instellingen van PheNetic karakteriseren en de biologische correctheid van de bekomen resultaten bewijzen.