Identifikation af potentielle microRNA gener ved hjælp af komparativ genomanalyse

Per Tøfting

11. juli 2008

Speciale i softwarekonstruktion
IT-Vest
Aarhus Universitet

Indhold

Figurer

Tabeller

1 Indledning

microRNAer (miRNAer) er endogene ikke-kodende RNAer, som ca. 21 nukleotider lange, og er blevet opdaget inden for de seneste år. De spiller en vigtig rolle i gen reguleringen i planter og dyr. Der er to primære fremgangsmåder til at at opdage nye miRNAer på, som er eksperimentel- og computerbaseret identifikation.

Eftersom nogle miRNAer er udtrykt ved et lavt niveau, og mange kommer til udtryk på forskellig tidspunker og i forskellige celletyper, kan det være svært at finde dem eksperimentelt.

Hvorimod computerbaserede identifikation er uafhængig af niveau, tidspunkt og sted for miRNA aktivitet, hvilket gør fremgangsmåden velegnet til at finde potentielle miRNAer. De er dog kun potentielle miRNAer, de skal efterfølgende verificeres eksperimentelt, men den eksperimentelle søgning efter dem bliver mere målrettet, når man har en ide om, hvor man skal lede.

1.1 Formål

Dette speciale projekt har til formål at udvikle en pipeline, der gennem komparativ genomanalyse af tre arters genomer kan finde DNA-sekvenser, der indeholder potentielle miRNA gener.

Om de fundne DNA-sekvenser rent faktisk koder for miRNAer skal afgøres eksperimentelt, som er uden for dette projekts område.

Den udviklede pipeline vil bestå af egenudviklet programmer og allerede eksisterende programmer.

1.2 Specialets opbygning

Resten af rapporten er opbygget på følgende måde:

2 Biologien: De relevante biologiske begreber bliver beskrevet i dette afsnit.

3 microRNA: Projektet er rettet mod miRNA gener, de er nærmere beskrevet i dette afsnit.

4 Strategien: Strategien beskriver fremgangsmåden, der er valgt i dette projekt, til at finde potentielle miRNA gener. Det forklares i afsnittet, hvorfor den valgte strategi er foretrukket fremfor andre.

5 Pipelinen: Pipelinen er implementationen af den valgte strategi. Hvordan det er gjort er beskrevet i dette afsnit.

6 Programmerne: Pipelinen består dels af eksisterende- og nyudviklede programmer, sidstnævnte er beskrevet i dette afsnit.

7 Test: Pipelinen er blevet testet, for at undersøge, hvor effektiv den er, til at finde potentielle miRNA gener. Resultatet af testen er beskrevet i dette afsnit.

Konklusion: Resultatet af projektet sammenfattes i dette afsnit.

Fil formater: Der benyttes flere fil formater, de er beskerevet i dette afsnit.

1.3 Hjemmeside for specialet

Til specialet er opretten en hjemmeside, hvor de udviklede programmer findes og kan downloades, samt links til de programmer, der er benyttet i forbindelse med specialet, og diverse andre links, med relevans til specialet. Desuden findes der datasæt og resultater fra testen. En pdf udgave af denne rapport er der også.

Hjemmesiden findes på adressen `http://pertoefting.dk/swk/speciale/`

2 Biologien

For at kunne forstå den resterende del af rapporten beskrives her kort de relevante biologiske begreber.Afsnittet er hovedsageligt baseret på, hvis ikke andet er nævnt Kjeldsen and Nørby (2006), Jensen and Prentø (2003) og Nørby (2003).

De levende organismer deles op i tre hoved grupper eller domæner: Bakterier (Bacteria), arkebakterier (Archhaea) og eukaryote (Eucaria) (Woese et al., 1990).

De to første domæner tilhører prokaryoterne, der er encellede organismer og deres arvemateriale er ikke adskilt fra resten af cellen, den ligger frit i cytoplasmaet.

Det sidste domæne eukaryoterne er en- eller flercellede organismer, hvor den overvejende del af arvematerialet findes i en cellekerne, som er omgivet af en cellemembran, og derfor adskilt fra cytoplasmaet. Mitokondrier og grønkorn har deres egen arvemateriale hos eukaryoterne. Mitokondrierne har et cirkulært kromosom, som findes i flere kopier i hver mitokondrie.

2.1 Genomet

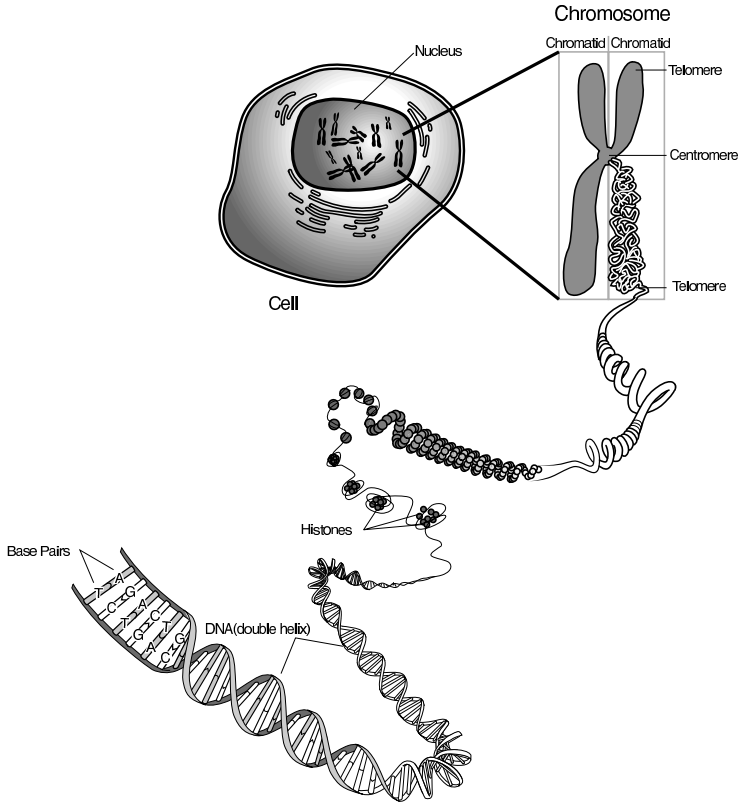
En organismes samlede arvemateriale betegnes genomet. Genomet er organiseret som kromosomer. Prokaryote har ofte en cirkelformet kromosom. Eukaryote har arvematerialet fordelt på flere lineære kromosomer, og mitokondriernes kromosom er cirkulært, der findes i flere kopier i hver mitokondrie.

Arvematerialet består af DNA (deoxyribonuklinsyre), dog benytter enkelte viruser RNA (ribonuklinsyre) som arvemateriale. Det er DNAet, der indeholder den information, som er nødvendigt for at opbygge et hvert protein eller funktionel RNA, som er nødvendig for at organismen kan danne og vedligeholde sine celler, væv og organer.

Alle celler i en organisme indeholder den samlede arvemateriale, derfor sker der ved hver celledeling en kopiering af arvematerialet, således at døtrecellerne hver får den samlede arvemateriale.

Normalt er kromosomerne fordelt over hele kernen, men under celledeling sker der en kraftig kondensering, hvor kromosomerne er opbygget af to identiske strenge, kromatider, som er forbundet i et enkelt punkt, centromeren. Hver kromatid indeholder et langt lineært dobbeltstrenget DNA-molekyle, der er bundet til proteiner primært histoner. Det er DNA-molekylet der er arvematerialet. De to DNA-molekyler er identiske, på nær eventuelle mutationer, og ender i hver sin celle efter celledelingen. Figur 1 viser et eukaryot kromosoms opbygning.

Genomet kan deles ind i forskellige elementer, se figur 2. Hos mennesket består kun ca. 1,5% af genomet af protein-kodende exons. 27% består af introns og untranslaterede regioner (UTR) fra protein-kodende gener. 46%



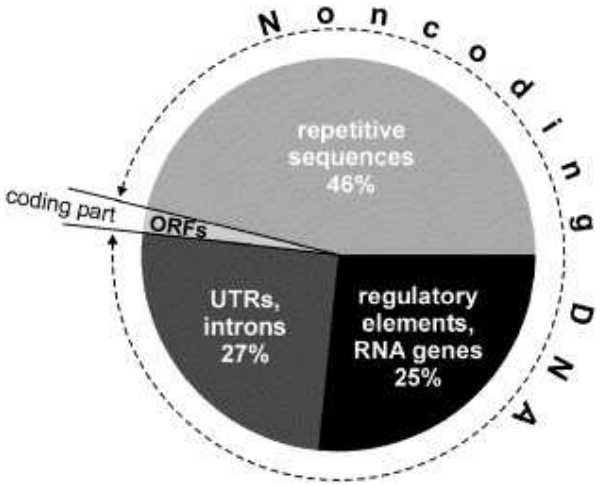
Figur 1: Opbygningen af et kromosoms hos en eukaryot organisme. De to kromatider går til hver sin celle i forbindelse med celledelingen. Fra Talking Glossary – chromosome, `http://www.genome.gov/glossary.cfm`.

består af repeterende sekvenser, og de sidste 25% er relativt ukendt, men består bl.a. af regulerende elementer og ikke-protein kodende gener (Szymanski et al., 2005). Det har vist sig, at jo mere komplekst et organisme er, jo minde del af genomet er protein-kodende gener.

De repeterende sekvenser kan indeles i to overordnede grupper: 1) interspersed repeats, hvis individuelle repeterende enheder er fordelt over hele genomet på en tilsyneladende tilfældig måde, 2) tandem-repeterende DNA hvis repeterende enheder ligger ved siden af hinanden.

2.2 DNA

Et DNA (deoxyribonuklinsyre) molekyle er et polynukleotid, bestående af en kortere eller længere uforgrenet kæde af nukleotider. Et nukleotid består af fosfat, en pentose og en af fire forskellige nitrogenholdige baser. To af de fire baser er pyrimidiner, cytosin (C) og thymin (T), og to er puriner, adenin



Figur 2: Genomet kan deles ind i forskellige elementer. Her vist for mennesket. Fra Szymanski et al. (2005).

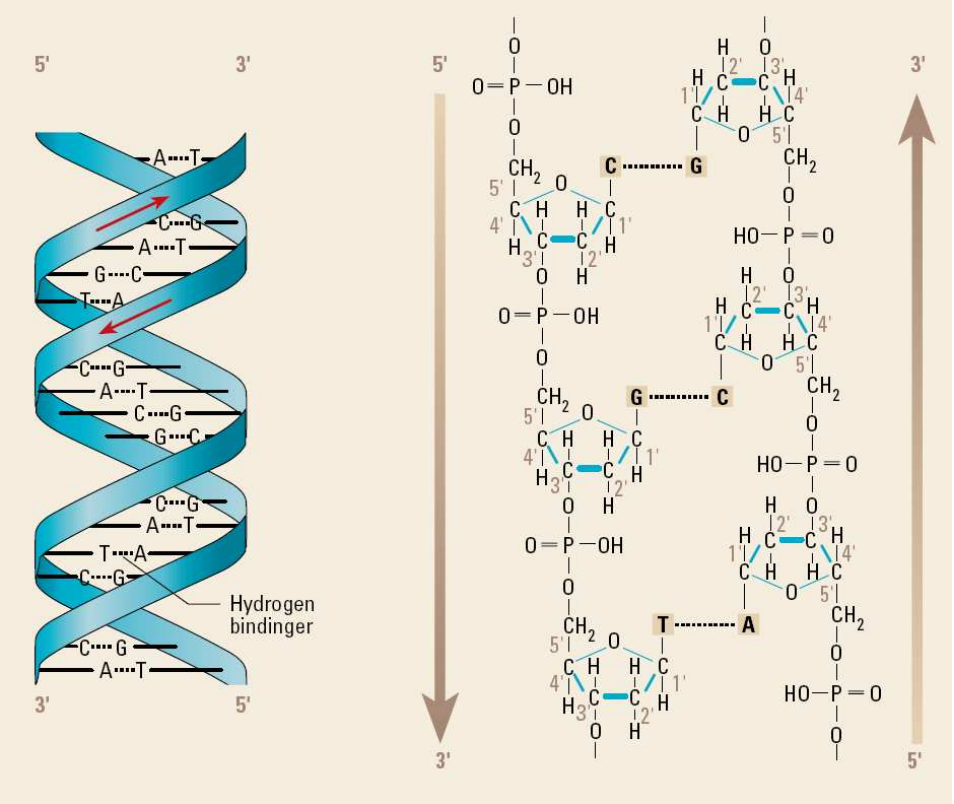
(A), guanin (G). Pentosen er 2'-deoxyribose.

Nukleotidernes rygrad består af skiftevis pentose- og fosfatmolekyler, der er bundet sammen med diesterbindinger mellem fosfat og henholdsvis kulstof nr. 3 og nr. 5 på pentosen. Baserne er bundet til rygraden med en N-glycosidbinding til pentosens kulstof nr. 1, se figur 3.

Et polynukleotidet er orienteret, fosfatet i den første nukleotid i et polynukleotid indgår ikke i fosfatesterbinding, mens dens pentose-3'-OH-gruppe esterbinder med det følgende nukleotids 5'-fosfatgruppe. En nukleinsyre starter derfor med en fri fosfatgruppe (5'-enden) og slutter med en fri 3'-OH gruppe (3'-enden). Synteseretningen for en nukleinsyre går altid fra 5' til 3' (5' $\rightarrow$ 3').

Normalt er DNA dobbeltstrenget, opbygget af to strenge, der er spiralsnoet om hinanden og danner en højredrejet dobbelthelix. De to strenge er forbundet via hydrogenbinder mellem baserne, der er beliggende over for hinanden, og danner på den måde molekylets basepar. De to strenge er modsat orienterede, for at baserne kan få den rette orientering i forhold til hinanden til at danne indbyrdes hydrogenbindinger, strengene er antiparallelle i forhold til hinanden (5' $\rightarrow$ 3'/3' $\leftarrow$ 5').

I dobbeltstrenget DNA dannes hydrogenbindingerne altid efter et bestemt mønster pga. basernes kemiske struktur, adenin danner altid par med thymin og guanin altid med cytosin. Der dannes to hydrogenbindinger mellem adenin og thymin, mens der dannes tre hydrogenbindinger mellem guanin og cytosin, se figur 4. Dette betyder at base sekvenserne i de to DNA-strenge er omvendt komplementære, sekvensen af den ene streng kan altid fastlægges ud fra den anden streng.



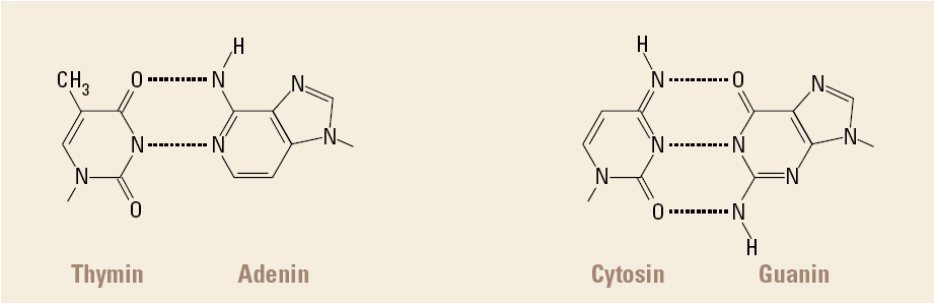
Figur 3: DNA-molekylets struktur. Til venstre er vist en DNA-dobbelthelix, hvor de to DNA-molekyler er snoet om hinanden og danner en højredrejet spiral, der holdes sammen af hydrogenbinder mellem baserne. Til højre er vist DNA-molekylernes rygrad bestående af skiftevis 2'deoxyribose- og fosfatmolekyler, og baserne som er bundet til deoxyribosen. Pilene viser orienteringen af de to antiparallelle DNA-strenge. Fra Kjeldsen and Nørby (2006).

DNA-dobbelthelixen en ensartet struktur, hvor det eneste der varierer, er rækkefølgen af baserne. Det er rækkefølgen af A, C, G og T, der koder for den arvelige information i generne. DNA indgår i to processer replikation og genekspression.

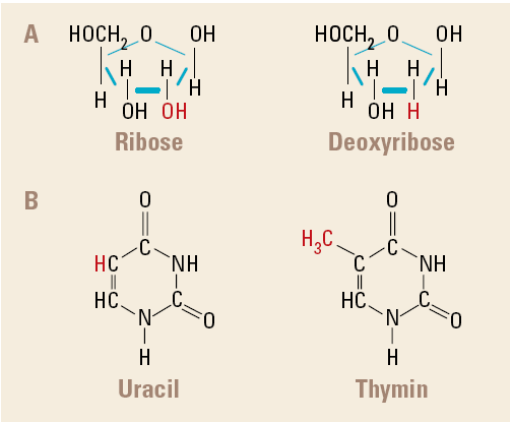
2.3 RNA

RNA-molekyler er meget lig DNA-molekyler. Dog er pentosen ribose i stedet for deoxyribose og pyrimidinen uracil (U) indgår i stedet for thymin, se figur 5. Uracil danner hydrogenbindinger med adenin på samme måde som thymin.

RNA-molekyler er enkeltstrenget, og danner ikke dobbelthelix som DNA-



Figur 4: DNA-molekylets basepar, der dannes to hydrogenbindinger mellem thymin (T) og adenin (A) og tre hydrogenbindinger mellem cytosin (C) og guanin (G). Fra (Kjeldsen and Nørby, 2006).



Figur 5: Strukturelle forskelle mellem RNA og DNA. A. RNA indeholder ribose i stedet for deoxyribose. B. RNA inderholder basen uracil (U) i stedet for thymin (T). Fra Kjeldsen and Nørby (2006).

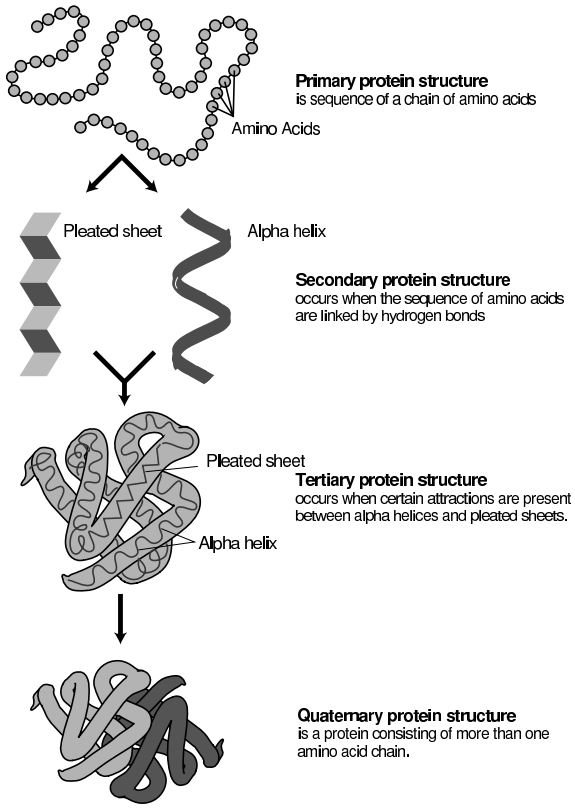
molekyler, til gengæld kan dele af et RNA-molekyle bindes med hydrogenbindinger til andre dele af sig selv, hvor der er komplementaritet, og derved danne dobbeltstrenget strukturer. Disse er adskilt af regioner uden komplementære baseparring, hvilket medfører dannelse af komplicerede tre dimensionelle foldestrukturer. Derfor er RNA-molekylers tre dimensionelle struktur meget mere varieret end for DNA.

DNA har essentielt kun en funktion at kode for den genetiske information, mens RNA findes i flere typer, der udfører forskellige funktioner.

2.4 Protein

Et protein er et polypeptid og er opbygget af mindre molekyler, aminosyrer. Et aminosyre består af en central kulstofatom, hvortil der er bundet et hydrogen atom, en carboxylsyregruppe (-COOH), en aminogruppe (-NH2) og en side kæde. Det er sidekæden der adskiller de forskellige aminosyrer. De enkelte aminosyre molekyler bindes sammen af peptid bindinger (-CO-NH-) ved at carboxylsyregruppen i en aminosyre bindes til aminogruppen i den næste aminosyre, og derved danner proteinets rygrad. Proteiner også orienterede, i den ene ende er der en fri aminogruppe (N-terminalen) og i den anden ende en fri carboxylsyregruppe (C-terminalen). Tyve forskellige slags aminosyrer indgår i protein.

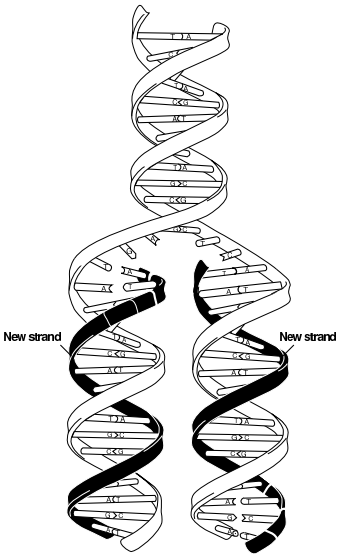
Proteiner foldes sammen i meget specifikke tre dimensionelle strukturer, se figur 6. Det er aminosyrenes rækkefølge, der bestemmer proteinets struktur, som derfor også bestemmer dets funktion.



Figur 6: Protein struktur. Fra Talking Glossary – protein, `http://www.genome.gov/glossary.cfm`.

2.5 DNA replikationen

DNA replikationen er den proces hvorved et DNA-molekyle kopieres, på sådan en måde at kopien er magen til originalen. Replikationen sker før hver celledeling, således at hver af dattercellerne har en komplet kopi af arvematerialet. Processen starter ved at DNA-molekylets to strenge går fra hinanden, således at de begge kan fungere som skabelon for dannelsen af en ny komplementær streng, som den bindes sammen med. Dette bevirker at hver af de nye DNA-molekyler består af en oprindelig og en ny streng, se figur 7. Syntesen af de nye DNA-strenge foregår altid i retningen 5' → 3'.



Figur 7: Ved DNA replikationen dannes to identiske dattermolekyler pga. baseparrings reglerne. Fra Talking Glossary – DNA replication, `http://www.genome.gov/glossary.cfm`.

2.6 Gen

Et arveanlæg eller gen er et sammenhængende stykke DNA der indeholder informationen til at bygge et protein eller funktionel RNA molekyle.

Genstrukturen hos prokaryoter er relativt simpel, den består af en promotor region, der bestemmer hvornår et gen udtrykkes og en kodende region der indeholder DNA sekvensen for genet. Eukaryoters genstruktur er mere kompliceret, de har også promoter regioner men de er ofte større og kan ligge langt fra den del af genet der transkriberes. Desuden har mange gener enhanceres og suppressors, som er DNA sekvenser, der er med til at regulere gen ekspressionen, og de kan ligge flere kilobaser væk fra den transkriberede del

af genet. Generne hos eukaryoter er ofte delt op i kodende sekvenser, kaldet exons, son er adskilt af ikke-kodende sekvenser, kaldet introns, se figur 8 .

2.7 Gen ekspression

Når et proteinkodende gen udtrykkes, sker det ved to processer der følger efter hinanden, henholdsvis transkription og translation. Hvis det er et gen, der koder for en funktionel RNA molekyle, er det kun transkriptionen, der foregår.

Transkriptionen

Transkriptionen er den proses, hvorved et gens ene DNA-streng anvendes som skabelon for syntetiseringen af et enkeltstrenget RNA-molekyle, med en basesekvens, der er omvendt komplementær til den pågældende DNA-strengen. RNA-molekylets basesekvens bliver derved identisk med genets anden DNA-streng på nær at T er udskiftet med U i RNAet. Denne DNA-streng kaldes den kodende streng eller sense strengen. DNA-strengen der fungerer som skabelon for syntetiseringen af RNA-molekylet, kaldes ikke-kodende streng eller antisense strengen.

Transkriptionen af et gen foregår altid i retningen 5' $\rightarrow$ 3', således at RNA forlængelsen foregår i den ende, som har en fri 3'-OH gruppe. Nogle gener i et kromosom kodes fra den ene streng mens andre kodes fra den anden streng.

RNA-molekyler kan deles ind i to adskilte klasser, messenger RNA (mRNA) som translateres til proteiner, og funktionelle RNA som fungerer på RNA niveau.

RNA-molekylerne hos eukaryote organismer bliver ofte udsat for post-transkriptionelle modifikationer. Det primære RNA-transkript fra protein-kodende gener får i 5'-enden påsat en såkaldt 5'cap, i 3'enden påsættes en poly(A)-hale. Introns, de ikke kodende dele, bliver spejet ud af RNA-molekylet, hvorved exons samles til en protein kodende sekvens, selve messengerRNA-molekylet. Mange RNA-molekyler fra proteinkodende gener undergår såkaldt alternativ splejsning, hvorved der dannes mRNA-molekyler med forskellig kombinationer af exons. Fra et givet gen kan der, derfor dannes forskellige proteiner.

Translation

Translationen er betegnelsen for syntetiseringen af proteiner udfra mRNAer. Hos eukaryote skal mRNAerne først transporteres ud af cellekernen til cytoplasmaet. Hvor protein syntetiserende partikler kaldet ribosomer oversætter mRNAets information til protein. Oversættelsen foregår vha. den genetiske kode, hvis enheder benævnes kodoner, der hver består af tre på hinanden følgende baser. Hver kodon svarer til en aminosyre. Se tabel 1.

1. base	2. base				3. base
(5'-ende)	U	C	A	G	(3'-ende)
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	stop	stop/Sec	A
	Leu	Ser	stop	Tpr	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	start/Met	Thr	Lys	Arg	G
G	Val	Aln	Asp	Gly	U
	Val	Aln	Asp	Gly	C
	Val	Aln	Glu	Gly	A
	Val	Aln	Glu	Gly	G

Tabel 1: Den genetiske kodetabel. Den genetiske kodes enheder betegnes kodoner. En kodon består af tre på hinanden følgende baser, en triplet, i det pågældende mRNA. Den viste genetiske kode er standart koden.

Fordi der er tre baser i hver kodon, og der er fire forskellige baser, er det muligt at danne 64 forskellige kodoner, men der er kun tyve forskellige aminosyrer, derfor koder flere forskellige kodoner for den samme aminosyre. Man siger derfor også, at den genetiske kode er degenereret.

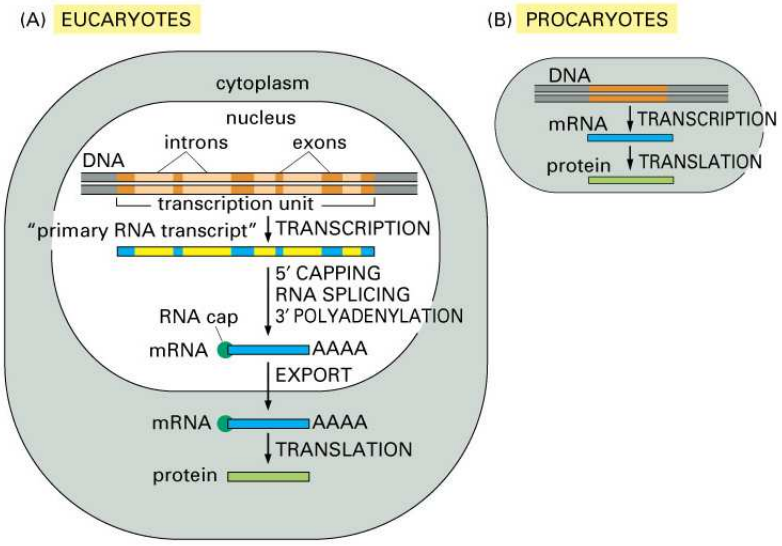
Det er ikke hele mRNAet der bliver translateret. Translationen begynder med kodonet AUG. Tre kodoner koder ikke for nogen aminosyre det er UAA, UAG og UGA, de fungere som stopkodoner. Når et ribosom møder et stopkodon stoppes translationen og proteinet frigives.

Hos langt de fleste arter anvendes den samme genetiske kode, men der findes arter med variationer i den genetiske koden. Mitokondrier har også en lidt anderledes genetisk kode.

Forskellen mellem proteinsyntesen hos henholdsvis prokaryote og eukaryote er vist i figur 8.

2.8 Funktionelle RNAer

Funktioneller RNAer betegnes ofte som ikke-kodende RNAer eller non-coding RNAer (ncRNAer), fordi de ikke translateres til proteiner.



Figur 8: Protein syntese hos henholdsvis eukaryoter (A) og prokaryoter (B). Fra `http://www.accessexcellence.org/RC/VL/GG/ecb/gene_to_protein.php`.

I mange år var opfattelsen, at der kun var få funktionelle RNAer, og de blev betragtet som hjælpe komponenter i protein syntesen, det drejer sig bl.a. om tranfer RNA (tRNA) og ribosomal RNA (rRNA) involveret i translationen, small nuclear RNA (snRNA) involveret i pre-mRNA splicing og small nuleolar RNA (snoRNA) involveret i modifikation af rRNA. (Szymanski et al., 2005; Mattick and Makunin, 2006).

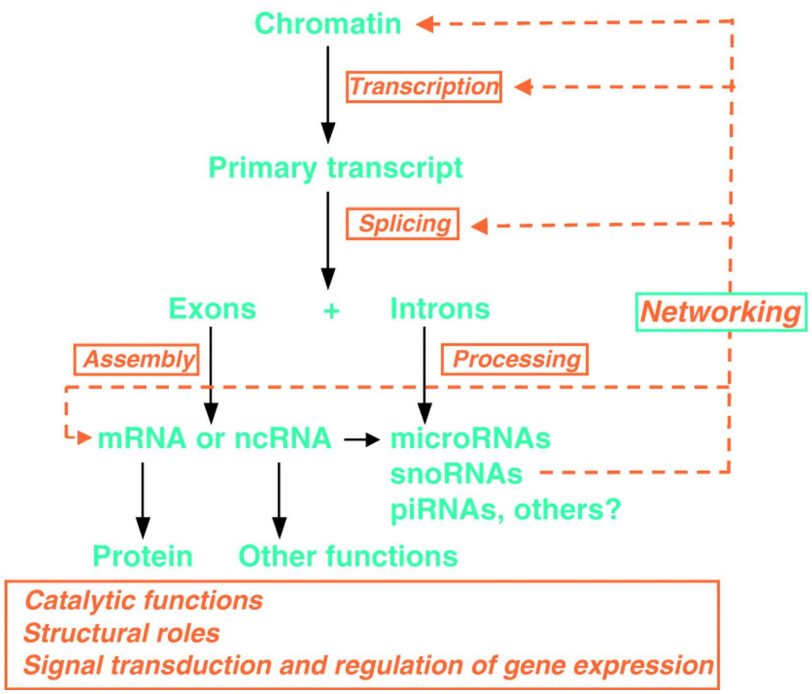
Inden for de seneste år er det blevet klart, at der er et stigende antal af andre funktionelle RNAer, som regulerer gen ekspressionen på forskellig måde i komplekse organismer, bl.a. microRNA (miRNA), small nuleolar RNA (snoRNA) og Piwi-interacting RNA (piRNA) (Mattick, 2007).

Dette har vist, at flowet af gen information er mere kompleks end hidtil antaget, se figur 9. En af disse er microRNAer, som er emnet for dette speciale.

2.9 Evolution

Evolutionen er den proces der former en arts udvikling og tilpasning til omgivelserne over tid. Evolutionen kan forklares ud fra tre mekanismer: naturlig selektion, mutationer og genetisk drift (Korf et al., 2003).

Den naturlige selektion består af tre antagelser:



Figur 9: Gen informations flowet hos højere eukaryoter. Det primære transkript kan blive (alternativt) splejset og yderligere behandlet til at danne en række protein isoformer eller funktionelle RNAer af forskellig type, som er involveret i en kompleks netværk af strukturelle, funktionelle og regulerende interaktioner. Fra Mattick (2007).

- Individerne inden for en art varierer
- Variationen skal være arvelig
- Variationen betyder, at nogle individer klarer sig bedre og får mere afkom end andre

Hvis de egenskaber, hos de individer der klare sig bedre og får mere afkom er arvelig, vil arveanlæggene for disse egenskaber med tiden blive mere dominerende hos arten, fordi afkommet, som har arvet egenskaberne, også får mere afkom.

Variationen mellem individerne skabes gennem mutationer. En mutation er en ændring i arvematerialet. De fleste mutationer sker i forbindelse med DNA-replikationen. Mutationer kan opstå i såvel legems celler (somatiske mutationer) som kønsceller (gametiske mutationer), men kun mutationer i kønsceller kan nedarves til de efterfølgende generationer, og kan på den måde resultere i nye karaktertræk.

Mutationer er grundlaget for al variation i arvematerialet og dermed en forudsætning for, at evolutionen kan finde sted.

Der findes flere forskellige typer af mutationer, se også figur 10:

Substitution: Ændring at en base til et af de andre tre baser.

Insertion: Indsættelse af en eller flere baser i et DNA-molekyle.

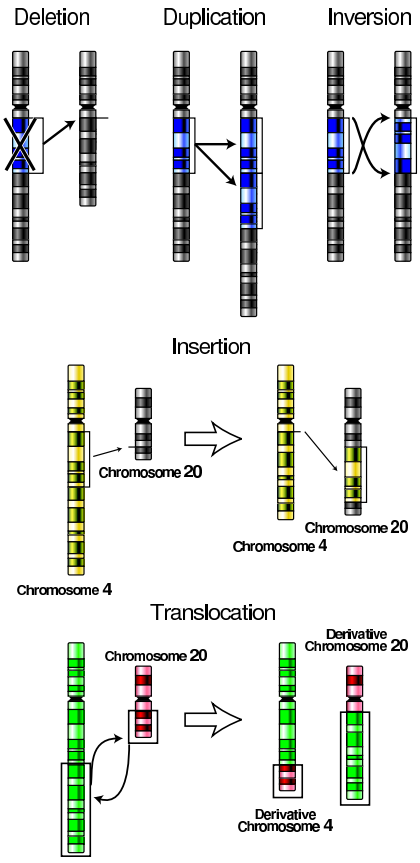
Deletion: Tab af et eller flere baser i DNA-molekylet, eller hvor en større eller mindre del af kromosomet går tabt.

Translokation: Flytning af et kromosom segment til et andet sted enten på samme kromosom eller på et andet kromosom.

Dublikation: Et kromosom segment kopieres og indsættes i det samme kromosom.

Inversion: Et kromosom segment roteres 180, således at DNA-sekvens rækkefølgen i det pågældende segment bliver den omvende af det normale.

Mutationerne er tilfældige og skaber den tilfældige variation, som tilpasses til omgivelserne gennem den naturlige selektion.



Figur 10: Forskellige typer af mutationer. Fra Talking Glossary – mutation, `http://www.genome.gov/glossary.cfm`.

3 microRNA

microRNAer (miRNAer) er små ikke-kodende RNAer med en længde på ca. 22 nukleotider, som post-transkriptionel regulere gen ekspressionen ved baseparring med target mRNAer, hvilket medfører kløvning af mRNAet eller repression af translationen.

Nu kendes mere end 6000 miRNAer (miRBase release 11.0), og de er fundet i både planter, dyr og viruser. I menneskets genom er indtil videre fundet 678, og kan muligvis indeholde op til 800-1000 miRNAer (Berezikov et al., 2006).

3.1 Opdagelsen af miRNAer

Den første miRNA lin-4 blev identificeret i 1993, i en undersøgelse af mutationer der forstyrer timingen af den post-embryonale udvikling i nematoden Caenorhabditis elegans (Lee et al., 1993). Det opdagedes, at lin-4 ikke kodede for et protein men for et par RNAer, en kort ca. 22 nt. langt og en lang ca. 61 nt. langt. Den lange kunne foldes til et stem-loop struktur, og man mente, at den var en forstadie til den korte.

Desuden opdagedes det, at lin-4 var antisense komplementær til flere steder på lin-14 genets 3'UTR. Disse komplementære steder fandtes i en region af 3'UTR som tidligere var formodet til at mediere lin-4s repression af lin-14 (Wightman et al., 1993).

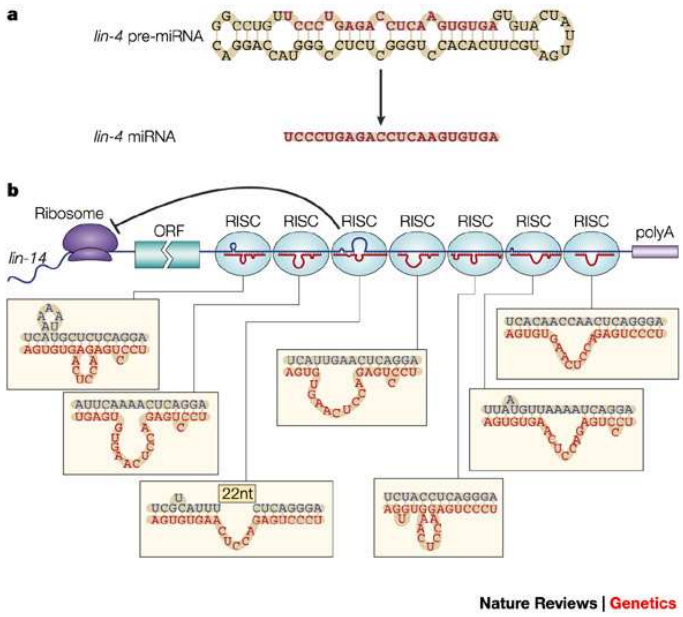
Det blev demonstreret, at disse komplementære steder var vigtige for lin-4s regulering af lin-14, og viste samtidig at denne regulering reducerede mængden af LIN-14 proteinet betydelig uden samtidig at reducere mængden af lin-14 mRNA. Disse opdagelser støttede en model hvor lin-4 RNA parres til lin-14s 3'UTR for specifikt at hæmme translationen af lin-14s mRNA, som en del af den regulering der starter transitionen fra første larvestadie til anden larvestadie hos nematoder (Lee et al., 1993; Wightman et al., 1993), se figur 11

Der var ingen beviser for at lin-4 lignende RNAer fandtes hos andre arter end nematoder, og heller ingen tegn på at der fandtes tilsvarende ncRNAer hos nematoderne (Bartel, 2004). Derfor blev lin-4 betragtet som lidt af et kuriosum.

Først i 2000 fandt man et gen let-7 der også kodede for en ca. 22 nt. langt RNA, og som fungerede på samme måde som lin-4. let-7 regulere transitionen fra sen larvestadie til voksenstadie i C. elegans ved at nedregulere ekspressionen af lin-41 (Reinhart et al., 2000).

Man opdagede desuden at let-7 var konserveret blandt flere forskellige metazoer (de flercellede dyr) (Pasquinelli et al., 2000).

På grund af deres fælles rolle i at kontrollere timingen for larveudviklingen hos nematoder, blev lin-4 og let-7 kaldt small temporal RNAs (stRNAs). Man forventede at yderligere regulerende RNAer af denne type ville blive fundet



Figur 11: Lin-4. a) Pre-miRNA struktur og modne miRNA sekvens for lin-4. b) Sekvens komplementaritet mellem lin-4 (rød) og 3'UTR af lin-14 mRNA (blå). Fra He and Hannon (2004).

(Pasquinelli et al., 2000).

Året efter kom gennembruddet, tre forskellige laboratorier havde ialt fundet over et hundrede gener, der kodede for små ncRNAer fra bananfluer, nematoder og mennesket (Lagos-Quintana et al., 2001; Lau et al., 2001; Lee and Ambros, 2001).

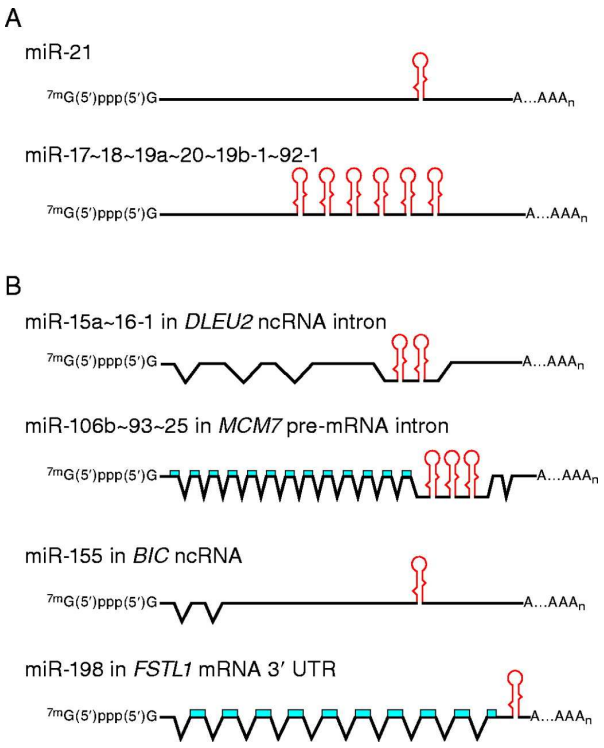
Disse små ncRNA lignede lin-4 og let-7 stRNAer, idet de var ca. 22 nt. endogene RNAer og de var generelt konserverede, nogle ret bredt andre begrænset til nært beslægtede arter.

Men modsat lin-4 og let-7, var mange af de netop identificerede RNAer var ekpressionen ikke begrænset til bestemte udviklingsstadier, men mere begrænset til bestemte celletyper. Derfor blev disse nye små ncRNAer og stRNAerne kaldt microRNAer eller miRNAer (Lagos-Quintana et al., 2001; Lau et al., 2001; Lee and Ambros, 2001).

3.2 miRNA genstruktur

miRNA gener kan klassificeres ud fra deres placering i genomet. Se figur 12

De fleste miRNA gener transkriberes fra regioner der ligger langt fra protein-kodende gener. Nogle af disse miRNA gener ligger også langt fra andre miRNA gener, og det formodes at de udgør deres egen transkriptions enhed, andre er samlet i grupper (cluster) og har samme ekspressions møn-



Figur 12: Pri-miRNA strukturer. Strukturen af menneske pri-miRNAer. Pri-miRNAer kan klassificeres på grundlag af deres placering på genomet i forhold til andre gener. A) To eksempler på selvstændige transkriptions enheder, for en enkelt miRNA og en polycistronisk cluster. B) miRNAer der transkriberes sammen med andre gener. Intron i ikke-kodende RNA, intron i pre-mRNA, exon i ikke-kodende RNA eller exon fra en protein-kodende mRNA. Fra Du and Zamore (2005).

ster, dette indikerer at de transkriberes som en polycistronisk transkript (Lagos-Quintana et al., 2001; Lau et al., 2001; Lee and Ambros, 2001; Reinhart et al., 2002). I stedet for at ligge i en selvstændig transkribtions enhed er omkring halvdelen af de kendte miRNA gerer hos pattedyr placeret inde i protein-kodende geners introns eller inde i ikke kodende RNA geners exons eller introns (Rodriguez et al., 2004). miRNAer i introns har gerne samme orientering og ekspression som den pre-mRNA som de er placeret i, hvilket betyder at de deler en enkelt fælles transkript (Rodriguez et al., 2004). Meget få miRNA gener er placeret i protein-kodende mRNAers utranslaterede regioner. Det formodes at disse transkripter enten danner enten miRNAen eller proteinet men ikke begge fra en enkelt mRNA (Cullen, 2004).

3.3 miRNA biogenese og funktion

Der er forskelle i miRNA biogenesen hos dyr og planter, da projektet er rettet mod dyr, beskrives kun biogenesen for dyr her.

De fleste miRNA gener transkriberes af RNA polymerase II som lange primære miRNAer (pri-miRNAer) med en 5'-methyl7G cap struktur og en poly(A) hale (Lee et al., 2004; Cai et al., 2004). Enkelte bliver dog transkriberet af RNA polymerase III (Borchert et al., 2006).

Dannelsen af den modne miRNA består af to forarbejdningstrin. Den første foregår i kernen, hvor pri-miRNAen bliver forarbejdet til 70 nt. uperfekte stem-loop precusor miRNA (pre-miRNA). Og den anden i cytoplasmaet hvor pre-miRNAen kløves til at danne 21-25 nt. lange modne miRNA. Se figur 13. De to forarbejdningstrin katalyseres hver af en ribonuclease III (RNase III) endonuklease, henholdsvis Drosha og Dicer. De er begge dsRNA-specifikke endonukleaser (Lee et al., 2003; Hutvagner et al., 2001).

Første forarbejdningstrin foregår i cellens nukleous, hvor Drosha og en dobbelt strenget RNA-bindings protein, kendt som DGCR8 i pattedyr, kløver pri-miRNAen nær basen af den dobbelt strengede struktur, og frigiver den ca. 70 nt. lange precursor miRNA (pre-miRNA). Det resulterende pre-miRNA har en 5' fosfat ende og et 2 nt. overhæng i 3'-enden (Lee et al., 2003). Kløvningen definerer den ene ende af den modne miRNA.

pre-miRNAen bliver efterfølgende eksporteret til cytoplasmaet af Exportin-5/RanGTP, som specifikt genkender den karakteristiske ende struktur hos pre-miRNAer (Yi et al., 2003).

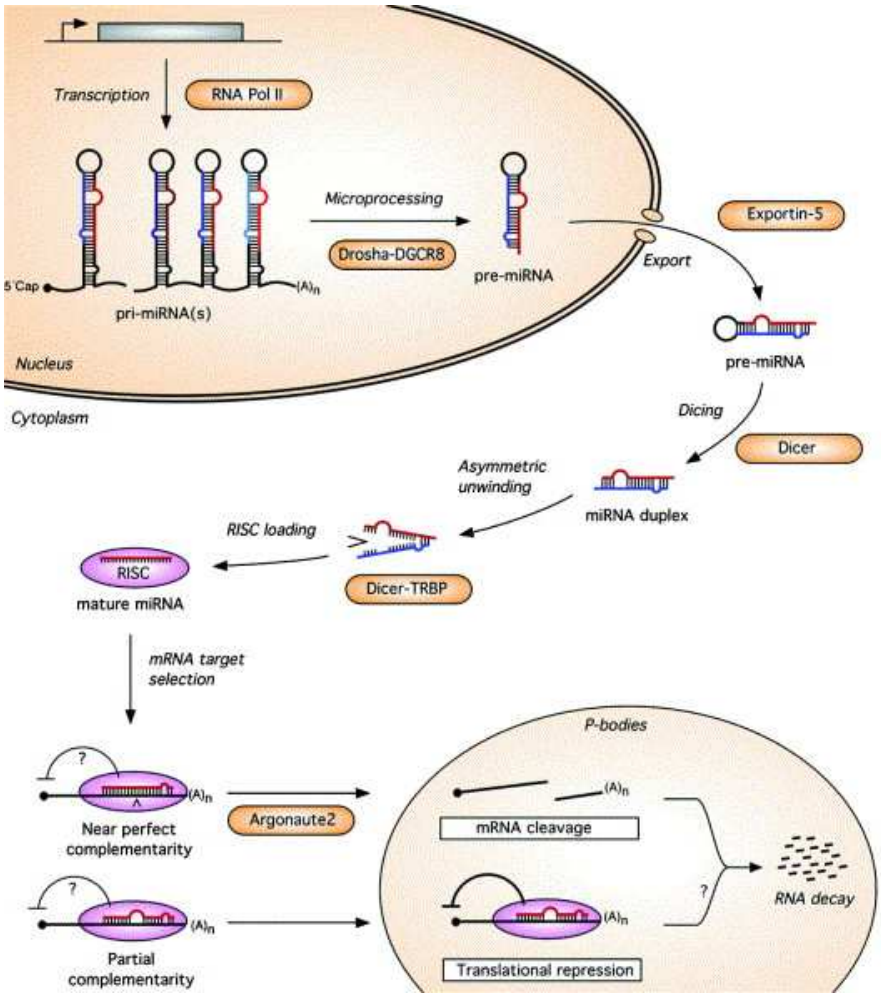
I cytoplasmaet foregår det andet forarbejdningstrin, Dicer foretager sammen med et dsRNA protein trans-activator RNA (tar)-bindings protein (TRBP) i mennesker, kløvningen af pre-miRNAen, hvorved der frigives et 21 nt. RNA duplex med 5' fosfat ende og et 2 nt. overhæng i 3'-enden. Kløvningen definerer den anden ende af den modne miRNA. Den resulterende duplex betegnes normalt miRNA/miRNA* duplexet.

Fra miRNA/miRNA* duplexet bindes den ene streng (miRNA strengen) til et agonaut protein (Liu et al., 2008). Dette kompleks betegnes miRNP (miRNA-contaning ribonucleoprotein complex), mirgonaute eller miRISC (miRNA-containing RNA-induced silencing complex) (Kim, 2005). Den streng i duplexet, der har 5'-enden i den minst termodynamiske stabile ende af duplexet, er den streng, som bliver den modne miRNA (Khvorova et al., 2003).

Det er miRNP komplekset, der foretager repressionen af target gen expressionen. miRNP komplekset sammen med target mRNAet akkumuleres i processing bodies (P-bodies) (Liu et al., 2005)

miRNAer regulerer deres target gener via to hoved mekanismer: Target mRNA kløvning og repression af translationen.

Target mRNA kløvning er den miRNA mekanisme, som oftest findes hos planter. De fleste miRNAer hos planter har en perfekt eller næsten perfekt



Figur 13: miRNA biogenesis. Fra Wienholds and Plasterk (2005).

komplementaritet til deres mRNA tagets (Rhoades et al., 2002). Bindings stedet kan være lokaliseret over hele den transkriberede region af target genet (He and Hannon, 2004). Når miRNAen bindes til mRNA kløves mRNAen, mellem miRNAens nukleotid 10 og 11. Den kløvede mRNA nedbrydes efterfølgende. (Liu et al., 2008).

Hos dyr er der meget få eksempler, hvor miRNAer regulerer deres mRNA targets ved kløvning, derimod foretages repression af translationen hos target mRNAerne. Normalt baseparres miRNAerne til target mRNAernes 3'UTR-region (Liu et al., 2008). Modsat planter er komplementariteten mellem miRNA fra dyr og deres targets normalt begrænset til 5'-regionen af miRNAen, dvs. nukleotid 2-8 eller 2-7 (Lewis et al., 2005). Denne 5'-region er blevet betegnet "seed region" for at vise dens vigtighed for target mRNA

bindingen. Baseparring mellem miRNAens 3'-region og target mRNA er ikke altid nødvendig for repressionen, men stærk baseparring i denne region kan delvist kompensere for svagere seed parring eller øge repressionen (Brennecke et al., 2005). Flere af den samme eller forskellige miRNAer kan bindes til den samme target mRNA 3'UTR-region, og dermed forstærke repressionen (Doench and Sharp, 2004).

3.4 miRNA identifikation

Der er to primære fremgangsmåder til at at opdage nye miRNAer på, som er eksperimentel- og computerbaseret identifikation.

Eftersom nogle miRNAer er udtrykt ved et lavt niveau, og mange kommer til udtryk på forskellig tidspunker og i forskellige celletyper, kan det være svært at finde dem eksperimentelt (Bentwich, 2005).

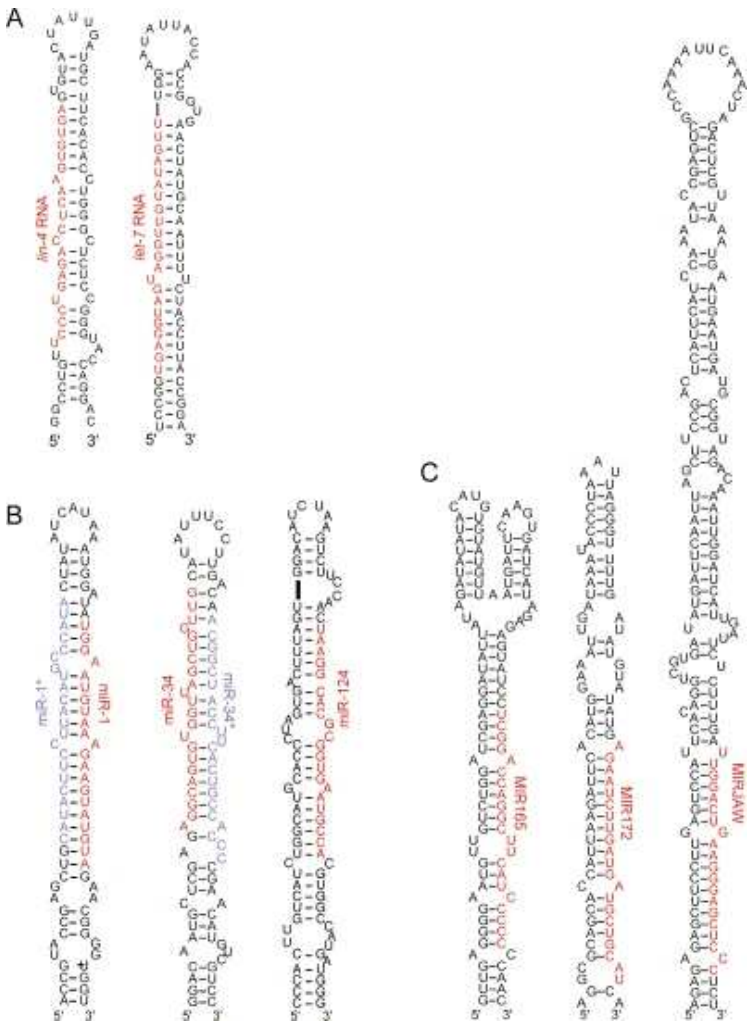
Hvorimod computerbaserede identifikation er uafhængig af niveau, tidspunkt og sted for miRNA aktivitet, hvilket gør fremgangsmåden velegnet til at finde potentielle miRNAer. De er dog kun potentielle miRNAer, de skal efterfølgende verificeres eksperimentelt, men den eksperimentelle søgning efter dem bliver mere målrettet, når man har en ide om, hvor man skal lede.

miRNAer er ofte evolutionært konserverede mellem forskellige arter mange er konserveret fra rundorm til menneske Lagos-Quintana et al. (2001); Weber (2005). Dette kan udnyttes til en simpel computerbaserede identifikations måde, til at finde nye miRNAer, ved at søge efter homologe til allerede kendte pre-miRNAer (Lindow and Gorodkin, 2007). Det er bl.a. gjort af Dezulian et al. (2005); Weber (2005); Zhang et al. (2005). Begrænsningen i metoden er, at der kun findes nye miRNAer, der er homologe med allerede kendte miRNAer.

Pre-miRNAer har en karakteristisk termodynamisk stabil stem-loop struktur med uafbrudte baseparringer og nogle få internal- og bulge loops, se figur 14. I dyr er pre-miRNAerne ca. 60-80 nt. lange , hvorimod i planter kan deres længde variere fra 60 til mere end 400 nt, og virale pre-miRNAer har en længde på 60-119 nt. (Ghosh et al., 2007). Bonnet et al. (2004) har vist at pre-miRNAer har, i modsætning til tRNA og rRNA, en laverer foldnings fri energi end en tilfældig sekvens med samme nukleotid sammensætning. RNA strukturer er yderlige gennemgået i afsnittet "Pre-miRNAer danner stabile stem-loop strukturer".

Søgning efter sådanne stem-loops er ikke nok, til at finde miRNA gener, fordi Bentwich et al. (2005) fandt ca. 11 millioner hairpins i det menneskelige genom, når de ikke behøvede at være konserveret i andre arter. Det er, derfor nødvendigt at inddrage yderligere karakteristika fra kendte miRNAer, for at kunne anvende computerbaseret metoder til identifikation af miRNaer.

Til at afgøre om en stem-loop er en mulig pre-miRNA, er der blevet brugt to forskellige fremgangsmåder, baseret på undersøgelser af strukturer af kendte pre-miRNAer (Lindow and Gorodkin, 2007). Det er regel-baseret



Figur 14: Pre-miRNA stem-loop strukturer. A) lin-4 og let-7 de to første miRNAer. B) Pre-miRNAer fra dyr. C) Pre-miRNAer fra planter. Fra Bartel (2004).

klassifikationer udtænkt af mennesker og maskinlærings metoder.

Regel-baserede klassifikationer ser typisk på om en hairpin struktur har de samme karakteristika som kendte pre-miRNAer. Det kunne være længden af stem regionen, et begrænset antal uparrede nukleotider i regionen, hvor den modne miRNA sidder. Nogle regler inkluderer nukleotid kompositionen uden at betragte strukturen. Det kunne være GC indhold, som skal være indefor bestemte grænseværdier.

MiRScan (Lim et al., 2003b,a) er et eksempel på regel-baseret klassifikation. MiRScan scorer en alignment af to sekvenser fra to forskellige genomer.

Dette gøres på basis af syv egenskaber: 1) Antal base parringer til den formodede modne miRNA, 2) Antal base parringer eksklusiv parringerne til den modne miRNA, 3) Konserveringen i 5'-enden af alignmentet, 4) Tilsvarende konservering i 3'-enden, 5) Bulge symmetri, 6) Afstanden fra den modne miRNA til hairpin loopen, og 7) De specifikke nukleotider på de første fem positioner af formodede modne miRNA. For hver egenskab beregnes en log-odds score, for et sæt af kendte miRNAer og 36.000 "baggrunds hairpins".

Maskinlærings metoderne adskiller sig fra de regel-baserede ved, at reglerne ikke skabes manuelt, men er lært eller trænet fra eksempler ved hjælp af en automatisk procedure. Maskinen lærer at approksimere en funktion, der kortlægger input data til ønsket output. Input er et sæt af egenskaber, der beskriver miRNAen, og output vil være 1 eller 0, hvor 1 indikerer, at det er en miRNA. Maskinen trænes med input eksempler, hvor det ønskede output er kendt. Ideen er så, at maskinen er i stand til at generalisere fra disse eksempler, og efterfølgende kan klassificere kandidater korrekt, hvor outputtet er ukendt. Valg af egenskaber, der skal med i maskinen foregår ikke automatisk, og valget af egenskaber kan have stor indflydelse på maskines ydeevne.

Denne type af maskinlæring betegnes "supervised learning". Normalt kræver den både positive og negative eksempler til træningen. For miRNAer er det nemt at finde positive eksempler, nemlig de kendte miRNAer. Derimod kan det være vanskeligt at finde negative eksempler, for det kan være svært at afgøre om en kandidat ikke er en pre-miRNA.

RNAmicro (Hertel and Stadler, 2006a) anvender en maskinlærings metode – support vektor maskine. Den er anvendt i pipelinen i dette projekt, og er beskevet nærmere i afsnittet "Identifikation af potentielle miRNA gener".

4 Strategien

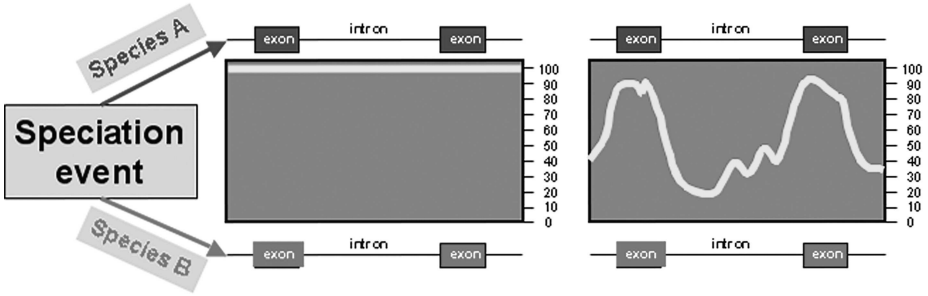
Den overordnede strategien til identifikation af potentielle miRNA gener er i dette projekt, valgt til at være komparativ genomanalyse.

4.1 Komparativ genomanalyse

Komparativ genomanalyse baserer sig på moderne evolutionsteori. Antagelsen der ligger til grund for sammenligning af genom sekvenser er, at de genomer der undersøges har en fælles stamfar, og derfor kan hver basepar i hver af de to organismer forklares ved en kombination af den fælles stamfaderens genom og evolutionen.

Evolutionen kan beskrives kort som en kombination af to processer: Mutationskræfter der skaber tilfældige mutationer i genom sekvensen og selektionstrykket, der eliminerer skadelige mutationer (negativ selektion), har ingen effekt på mutationer (neutral selektion) eller øger hyppigheden af muterede aleller i populationen, som en resultat af øget fitness (positiv selektion) (Ureta-Vidal et al., 2003).

Dette betyder at de sekvenser, der udsættes for negativ selektion i højere grad forbliver konserverede mellem to arter der sammenlignes. Se figur 15.



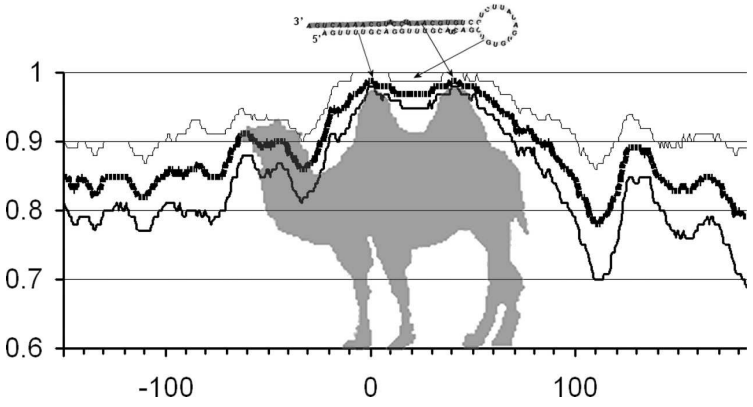
Figur 15: Evolution af funktionelle regioner over tid. Lige efter en artsdannelse, er de to genomer 100% identiske. Med tiden vil de regioner under lidt eller ingen selektionstryk, som f.eks. introns blive mættede med mutationer, hvorimod regioner under negativ selektion, som de fleste exons, vil i højere grad være lig hinanden. Mange sekvenser involveret i regulering af gen ekspression bibeholder også en højere grad af lighed end sekvenser uden funktion. Fra Miller et al. (2004).

De genomsekevnser, der udsættes for negativ selektion, formodes at have en biologisk funktion i organismen. Det er enten protein kodende gener, gener for funktionelle RNAer eller sekvenser med en regulerende funktion.

Det er estimeret at mindst 5% af det menneskelige genom er under negativ selektion og derfor formodentligt funktionel (Chiaromonte et al., 2003).

miRNA-gener er konserveret mellem beslægtede arter (Bartel, 2004), derfor er komparativ genomanalyse en velegnet metode til at finde DNA-sekvenser, som potentielt indeholder miRNA gener.

Berezikov et al. (2005) har beregnet en samlet konserveringsprofil for miRNA gen regioner. 64 miRNA gener er blevet anvendt, hvor 38 har den modne miRNA på den venstre arm af pre-miRNAen, og 26 har den på den højre arm. Konserverings profiler for de individuelle miRNAer er transformeret til relative koordinater med 0 svarende til den første nukleotid i den modne miRNA, hvis det er en venstre-arms miRNA, og den nukleotid, der parres med den sidste base i den modne miRNA hvis det er en højre-arms miRNA. Den gennemsnitlige konserveringsniveau med 95% konfidens intervaller blev beregnet for hver position, se figur 16. Det ses tydeligt at pre-miRNA regionen er højt konserveret, med en lille fald i loop regionen. Desuden ses en markant fald i konservationen umiddelbart uden for pre-miRNA regionen. Dette viser at komparativ genomanalyse er velegnet til at lede efter miRNA gener.



Figur 16: Kamel formet konserverings profil for miRNA gener. Den gennemsnitlige konserveringsniveau (linien i midten) med 95% konfidens intervaller (øverste og nederste linie) blev beregnet for 64 miRNAer. Pre-miRNAs korresponderende regioner er vist med pile. Fra Berezikov and Plasterk (2005).

4.2 Strategien til identifikation af miRNA gener

Strategien til computerbaseret identifikation af potentielle miRNA gener tager udgangspunkt i følgende karakteristika for allerede kendte miRNAer:

- miRNAer er konserverede mellem beslægtede arter
- Pre-miRNAer danner stabile stem-loop strukturer
- Andre sekvens og strukturelle miRNA karakteristika

I det følgende bliver det beskrevet, hvordan disse karakteristika bliver brugt til at vælge strategien til identifikation af potentielle miRNA gener. Det beskrives hvorfor den anvende strategi er valgt, desuden beskrives de bioinformatiske principper, der anvendes i strategien.

Til at implementerer en del af den valgte strategi, anvendes allerede eksisterende programmer, disse beskrives, og det forklares, hvorfor de er blevet valgt, i forhold til alternative løsninger. Implementationen af strategien bliver beskrevet i det følgende hovedafsnit Pipelinen.

4.3 Inputdatasættet

Første trin i strategien er at vælge hvilke DNA-sekvenser, der skal anvendes til at søge efter potentielle miRNA gener. Disse DNA-sekvenser betegnes her inputdatasættet. Valget af inputdatasættet vedrører hvilke arter og hvor mange arter, der skal indgå søgningen, og hvor meget af en art genom, der skal gennemsøges.

Strategien er valgt til at være rettet mod at finde miRNA gener i dyr, det kan derfor ikke forventes, at strategien direkte er anvendelig til at finde miRNA gener i planter eller virus. Det skyldes at pre-miRNAs loop region ofte har en mere varieret størrelse hos planter. miRNAerne er kun sjældent konserveret hos viruser, dette kan dog skyldes, at de undersøgte arter er evolutionært for langt fra hinanden, for der er fundet nærtbeslægtede virusarter med konserverede miRNAer. Ved at lave en strategi, der er rettet mod dyr, gør det muligt at inddrage mennesket som en af arterne i inputdatasættet.

Antallet af arter, der skal indgå i inputdatasættet er sat til tre, for at de fundne sekvenser er konserverede mellem de valgte arter. Det gør det mere sandsynligt at de sekvenser, der bliver alignet er homologe og ikke er tilfældige sekvenser, som kan alignes (Boffelli et al., 2004).

Strategien er rettet mod, at inputdatasættet kan være hele genomer, hele kromosomer eller udvalgte kromosomdele fra de valgte arter. Dette betyder at inputdatasættet kan være stort. Det skal derfor være muligt at maskere allerede kendte gener eller deres exons og repeterende DNA-sekvenser, således at de ikke indgår i inputdatasættet. Derved er det muligt undgå at der bliver brugt tid på bl.a. at analyserer kendte proteinkodende geners exons.

For at gøre strategien fleksibel med hensyn til valg af arter, der kan indgå i identificeringen af potentielle miRNA gener, er strategien valgt til at være uafhængig af eksisterende alignments, som f.eks. kan downloades fra internettet.

Derfor bliver første trin i strategien:

- DNA-sekvenser fra tre beslægtede dyrearter

4.4 miRNAer er konserverede mellem beslægtede arter

Det er muligt enten at søge efter konserverede sekvenser eller stem-loop strukturer som det første. Her er valgt at søge efter konserverede sekvenser først, for det vil give flere alternative muligheder for at søge efter stem-loop strukturer efterfølgende. Det betyder til gengæld, at med denne strategi vil det ikke være muligt at finde ikke-konserverede miRNAer.

Andet trin i strategien tager derfor sit udgangspunkt i, at miRNAer er konserverede mellem beslægtede arter. Med konserverede miRNAer forstås miRNA sekvenser, som findes hos beslægtede arter, og ligner hinanden så meget eller er identiske, at det må formodes, at sekvenserne fandtes hos en fælles stamfader. Og at forskellene mellem de nuværende miRNAer kan forklares med, at der er sket mutationer i sekvenserne og selektion. Sekvenser hos forskellige arter, som stammer fra en fælles stamfader betegnes også homologe sekvenser.

Den simpleste måde at søge efter konserverede miRNAer, er at anvende allerede kendte miRNA sekvenser, til at finde ligende sekvenser i andre arters genomer. Denne metode er benyttet af flere, bl.a. Dezulian et al. (2005); Weber (2005); Zhang et al. (2005). Begrænsningen i metoden er, at der kun findes nye miRNAer, som har sekvenser, der er homologe med allerede kendte miRNAer.

Strategien i dette projekt er valgt til at være mere omfattende, det skal også muligt at finde finde sekvenser med miRNA gener, der ikke allerede kendes fra andre arter. Derfor vil anden trin i strategien ikke værre at finde konserverede miRNAer, men at finde konserverede DNA-sekvenser mellem beslægtede arter. Og først i de efterfølgende trin i strategien afgøre om de fundne sekvenser er potentielle miRNAer.

Derfor bliver anden trin i strategien:

- Identifikation af konserverede DNA-sekvenser

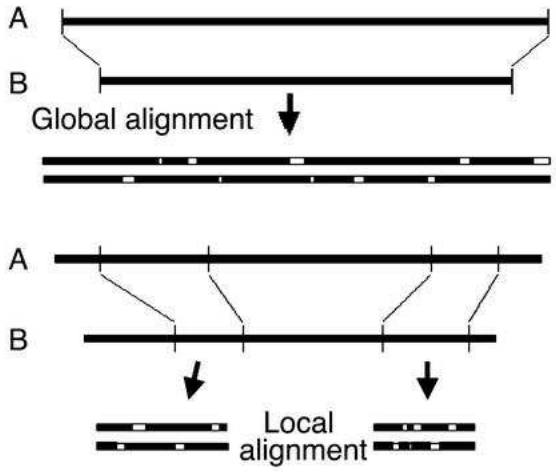
Ved at sammenligne DNA-sekvenser fra beslægtede arter, er det muligt at identificere sekvenser, som er konserveret mellem dem. Til at sammenligne DNA-sekvenser benyttes en metode, der kaldes sekvens alignment. Sammenligning af to sekvenser betegnes parvis alignment, og hvis der er tale om flere sekvenser betegnes det multiple alignment. Her vil blive brugt parvis alignment til at finde de konserverede DNA-sekvenser.

4.4.1 Parvis alignment

Et parvis alignment kan beskrives som en kortlægning af nukleotiderne i en DNA-sekvens over på nukleotiderne i en anden evolutionært beslægtet DNA-sekvens for at finde regioner, som er konserverede (Frazer et al., 2003). Der findes to grundlæggende typer af parvis alignment metoder: global- og lokal alignment.

I global alignment bliver de to sekvenser alignet i hele deres længde, se figur 17. Sekvenser der er ret ens og har omtrent den samme længde er velegnet til global alignment. Er især velegnet hvis homologi er fastslået på forhånd (Mount, 2001).

I lokal alignment bliver kun delsekvenser alignet med hinanden, se figur 17. Lokal alignment er velegnet til at aligne sekvenser, der er ret ens i delsekvenser og forskellige i andre områder, sekvenser der har forskellige længder eller sekvenser med konserverede regioner (Mount, 2001).



Figur 17: Sammenligning af global- og lokal alignment. Fra Pennacchio and Rubin (2003).

Et global alignment beskrives bedst med et eksempel.

To sekvenser som skal alignes:

S1: `TTCATA`
S2: `TGCTCGTA`

For at kunne bestemme den optimale globale alignment af disse to sekvenser, skal der for hver mulig alignment beregnes en score. Den beregnes ved at hver kolonne i alignmentet tildeles en score afhængig af dens indhold. Den totale score er summen af scorerne, som de enkelte kolonner er blevet tildelt. Hvis de to bogstaver i en kolonne er ens er der tale om et match, nukleotiden i de to sekvenser er konserveret. Hvis de to bogstaver er forskellige, er der tale om et mismatch eller substitution, da der er sket en substitution af en nukleotid i en af sekvenserne. Hvis et bogstav er placeret over for et

'-', er der tale om et gap. Et gap betyder at der enten er sket en insertion i den ene sekvens eller der er sket en deletion i den andensekvens. Det er ikke muligt at afgøre hvilken af de to hændelser der er den korrekte.

Den alignment med den højeste score, er den optimale alignment. Der kan godt forekomme flere forskellige alignments med den højeste score. Scoren betegnes også similariteten, og er udtryk for hvor ens de to sekvenser er, og dermed hvor sandsynligt det er, at de stammer fra en fælles stamfader.

Hvis scoren er: Match: $+5$, mismatch: -2 og gap: -6. Giver det følgende optimale alignment:

S1: `T--TCATA`
S2: `TGCTCGTA`

Score: $+5-6-6+5+5$-$2+5+5 = 11$

Når sekvenserne er korte som her, er det forholdsvis hurtigt at beregne den optimale alignment. Men jo længere sekvenserne bliver des flere forskellige alignments er der, som skal scores, for at finde det alignment med den højeste score. Det bliver hurtigt praktisk umulig at score alle mulige alignments, for med to sekvenser med en længde på N vil der være omkring $2^{2N}/\sqrt{2\pi N}$ mulige globale alignments. For to sekvenser med en længde på 300 nukleotider bliver det til 10^{179} forskellige alignments (Eddy, 2004).

Der findes to klasser af sammenligningsalgoritmer til at løse dette problem, så det ikke er nødvendig at beregne scoren for alle mulige alignments, og det er dynamisk programmerings algoritmer og heuristiske algoritmer.

4.4.2 Dynamisk programmerings algoritmer

Dynamisk programmerings algoritmer garanterer, at den fundne alignment, er den optimale med hensyn til den anvendte score. Men det er ingen garanti for, at det den biologiske korrekte alignment.

Dynamisk programmering gør brug af to forskellige algoritmer Needleman-Wunsch (Needleman and Wunsch, 1970) ved globale alignments og Smith-Waterman (Smith and Waterman, 1981) ved lokale alignments.

Både tids- og pladskompleksiteten for dynamisk programmerings algoritmer er $O(NM)$, hvor N og M betegner længden på de to sekvenser, der alignes. Hvis begge sekvenser er omtrent lige lange, så bliver kompleksiteten $O(N^2)$, dvs. at kompleksiteten er kvadratisk.

Selv om at dynamisk programmering er meget hurtige end at beregne alle mulige alignments, så er dynamisk programmering beregningsmæssig meget krævende, det er derfor ikke muligt at foretage alignments på genom niveau med dynamisk programmering Derfor er der udviklet alignments program-

mer, der er gode approksimationer til dynamisk programmerings alignments. Såkaldte heuristiske algoritmer, hvor beregningstiden er subkvadratisk.

4.4.3 Heuristiske algoritmer

For at øge hastigheden benytter heuristiske algoritmer en seeding strategi til at reducere beregningstiden. En seeding strategi er baseret på den ide at en korrekt alignment vil indeholde betydelige strækninger med matchende kolonner uden gaps. Algoritmen kan derved undgå en lang række beregninger, som skulle foretages, hvis dynamisk programmering var benyttet (Ureta-Vidal et al., 2003).

Generelt går alle heuristiske algoritmer igennem tre hoved trin. I første trin findes seeds med den seeding strategi, som algoritmen anvender. I anden trin benyttes seeds som udgangspunkt for udvidelse af alignmentet. I trin tre anvendes dynamisk programmering til at udvide alignmentet yderligere ved også at anvende gaps (Ureta-Vidal et al., 2003). Den mest almindelige seeding strategi er at søge efter et perfekt match mellem ord med en bestemt længde fra de to sekvenser.
Der findes flere forskellige seeding strategier (Batzoglou, 2005):

- At søge efter et perfekt match mellem ord med en bestemt længde fra de to sekvenser
- At benytte kortere ordlængde men samtidig kræve to ord med perfekt match, der ligger indenfor en bestemt afstand af hinanden
- At finde ord af en bestemt længde med næsten perfekt match, hvor et bogstav må være et mismatch
- Et mønstermatch, hvor de machende bogstaver ikke er fortløbende, men følger et mønster, f.eks. 12 af 19 11101001100101011111 hvor 1 betyder match og 0 betyder uden betydning

En væsentlig egenskab ved den heuristiske metode, er afvejningen mellem hastighed og evnen til at finde homologi. Afvejningen styres hovedsagelig af ordlængden på seeds. En kort ordlængde gør det mere sandsynligt at finde en signifikant lokal alignment mellem de to sekvenser, men samtidig øges antallet af tilfældige seed match, som resulterer i tidskrævende alignment udvidelser for at finde de signifikante homologier.

Af de fire nævnte seedig strategier, er de tre sidste teknikker med til at øge evnen til af finde stærke lokale alignments og undgå tilfældige ord match, der ikke fører til alignments, i forhold til den første.

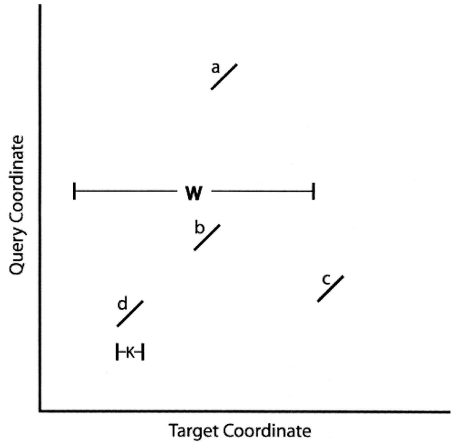
Da inputdatasættet kan være op på genom størrelse, skal det være en heuristisk baseret alignmentprogram, der skal benyttes, for at gøre pipelinen hurtig og dermed praktisk anvendelig.

4.4.4 Valg af heuristisk lokal alignmentprogram

To af de best kendte heuristiske alignment programmer er FASTA (Pearson and Lipman, 1988; Pearson, 2000) og BLAST (Altschul et al., 1990, 1997). Disse metoder er 5-50 gange hurtigere end Smith-Waterman algoritme, og kan ofte producere resultater af samme kvalitet (Pearson, 2001). Men hvis det er meget lange sekvenser der sammenlignes, vil disse programmer køre meget langsomt og bruge meget hukommelse (Ma et al., 2002).

Der findes en lang række af programmer til lokal alignments, i dette projekt er primært gået efter et hurtigt program. Blandt alignment programmerne, er der to som udviklerne selv betegner som værende hurtige: PatternHunter (Ma et al., 2002) og BLAT (Kent, 2002). PatternHunter er meget hurtigere end BLAST (`http://bioinformaticssolutions.com/products/ph/DBbenchmarks.php`), men desværre findes den ikke i enn akademisk gratis version til Linux, derfor er programmet udelukket fra at kunne bruges i dette projekt. Derfor blev BLAT valgt til alignmentprogram.

BLAT er mere nøjagtig og 500 gange hurtigere end eksisternede programmer til mRNA/DNA alignments og 50 gange hurtigere til protein alignments (Kent, 2002). Med BLAT er det muligt at anvende tre forskellige seeding strategier: enkelt perfekt match, enkelt næsten perfekt match og multiple perfekt match. Et eksempel på 2 perfekt match kan ses i figur 18.



Figur 18: Fire match a, b, c og d er alle K bogstaver lange. b og d ligger på samme diagonal, derfor er de en 2 perfekt match. Fra Kent (2002).

Enkelt næsten perfekt match og multiple perfekt match har en væsentlig fordel over enkelt perfekt match. De reducerer drastisk antallet af alignments, der skal tjekkes, for at opnå en bestemt sensitivites niveau. Multiple perfekt match kriteriet kan modificeres til at tillade små insertioner og deletioner, ved at tillade match som er nær hinanden at klumpes sammen, hvis de næste ligger på den samme diagonal. Dette øger sensitiviteten, men også antallet

af alignments.

Algoritmen som BLAT anvender til alignments, er lidt for omfangsrig til at blive beskrevet her, derfor henvises til Kent (2002).

Mulige problematiske forhold ved at anvende BLAT er, at den er udviklet til nukleotid alignments mellem mRNA og DNA fra den samme art, at den ikke er optimeret til alignment af sekvenser fra forskellige arter, og at alignment strategien, som BLAT anvender for nukleotid alignments bliver mindre effektive når sekvensidentiteten kommer under 90% identitet (Kent, 2002). Alligevel er BLAT valgt, for der findes andre alignmentprogrammer BLASTz (Schwartz et al., 2003) og BLATz (`http://hgwdev.cse.ucsc.edu/~kent/exe/linux/blatz.zip/`), der kan anvendes som alignmentprogram uden problemer, hvis BLAT ikke er sensitiv nok, i forhold til de arter, der skal alignes. Det vil dog betyde en væsentlig øgning af udførselstiden. BLASTZ og BLATZ kan anvendes i stedet for BLAT, fordi axt formatet anvendes som outputformat fra BLAT i dette projekt. BLATZ har også axt som en mulig outputformat, og BLASTZs output kan konverteres til axt formatet med programmet lavToAxt (Kent et al., 2003)

BLAT er også blevet brugt af andre med succes til alignment mellem arter, herunder i forbindelse med miRNA identifikation bl.a. Lim et al. (2003a); Seitz et al. (2004); Weber (2005).

4.4.5 BLAT alignments

BLAT kan kun foretage alignment af to sekvenser af gangen. Derfor skal alignmentet af de tre arters sekvenser foretages i to omgange.

Først foretages en alignment af to af arternes DNA-sekvenser, for at finde konserverede delsekvenser mellem disse. Dette er den første alignment.

Efterfølgende foretages en alignment mellem de fundne konserverede delsekvenser fra en af arterne i den første alignment, og den art der ikke indgik i den første alignment, for at finde konserverede delsekvenser mellem disse. Dette er den anden alignment.

Til slut skal resultaterne fra de to alignments samles. Så slutresultatet bliver de delsekvenser, der er konserverede mellem alle tre arter. De skal efterfølgende anvendes i næste hovedtrin i strategien, der bliver beskrevet i det følgende.

4.5 Pre-miRNAer danner stabile stem-loop strukturer

Trejde trin i strategien tager udgangspunkt i at pre-miRNAer danner stabile stem-loop strukturer.

Først skal det forklares, hvad der menes med stabile stem-loop strukturer. Dernæst beskrives fremgangsmåder til at bestemme om en given RNA-molekyle kan danne en stabil stem-loop struktur.

4.5.1 RNA strukturer

pre-miRNA-molekyler er RNA-molekyler, som er enkeltstrengede molekyler, der foldes til specifikke tre-dimensionale strukturer pga. intramolekylære bindinger. RNA-molekyler kan derfor beskrives på fire forskellige strukturniveauer (Dwyer, 2003):

Primære struktur: Refererer til den lineære rækkefølge af nukleotider, som et RNA molekyle består af.

Sekundære struktur: Refererer til den baseparring ved hjælp af hydrogenbindinger, der dannes mellem molekylets komplementære baser, som ikke ligger ved siden af hinanden i den primære struktur. A-U og G-C par kaldes Watson-Crick par og G-U par kaldes wobble par. Disse baseparringer kaldes samlet kanoniske par, da de er de mest almindelige og mest stabile (Mathews et al., 2006).

Tertiære struktur: Refererer til de præcise position af hver atom i et RNA-molekyle i det tre dimensionelle rum.

Kvaternære struktur: Refererer til den relative placering i det tre dimensionelle rum af RNA-molekyler og proteiner, der går sammen og danner mere komplekse strukturer.

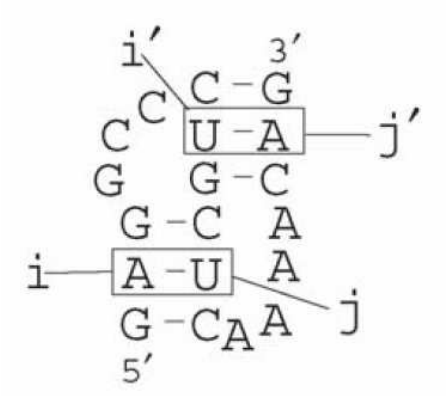
Det er den sekundære struktur, der tales om, når det beskrives, at premiRNAer danner stabile stem-loop strukturer.

4.5.2 RNA sekundære struktur

For at en sekundær struktur kan være gyldig, er baseparrene underlagt flere begrænsninger (Higgs, 2000). Lad nukleotiderne i et RNA-molekyle være nummereret 1 til N og i og j være to nukleotid positioner, hvor $5' - 1 <= i < j <= N-3'$, så kan der dannes et basepar mellem dem $i.j$, hvis baserne er komplementære og $|j-i| \geqq 4$, da der normalt skal være mindst tre uparrede nukleotider mellem et basepar. Lad i' og j' være et andet tilladt basepar $i'.j'$. Så vil disse basepar enten være adskilte $i < j < i' < j'$, indlejrede $i < i' < j' < j$ eller danne en pseudeknot $i < i' < j < j'$, se figur 19. Nogle opfatter pseudoknots som tilhørende den tertiære struktur.

Når et RNA-molekyles nukleotider danner basepar, opstår der forskellige karakteristiske delstrukturer i den sekundære struktur, således at hver nukleotid tilhører en bestemt delstruktur.

Et RNA molekyle deles op i følgende delstrukturer (Mathews et al., 2006; Zuker and Sankoff, 1984):



Figur 19: RNA pseudeknot. Fra Mathews and Turner (2006).

Stacked pair: Findes der to basepar $i.j$ og $i'.j'$ er det et stacked pair, hvis $i' = i + 1$ og $j' = j - 1$, dvs. at de to basepar ligger ved siden af hinanden. Hvis flere basepar er placeret ved siden af hinanden i strukturen, betegnes det et stem eller et helix.

Hairpin loop: Hvis alle nukleotiderne mellem et basepar $i.j$ er uparrede, er der tale om et hairpin loop.

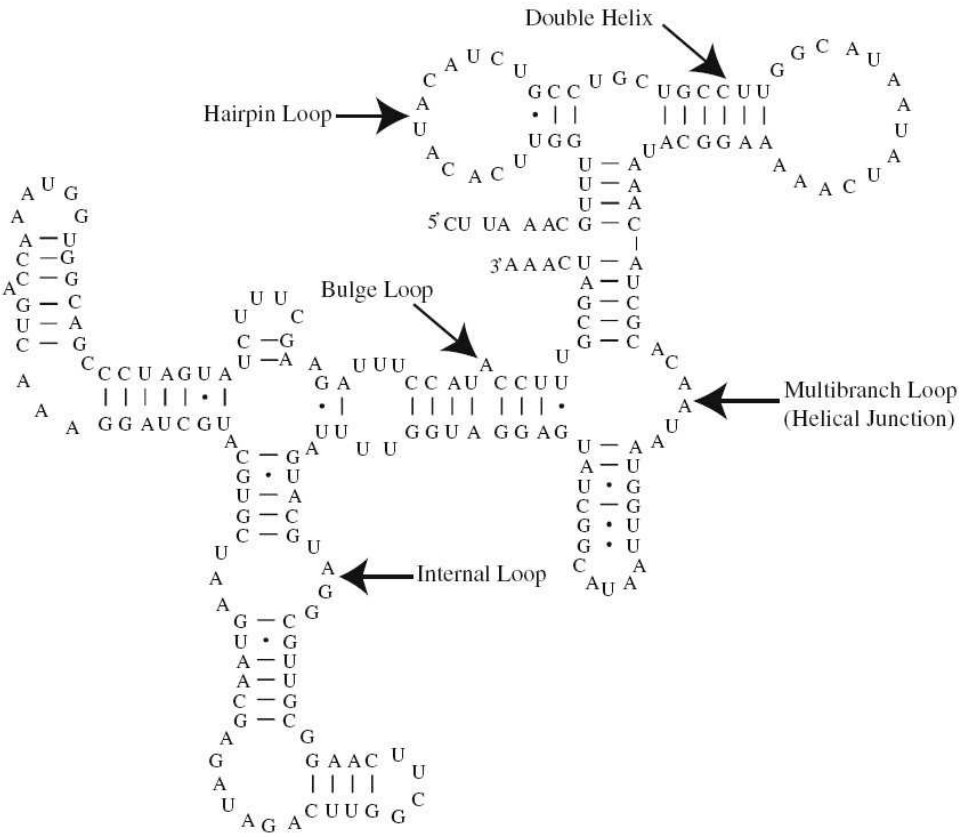
Internal loop: Findes der to basepar $i.j$ og $i'.j'$ hvor $i+1 < i' < j' < j-1$ og nukleotiderne mellem i og i' og mellem j og j' er uparrede, er der tale om et internal loop.

Bulge loop: En speciel slags internal loop, hvor enten $i' = i + 1$ eller $j' = j - 1$, betegnes en bulge eller bulge loop. Dvs. at der kun er uparrede nukleotider i den ene side mellem de to basepar $i.j$ og $i'.j$'.

Multibranced loop: Det er en loop, hvor tre eller flere stem's udgår fra.

Externals: De uparrde nukleotider, der ikke indgår i et loop betegnes e-xternals.

De forskellige delstrukturer er vist i figur 20. Eksempler på pre-miRNAer er vist i figur 14. Som det kan ses af figurerne, så består en pre-miRNA-molekyle af et hairpin loop og en stem region med et eller flere stems afbrudt af internal loops og bulge loops.



Figur 20: RNA sekundære delstrukturer. Fra Mathews (2006a).

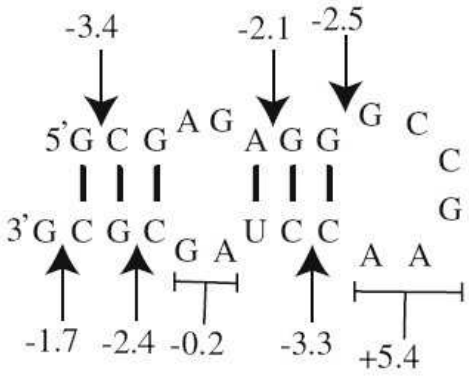
4.5.3 RNA stabilitet

Et RNA-molekyles stabilitet er hovedsageligt bestemt af baseparringernes stabiliserende effekt og den destabiliserende effekt af loops. Nogle loops har dog en stabiliserende effekt. Når det beskrives at pre-miRNA-molekyler skal være stabile, så er det den termodynamiske stabilitet, der tales om. Den termodynamiske mest stabile molekyle er den med minimal fri energi (MFE) (Higgs, 2000).

Ligevægten mellem en RNA-molekyle med struktur S_1 og en ustruktureret tilstand U, $U \rightleftharpoons S_1$, er bestemt af ligevægtskonstanten $K_1 = [S_1]/[U]$ givet ved $K_1 = e^{-\triangle G_1^\circ/RT}$, hvor $\triangle G_1^\circ$ er ændringen i fri energi for struktur S_1 og dermed et udtryk strukturens stabilitet, R er gaskonstanten og T er den absolutte temperatur. Tilsvarende for to strukturer S_1 og S_2 er forskellen i stabilitet givet ved $K_1/K_2 = [S_1]/[S_2] = e^{(\triangle G_2^\circ - \triangle G_1^\circ)/RT}$. Dette betyder at den struktur, der har den minimale fri energi, er den mest repræsenterede

struktur ved ligevægt (Mathews, 2006b).

Bestemmelsen af den fri energi for en RNA-molekyle foregår ved at beregne forskellen i den fri energi $\triangle G^\circ$ mellem den ufoldede RNA-sekvens og en bestemt sekundær struktur, og beregnes oftest efter nearest-neighbor modellen (Borer et al., 1974). Hovedantagelsen i denne model er, at stabiliteten af hver basepar eller anden struktur i et RNA-molekyle, er kun afhængig af identiteten de tilstødende basepar (Higgs, 2000). Og den samlede ændring i den fri energi er summen af de individuelle bidrag. De termodynamiske parametre, der indgår i nearest-neighbor modellen, er baseret på eksperimentelle analyser og estimater af den fri energi hos oligonukleotider (Xia et al., 1998; Mathews et al., 1999, 2004). Et eksempel på Nearest-neighbor beregning for en stem-loop er vist i figur 21.



$$\Delta G^\circ = -1.7 - 3.4 - 2.4 - 0.2 - 2.1 - 3.3 - 2.5 + 5.4$$
$$= -10.2 \text{ kcal/mol}$$

Figur 21: Eksempel på beregning af $\triangle G^\circ$ efter nearest-neighbor modellen. Bidraget for hvert motiv og den samlede stabilitets ændring er vist. Fra Mathews (2006a).

4.5.4 Bestemmelse af RNA-molekylers sekundære struktur

RNA foldningen sker hierarkisk (Tinoco and Bustamante, 1999). Dannelsen af den sekundære struktur sker på en hurtigere tidsskala end den tertiære struktur (Woodson, 2000). Og den sekundære strukturers bindinger er generelt stærkere end tertiære strukturers bindinger Styrken i de tertiære bindinger er generelt for svage til at bryde den allerede dannede sekundære struktur (Higgs, 2000). Derfor kan RNA-molekylers sekundære struktur ge-

nerelt forudsiges uden kendskab til den tertiære struktur (Mathews, 2006b).

Det er blevet beregnet at antallet af mulige sekundære strukturer en RNA-molekyle kan have er ca. $1,8^N$, hvor N er antallet af nukleotider i molekylet (Zuker and Sankoff, 1984). En RNA-molekyle indeholdende 100 nukleotider, vil give $3,4 * 10^{25}$ forskellige strukturer. Hvis en computer kan beregne den fri energi for 10.000 strukturer i sekundet, vil det tage $3,4 * 10^{21}$ sekunder eller $1,1 * 10^{14}$ år at beregne den fri energi for alle de mulige strukturer (Mathews, 2006a).

En løsning på dette problem er at anvende dynamisk programmering (Nussinov and Jacobson, 1980; Zuker and Stiegler, 1981). Først beregnes den minimale fri energi for en RNA-sekvens ved hjælp af en rekursions algoritme, der starter med at beregne den minimale fri energi for de mindste delsekvenser, som kan danne en struktur. Efterfølgende beregnes den minimale fri energi for længere og længere delsekvenser, ved at inddrage beregningerne for de kortere delsekvenser, hvilket speeder beregningsprocessen op. Til slut kan den minimale fri energi for hele sekvensen beregnes.

En struktur med den minimale fri energi kan derefter bestemmes ved backtracking gennem de minimale fri energier, der blev beregnet for delsekvenserne.

Dynamisk programmerings algoritmer garanterer at strukturen med den minimale fri energi findes, givet reglerne for energi modellen. De hurtigste algoritmer har en beregningstid på $O(N^3)$ og et pladsforbrug på $O(N^2)$, når pseudeknots er ekskluderet fra beregningerne (Mathews, 2006a).

Zuker-Stiegler algoritmen (Zuker and Stiegler, 1981) er implementeret i de to progammer mfold (Zuker et al., 1999; Zuker, 2003) og RNAfold (Hofacker et al., 1994; Hofacker, 2003) til at beregne den minimale fri energi struktur ved hjælp af nearest neighbor modellen. De anvender de samme termodynamiske parametre (Mathews et al., 1999). Der kan dog være mindre forskelle i resultaterne fra programmerne, men det skyldes små forskelle i implementationerne. Der er tilsyneladende ingen signifikante forskelle i algoritmenres nøjagtighed (Gardner and Giegerich, 2004).

mfold og RNAfold er de alt dominerende programmer, der anvendes til at folde en RNA-sekvens, for at undersøge om det er en potentiel pre-miRNA. Dette skyldes formodentligt at Ambros et al. (2003) direkte nævner, at det skal være mfold eller en anden konventionel RNA-foldnings program, der skal anvendes i forbindelse med at afgøre om en RNA-sekvens kan, foldes så den danner en korrekt pre-miRNA struktur.

4.5.5 RNA-foldnings programmernes nøjagtighed

Den sekundære struktur som findes med et RNA-foldnings program er en forudsigelse om, hvordan en given RNA-sekvens vil foldes i virkeligheden. Nøjagtigheden af en RNA-foldningprogram er kan testes ved at forudsige strukturer for RNA sekvenser, hvor strukturen er kendt.

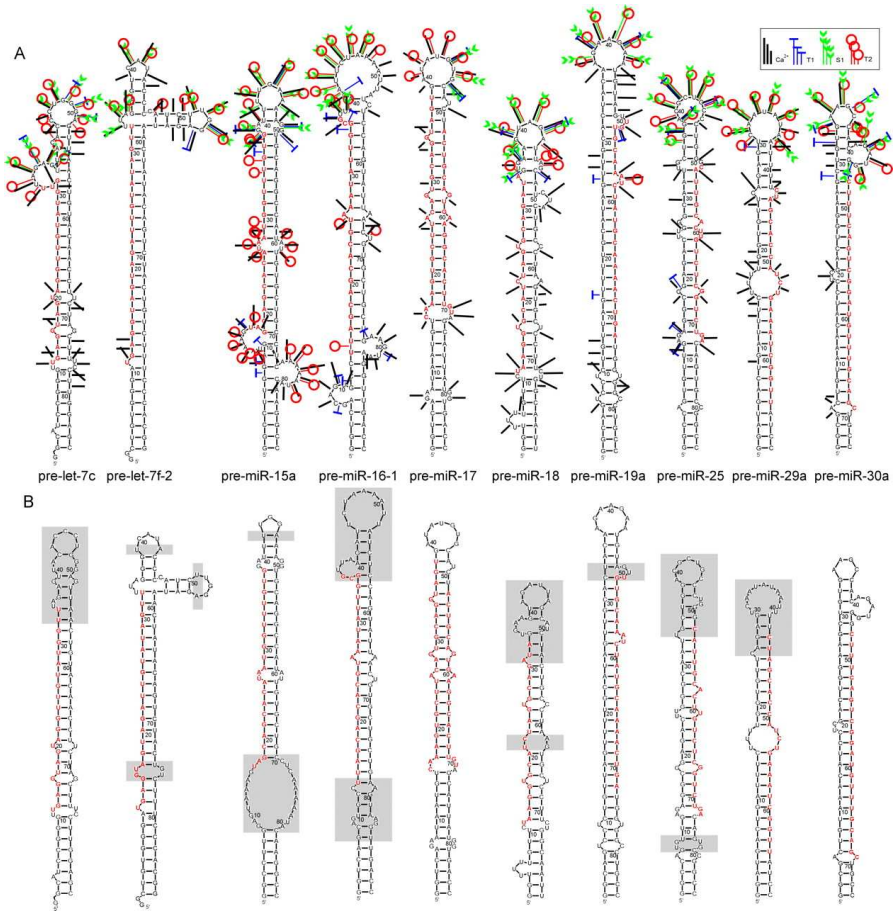
Ved at anvende en række forskellige RNA typer blev det bestemt at 73% af baseparene i de kendte strukturer kunne bestemmes korrekt med mfold (Mathews et al., 1999), hvis baseparringen var en base ved siden af den korrekte, så blev den regnet for korrekt i undersøgelsen. I en anden undersøgelse hvor kun helt korrekte baseparringer blev tilladt, blev nøjagtigheden, sensitiviteten, bestemt til at være 56% (Dowell and Eddy, 2004). I en tredje undersøgelse blev sensitiviteten bestemt til at ligge i området 22-63% i fire forskellige datasæt (Gardner and Giegerich, 2004). Der er flere grunde til, at RNA-foldningprogrammerne ikke er mere nøjagtige. De termodynamiske regler er ukomplette, algoritmerne bruger approksimationer, nogle strukturer delvist er bestemt af selve foldningsprocessen, og visse RNA-sekvenser bliver foldet til mere end en struktur (Mathews and Turner, 2006).

Der er dog ingen af disse undersøgelser der inddrager pre-miRNA-sekvenser i deres undersøgelser. Der findes en enkelt undersøgelse, hvor 10 pre-miRNAers strukturer er blevet bestemt eksperimentelt og vha. mfold og sammenlignet (Krol et al., 2004). I figur 22 er de to sæt strukturer vist. Forklaringerne til symbolerne i figur 22 A henvises til artiklen Krol et al. (2004).

Det mest intressante er figur 22 B, hvor forskellene mellem de de to sæt strukturer er angivet med gråt. Det kan ses at kun to pre-miRNAer er ens i de to sæt. Syv stededer er der små forskelle, det drejer sig om, at der mangler en eller to baseparringer eller der er en for meget. Der er i øvrigt en mere lille forskel, ca. midt på pre-miR-25, der ikke er markeret med gråt på figur 22 B. Der er syv større forskelle, hvor der er flere ændringer i paseparringerne. De fem vedrører hairpin loop-regionen og to vedrører to internal loops/bulge loops. Der er ikke foretaget nogen beregning på størrelsen af forskellene. Det væsentligste der kan siges ud fra figurerne er, at den karakteristiske stem-loop struktur for pre-miRNAer er bevaret på trods af forskellene. Det indikere at mfold er velegnet til at bestemme om RNA-sekvens kan foldes, til at give den karakteristiske pre-miRNA stem-loop struktur. Figur 22 viser også, at der kan forekomme pre-miRNAer med en forgrening og to loop regioner, som pre-let-7f-2.

4.5.6 Valg af RNA foldnings program

Gardner and Giegerich (2004) har sammenlignet mfold og RNAfold med andre metoder til at bestemme RNA-sekvensers sekundære struktur. Disse metoder betegner de "automatiske komparative RNA sekvens analyse", og de forudsætter homologe RNA-sekvenser. Metoderne deles ind i tre fremgangsmåder. Plan A) Først alignes sekvenserne med et standart multiple alignment program, derefter anvendes strukturneutrale mutationer til at bestemme en consensus struktur. Plan B) Samtidig alignment, foldning og bestemelse af en protosekvens for et sæt af homologe strukturelle RNA sekvenser baseret på Sankoff-Algoritmen (Sankoff, 1985). Denne algoritme er meget computer ressource krævende $O(n^3m)$ i tid og $O(n^2m)$ i plads, hvor n er sekvens læng-



Figur 22: Strukturer for 10 pre-miRNAer. A. Eksperimentel bestemte strukturer. B. Laveste fri energi struktur bestem med mfold. De steder hvor strukturerne er forskellige er markret med gråt. Fra Krol et al. (2004).

den og m er antal sekvenser. Plan C) Hvor ingen sekvens konservation kan observeres, bestemmes den sekundære struktur for hver sekvens og efterfølgende foretages en alignment af strukturerne.

Resultatet af Gardner and Giegerich (2004) undersøgelser var at to programmer fra plan A generelt præsterer godt og bedre end mfold og RNAfold. Programmerne er, Pfold (Knudsen and Hein, 1999, 2003) der er baseret på stokastisk kontekst fri grammatik, og RNAalifold (Hofacker et al., 2002) der er baseret på minimum fri energi og sekvens covariation.

Da strategiens anden trin består i at finde konserverede homologe sekvenser, ville det være oplagt at anvende enten Pfold eller RNAalifold til at finde pre-miRNA stem-loop strukturer i disse sekvenser. Det blev dog ingen af disse to programmer der blev valgt til at indgå i pipelinen, men i stedet

RNAz (Washietl et al., 2005), hvori indgår både RNAfold og RNAalifold. RNAz er en RNA klassifikationsprogram. Fordelen ved at anvende RNAz, at det ofte er nødvendigt at udelukke pre-miRNAer med forgreninger, pre-let-7f-2 som i figur 22, hvis der benyttes en decideret RNA-foldningsprogram i en miRNA søgestrategi.

4.5.7 Sekvens klassifikation

RNAz bruger en supprot vektor maskine (SVM) lærings algoritme, som er trænet på et stort test sæt af kendte ncRNAer, til at klassificere alignments som "strukturel RNA" eller "andet". To karakteristisk egenskab ved strukturelle RNAer z-score og struktur konserverings index (SCI) bruges til af foretage klassifikationen (Washietl et al., 2005; Washietl, 2006).

RNAz beregner et normaliseret mål for den termodynamiske stabilitet ved at sammenligne minimal fri energi (MFE) m for en given sekvens med MFEer af et stort antal tilfældige sekvenser af samme længde og nukleotid komposition. z-scoren beregnes som $z = (m-\mu)/\sigma$, hvor μ er middelværdien og σ standard afvigelsen for MFEerne for de tilfældige sekvenser. Negative z-scorer indikerer, at en sekvens er mere stabil end forventet for en tilfældig sekvens med samme længde og nukleotid komposition. RNAz beregner ikke z-scoren ved at sample tilfældige sekevneser, men benytter en SVM regressions model, uden væsentlig tab af nøjagtighed, men sparer ca. en faktor 1000 i computertid.

RNAz bruger RNAalifolds fremgangsmåde til at finde konsensus sekundær strukturen for en alignment. RNAalifold virker næsten på samme måde som enkelt sekvens foldnings algoritme, den vigtigste forskel er, at energi modellen er suppleret med covariation information. Kompenserende mutationer (f.eks. CG muterer til UA) konsistente mutationer (f.eks. AU muterer til GU) giver en bonus, mens mens inkonsistente mutationer (f.eks. CG muterer til CA) giver en straf. Dette resulterer i en konsensus MFE E_A. RNAz sammenligner denne konsensus MFE med den gennemsnitlige MFE for de individuelle sekvenser $\bar{E}$. Struktur konserverings indekset kan nu beregnes: SCI $= E_A/\bar{E}$. SCI tæt på nul indikerer, at RNAalifold ikke finder en konsensus struktur, hvorimod et sæt perfekt konserverede strukturer har SCI ≈ 1. En SCI 1 indikerer en perfekt konserveret struktur, som er støttet af kompenserende og/eller konsistente mutationer, som bidrager til E_A.

Sammenlignet med andre RNA klassifikationsprogrammer, så er RNAz er mere nøjagtig end QRNA (Rivas and Eddy, 2001), MSARI (Coventry et al., 2004) og ddbRNA (di Bernardo et al., 2003). RNAz's tidskompleksiteten er $O(N \times n^3)$, og er betydeligt hurtigere end QRNA og ddbRNA, hvorimod MSARI formodes at være lige så hurtig som RNAz (Washietl et al., 2005).

Da RNAz er blevet valgt, til at indgå i pipelinen, skal trejde trin i strategien ændret til:

- Identifikation af ncRNA sekvenser

4.5.8 Multiple alignment

For at kunne anvende RNAz til at identificere ncRNA kodende sekvenser i de homologe konserverede DNA sekvenser, der er blevet fundet i andet trin i strategien, skal disse DNA sekvenser alignes. Det gøres med et global multiple alignment program, da der er tre DNA sekvenser, som skal alignes.

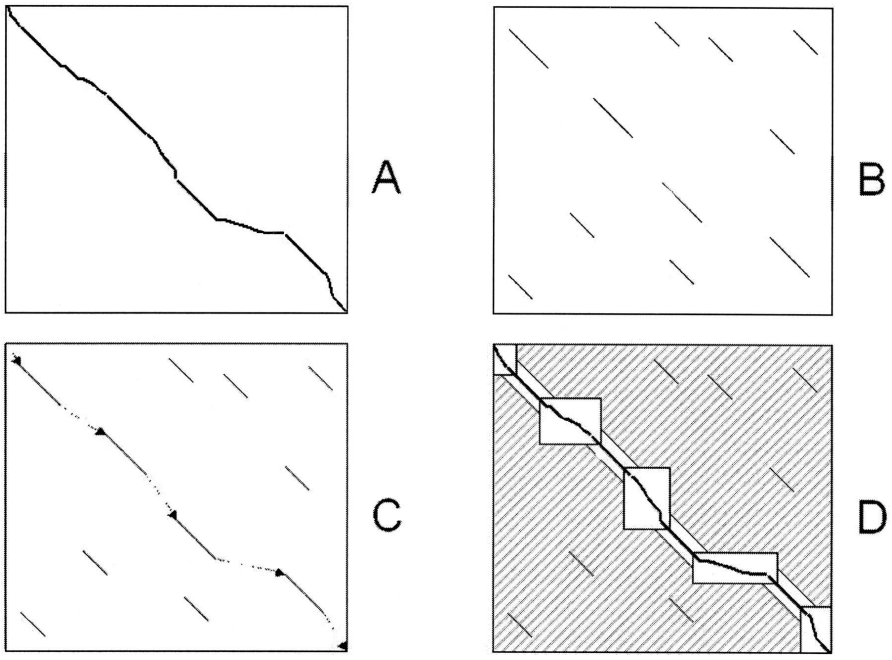
Muliple alignments er meget beregningstunge. Tre sekvenser med en længde på 5 nukleotider har > 14 milliarder forskellige alignments, og hvis der er tale om fem sekvenser er der $1,05 * 10^{18}$ forskellige alignments (Slowinski, 1998). Derfor bruges der heuristiske metoder, den mest almindelige er progressiv alignment, som udnytter den evolutionære struktur af data, nært beslægtede arter alignes først, hvorefter de bliver alignet til fjernere beslægtede arter. En måde til yderligere at speede processen op er brug af ankre, som er regioner med ret sikre alignments, som gør den totale søgerum for hele alignmentet væsentligt mindre (Margulies and Birney, 2008).

Multiple alignmets programmer kan deles ind i to grupper, de der har deres oprindelse i alignment af proteiner, som er korte, og de der er rettet mod alignments af lange DNA-sekvenser. Først nævnte type programmer får problemer med at køre, hvis sekvenserne bliver for lange. Derfor blev der valgt et multiple alignments program, der kan håndtere lange sekvenser. Så pipelinen bliver uafhængig af sekvenslængden i de to BLAT alignments.

Valget stod mellem to forskellige programmer MAVID (Bray and Pachter, 2004) og Multi-LAGAN (MLAGAN) (Brudno et al., 2003). MLAGAN blev valgt, fordi MAVID hurtigt blev valgt fra, det skyldes dog en misforståelse. Bray and Pachter (2004) gør meget ud af at bruge protein kondende geners exon som ankre i alinmentprocessen. Hvilket ikke er særligt anvendelig her, da en del af strategien i pipelinen, er at maskere alle proteinkodende geners exons. Ved en senere test af programmet viste det sig at den type ankre ikke er et krav. Men da var pipelinen implementeret med MLAGAN.

MLAGAN er baseret på progressiv alignment. En multiple alignment af K sekvenser konstrueres i $K-1$ parvise alignment trin, hvor i hver trin to sekvenser eller mellem multiple alignments alignes. MLAGAN bruger LAGAN som parvis-alignment subrutine. LAGAN foretager en global alignment af to sekvenser i tre hovedtrin. 1) Generering af lokale alignments mellem de to sekvenser. 2) Konstruktion af en grov global kortlægning, ved at sammenkæde den maksimalt scorende ordnet delmænge af de lokale alignments – ankrene. 3) Beregne den endelige globale alignment, ved at finde den bedste alignment, der forbliver inden for en begrænset område af den grove global kortlægning. Se figur 23.

MLAGAN aligner K sekvenser i tre hovedtrin. 1) Generering af grove globale kortlægning mellem hver sekvenspar. Dette gøres ved hjælp af LAGANs trin 1 og 2. 2) Foretag en global alignment mellem de to nærmeste sekvenser ifølge den givne fylogenetiske træ ved hjælp af LAGAN. Bestem den grove global kortlægning for denne multi-sekvens til alle andre multi-sekvenser.



Figur 23: LAGAN algoritmen. A) En global alignment mellem to sekvenser består af en sti fra den øverste venstre hjørne til den nederste højre hjørne. B) LAGAN finder først alle lokale alignemts mellem de to sekvenser. C) LAGAN beregner en maksimalt scorende ordnet delmængde af de lokale alignments – ankrene – og danner den grove global kortlægning. D) LAGAN begrænser beregnerne af den optimale Needleman-Wunsch alignment til boksene og omkring ankrene. Fra Brudno et al. (2003).

Dette gentages indtil alle sekvenser er indraget i en multiple aligment. 3) Iterativ forbedring af alignemtet. For hver sekvens i den multiple alignment, findes segmenter, der aligner bedre end en grænseværdi, disse anvendes som ankre for endnu en alignemnt af sekvensen til de andre sekvenser, her anvendes også LAGAN. Trin tre er valgfrit.

4.6 Identifikation af potentielle miRNA gener

Da tredje trin i strategien kun identificerede ncRNA sekvenser i konserverede DNA-sekvenser, er der endnu ikke vurderet om de fundne sekvenser har nogle miRNA karakteristika end at de er konserverede mellem beslægtede arter og indeholder mulige ncRNA strukturer.

Fjerde trin i strategien bliver derfor:

- Identifikation af potentielle miRNA gener

Valget stod mellem selv at udvikle et program eller at finde et eksisterende program, til at søge de fundne sekvenser igennem, for at finde potentielle miRNA sekvenser.

Det første alternativ ville sikkert ikke give et bedre resultat end hvis et eksisterende program afventes. Der var to programmer, som var særligt interessante Triplet-SVM-classifier (Xue et al., 2005) og RNAmicro (Hertel and Stadler, 2006a,b), der begge har en sensitivitet på ca. 90%, og de bruger bruger en support vektor maskine til at klassificere pre-miRNAer med.

Triplet-SVM-classifier klassificere en sekvens af gangen, der klassificeres som pre-miRNA (1) eller ikke pre-miRNA (-1). Hvorimod RNAmicro udover at klassificere en multiple alignment, som pre-miRNA (1) eller ikke pre-miRNA (-1), så giver den også en score, for de alignments, der klassificeres som miRNAer, hvilket gør det muligt at vælge en anden klassifikationsgrænse, blot den er højere end 0,50. Desuden kan RNAmicro klassificere længere alignments ved at anvende et glidende vindue med størrelser fra 70 nt. til 130 nt. Desuden er RNAmicro udviklet til at analysere multiple alignments, der er blevet klassificeret som ncRNA af RNAz. De ting der er nævnt her gjorde, at RNAmicro blev valgt til at identificere potentielle miRNA gener.

RNAmicro består af 1) en preprocessor der identificere konserverede "næsten-hairpins", 2) et modul der beregner en vektor af numeriske deskriptorer for hver "næsten-hairpins", og en support vektor maskine der klassificere kandidaten baseret på dens vektor af deskriptorer. RNAmicro bruger RNAalifold til at bestemme den sekundære struktur af alignmentet med.

RNAmicro version 1.1 bruger tolv forskellige egenskaber, der bruges som deskriptions vektorer til support vektor maskinen, og det er:

$\bar{z}$	Gennemsnitlig z-score
SCI	Struktur konserverings indeks
GPI	Gennemsnitlige parvise identitet
AMEF	Justeret minimal fri energi AMEF=(MFE/sekvenslængde)×100
MFEI	Minimal fri energi indeks MFEI=AMEF/(G+C)%.
S_5	Gennemsnitlige kolonnevis entropi for 5'-stem
S_0	Gennemsnitlige kolonnevis entropi for hairpin loop
S_3	Gennemsnitlige kolonnevis entropi for 3'-stem
S_min	Sekvensvinduet på 23 nt. med laveste entropi
l_s	Stem længde
l_h	Hairpin loop længde
G+C	GC indhold

Deskriptions vektorerne skaleres lineært til intervallet $[-1, +1]$. Entropien for en region ξ beregnes på følgende måde:

$$S_\xi = -\frac{1}{len(\xi)} \sum_{i \in \xi} \sum_{\alpha = A,C,G,C} p_{i,\alpha} ln p_{i,\alpha}$$

Hvor $p_{i,\alpha}$ er den brøkdel af α nukleotider på sekvens position i.

4.7 Support Vektor Maskine

Både RNAz og RNAmicro anvender support vektor maskiner (SVM) til at klassificere med. SVM er en maskinlærings metode, der bruges til at løse klassifikations problemer. Klassifikations algoritmers formål er at finde en mapping funktion fra et objekts egenskaber til en klasse.

En SVM lærer at klassificere objekter ud fra eksempler. Hver eksempel består af en input vektor, og en angivelse af hvilken klasse objektet tilhører, oftest anvendes to klasser. Input vektoren beskriver egenskaberne ved objektet, og disse skal være numeriske værdier (Sætrom and Snøve, 2007). Essencen i klassificering er at minimere sandsynligheden for fejl i brugen af den trænede klassificeringsalgoritme (Yang, 2004).

Kernen i SVM klassifikation kan forklares med fire begreber: A) det separerende hyperplan, b) maksimun-margen hyperplanen, c) soft margenen og d) kernel funktionen (Noble, 2006).

Den separerende hyperplan: Et punkt kan dele en en dimensionel linie. I to dimensioner kan en linie dele fladen i to halvdele. I tre dimensioner kan et plan dele rummet i to dele. Hyperplanen er den generelle betegnelse for en ret linie i et høj dimensionel rum. Den separerende hyperplan er en plan der deler et høj dimensionelt rum.

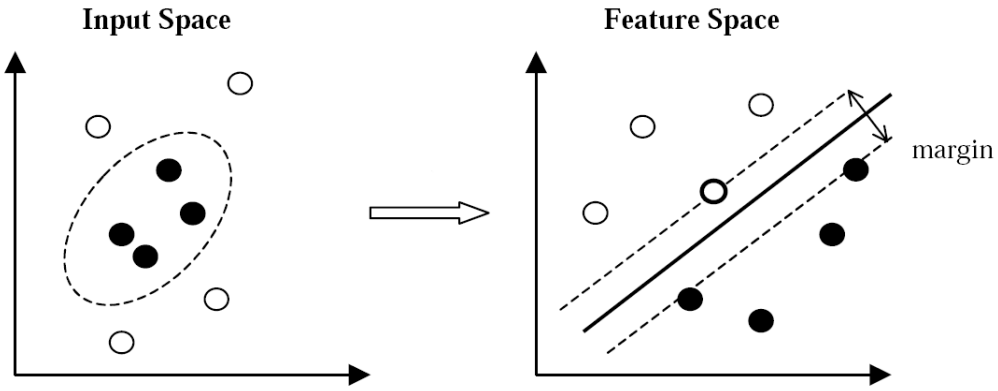
Maksimun-margen hyperplanen: Hvis afstanden fra den separerende hyperplan til de nærmeste input vektorer betegnes margenen til hyperplanen, så vil SVM maksimere denne afstand til de korrekt klassificerede vektorer, og få maksimun-margen hyperplanen, se figur 24.

Soft margenen: I praksis kan det meget sjældent lade sig gøre at finde en hard margen hyperplan. det vil sige en hyperplan, der gør at alle eksempler bliver klassificeret korrekt. For at SVMen kan håndtere disse fejl modificeres SVM algoritmen en soft margen. Soft margener minimere den den samlede afstand til alle forkert klassificerede eksempler.

Kernel funktionen I det tilfælde at en lineær adskillelse ikke er muligt fås hyperplanet ved at transformere input vektorerne i "input space" ind i et høj dimensionel "feature space" ved hjælp af en kernel funktion, se figur 24.

Efter at en SVM er blevet trænet på et sæt kendte eksempler. Anvendes den på at klassificere nye data.

På det seneste er det blevet ret populært at anvende SVM metoden til at identificere potentielle miRNAer. Følgende programmer er bl.a. blevet udviklet: miR-abela (Sewer et al., 2005), Microprocessor SVM og miRNA SVM (Helvik et al., 2007), miPred (NG Kwang Loong and Mishra, 2007), miMatcher (Lindow et al., 2007) og mirCoS (Sheng et al., 2007).



Figur 24: Kortlægning af input data ind i en højere dimonsionel "feature space". Problemer der ikke kan adskilles lineært i "input space" kan adskilles lineært i "feauture space". Hvide- og sorte cirkler er de to klasser. Input vektorerne på marginerne betegnes support vektorer. Modificeret udgave af figur fra Sun and Fan (2003).

5 Pipelinen

Pipelinen er implementationen af strategien, der blev beskrevet i det foregående afsnit, til identifikation af potentielle miRNA gener vha. komparativ genomanalyse.

Pipelinen består af følgende tre overordnede dele:

- Forbehandling
- Konserverede DNA-sekvenser
- Identifikation

Forbehandling: Første del af pipelinen består i at gøre inputdatasættet, de tre valgte arters DNA-sekvenser, klar til at kunne anvendes i resten af pipelinien.

Konserverede DNA-sekvenser: Anden del af pipelinen består i at finde de DNA-sekvenser, der er evolutionært konserveret mellem de tre valgte arters genomer. Dette gøres vha. to parvise alignments, og efterfølgende samle resultaterne fra de to alignments til sæt af konserverede DNA-sekvenser.

Identifikation: Tredje del af pipelinen består i at foretage en multiple alignment af hver af de sæt af konserverede DNA-sekvenser, som er blevet fundet i det foregående del af pipelinen. Efterfølgende analyseres disse multiple alignments nærmere for at finde potentielle miRNA gener.

Alle tre dele dele af pipelinen vedrører strategiens første trin, at det er DNA-sekvenser fra tre beslægtede dyrearter, pipelinen behandler. Anden del af pipelinen vedrører strategiens andet trin identifikation af konserverede DNA-sekvenser. Tredje del af pipelinen vedrør de to sidste trin i strategien, identifikation af ncRNA-sekvenser og potentielle miRNA gener, men er også afhængig af anden del af strategien.

Først beskrives hver af de tre overordnede dele af pipelinen i detaljer, og i det efterfølgende afsnit beskrives de programmer som pipelinen består af.

5.1 Forbehandling

Forbehandlingen består i at gøre indputdatasættet, DNA-sekvenserne for de tre valgte arter, klar til at kunne anvendes i resten af pipelinen.

DNA-sekvenserne der anvendes til input til pipelinen findes i filer på FASTA-formatet. FASTA-formatet er beskrevet i Appendiks A Fil-formater.

Der er to forskellige forbehandlingspipelines, en for FASTA-filer med en enkelt DNA-sekvens og en for FASTA-filer med flere DNA-sekvenser. De to pipelines er vist på henholdsvis figur 25, som viser forbehandligs delen af pipelinen for FASTA-filer bestående af en enkelt DNA-sekvens, og figur 26, der viser forbehandligs delen af pipelinen for FASTA-filer med mere end en DNA-sekvens.

Grunden til at der er to forbehandlingspipelines er, at for flere arter f.eks. mennesket findes en FASTA-fil med en sekvens for hver enkelt kromosom. Hvorimod for f.eks. fugu findes en FASTA-fil med hele genomet, som er delt op i mange sekvenser, fordi man endnu ikke har fået samlet DNA-sekvenserne kromosomvis, men kun i mindre delsekvenser. For fugus vedkommende kaldes disse delsekvenser scaffolds.

En mulighed er at splitte en FASTA-fil med flere sekvenser op i filer med kun en sekvens i hver, og så benytte pipelinen for FASTA-filer med en enkelt DNA-sekvens, men det bliver til 20.379 filer hvis fugu assembly 2.0 benyttes og til 7.213 filer hvis assembly 4.0 benyttes. Men det store antal filer vil være ret uhåndterlige, derfor er løsningen med to pipelines valgt.

Pipelinen til at håndtere scaffolds, er en udbygning af pipelinen for enkelt sekvenser, til også at kunne håndtere filer med flere sekvenser. At pipelinen til scaffolds ikke benyttes til alle filer inklusiv dem med en enkelt sekvens er, at der foretages flere ting, når scaffolds skal håndteres, som er unødvendige, når der er tale om FASTA-filer med enkeltsekvenser. Og på et tidspunkt bliver fugus scaffolds formodentlig samlet, så DNA-sekvenserne kommer med en DNA-sekvens pr. kromosom.

De to forbehandlingspipeliner består begge af følgende dele:

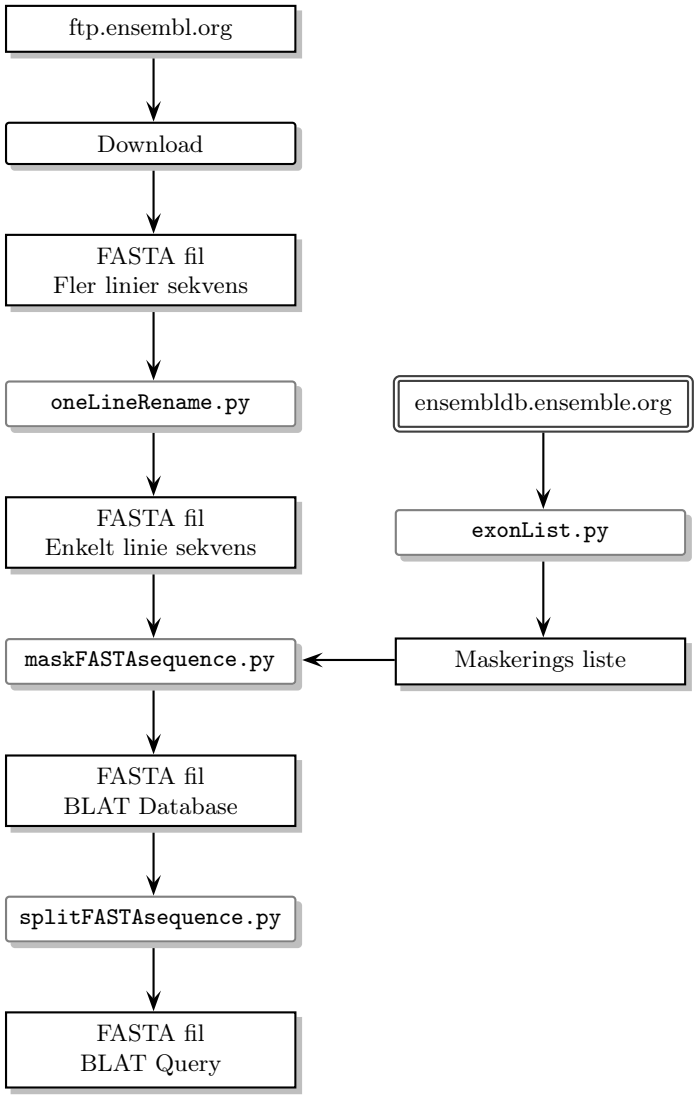
- Download af DNA-sekvenserne – FASTA-format
- Reformatering af FASTA-filer
- Maskering af DNA-sekvenserne
- Opdeling af lange DNA-sekvenser i kortere sekvenser

Desuden har forbehandlingspipelinen for scaffolds også følgende dele:

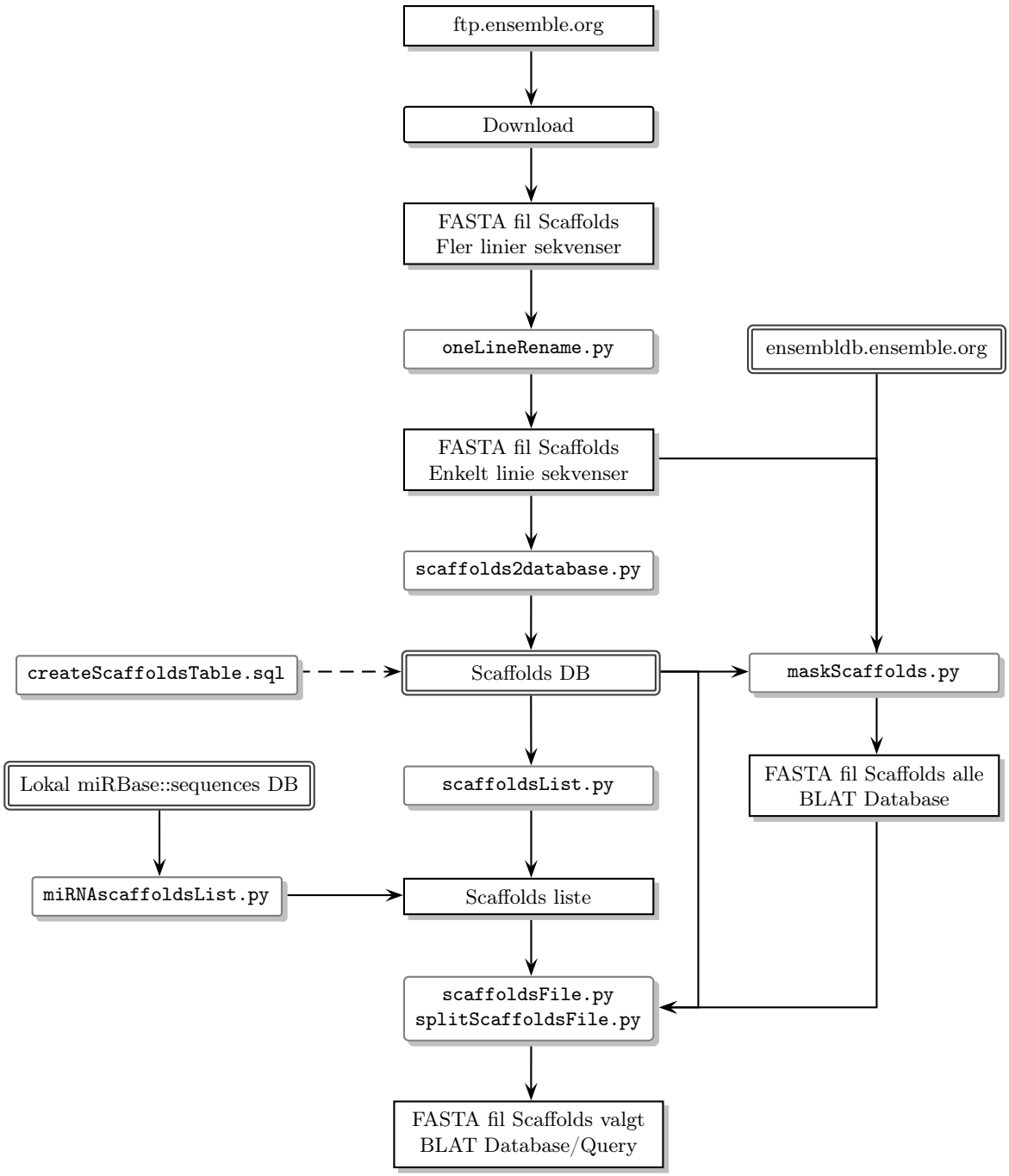
- Registrering af DNA-sekvensernes start positioner
- Generering af scaffoldsliste

5.1.1 Download af DNA-sekvenser

DNA-sekvenserne der benyttes som input til pipelinen, forventes at være downloaded fra Ensembl's ftp-website: `ftp.ensembl.org`, eller de skal



Figur 25: Forbehandlings pipeline for FASTA-filer med en enkelt DNA-sekvens.



Figur 26: Forbehandlings pipeline for FASTA-filer med mere end en DNA-sekvens – Scaffolds.

have et format som de DNA-sekvenser, der kan downloades fra Ensembl's ftp-website. DNA-sekvenserne, der kan downloades fra Ensembl er på FASTA-formatet, og findes som umaskerede DNA-sekvenser og som maskerede DNA-sekvenser. De sidstnævnte er maskeret med RepeatMasker (`www.repeatmasker.org`), som er et program, der screener DNA-sekvenser for interspersed repeats og lav kompleksitets DNA-sekvenser, disse sekvenser bliver maskeret ved at baserne A, C, G og T bliver erstattet med et tilsvarende antal N'er.

Det skal være de maskerede DNA-sekvenser, som skal avendes i pipelinen, da det har vist sig, at RNAz ellers vil klassificere repeat sekvenserne som ncRNA sekvenser, og RNAmicro klassificerer dem som miRNAer. Om det er alle eller kun en del af repeat sekvenserne, der fejlklassificeres er ikke undersøgt.

Ensembl er valgt, fordi de ofte opdaterer deres website `www.ensembl.org` med de nyeste data, det sker ca. hver anden måned. Tidligere udgaver fra de sidste to år er tilgængelige fra deres arkiv website `archive.ensembl.org`. Det er derfor også muligt at få adgang til bestemte udgaver af DNA-sekvenserne. Fra Ensembl's database `ensembldb.ensembl.org` er det desuden muligt at hente yderligere information om forskellige arters genomer, som f.eks. exon positioner, der bliver anvendt i denne pipeline.

5.1.2 Reformatering af FASTA-filer

Reformateringen af de downloadede FASTA-filer har til formål at gøre filerne mere praktisk anvendelige i resten af pipelinen. Ændringerne der skal foretages vedrører informationslinierne, filnavnene og nylinietegnene i FASTA-filerne med DNA-sekvenserne. For den måde informationslinierne i de downloadede FASTA-filer er formateret på er uhensigtsmæssig, når der skal foretages alignment af sekvenserne. Grunden er den, at alignment programmerne BLAT og BLATz kun medtager identifieren fra informationslinien i output filen når output formatet er axt. I dette projekt forudsættes det, at det er plus sekvenser, der reformateres, da det er den type af sekvenser, der kan downloades fra Ensembl.

Eksempel på informationslinien for en FASTA fil med en DNA-sekvens for et kromosom:

```
>17 dna:chromosome chromosome:NCBI36:17:1:78774742:1
```

Og eksempel på en af informationslinierne i en FASTA fil med scaffolds:

```
>scaffold_1 dna:scaffold scaffold:FUGU4:scaffold_1:1:7245445:1
```

Når kun identifieren medtages i outputfilen, kan det være svært at holde styr på, hvilken art en sekvens tilhører, når en sekvens kun refereres med et nummer f.eks. nummeret 17 i en alignment. Det ville derfor være nødvendig at holde styr på hvilken art 17 refererer til på en eller anden måde. Det er uhensigtsmæssig. Det er derfor nødvendigt at ændre på identifieren til noget mere praktisk anvendelig og entydig. Alle de informationer, der skal bevares igennem en alignment, skal placeres i selve identifieren. Derudover vil der ofte være tale om delsekvenser, der arbejdes med i pipelinen, derfor er delsekvensernes positioner i hele kromosomet også relevant at have med i identifieren. Da et kromosom består af to DNA-strenge er det også væsentligt at vide, hvilken streng der er tale om.

Informationerne der skal være i identifieren, for at gøre den mere anvendelig er art, kromosom- eller scaffoldnummer, sekvensstart og -slut og streng.

Formatet på identifieren for FASTA-filer bliver:

```
>Art-Kromosomnummer:Sekvensstart:Sekvensslut:Streng
```

Art-Kromosomnummer er artnavn eller lign. entydig betegnelse for arten samt kromosomnummer. Hvis outputtet fra den samlede pipeline skal analyseres for kendte miRNAer, skal artsnavnet være ligesom artsbetegnelserne, der benyttes i miRBase, f.eks. "hsa" for mennesket. Og der skal være bindestreg mellem artsnavn og kromosomnummer. Art-Kromosomnummer tilføjet ".fa" bruges som navnet på filen på den reformaterede FASTA-fil.

Formatet for FASTA-filer med en række scaffolds bliver:

```
>Art_scaffoldnummer:Sekvensstart:Sekvensslut:Streng
```

For filer med scaffolds er der tale om scaffoldnummer i stedet for kromosomnummer. Art og scaffoldnummer skal være adskilt med understregningstegnet "_", da tegnet bruges til at udtrække scaffoldnummeret fra identifieren i implementationen. For FASTA-filer med scaffolds bliver filnavnet art tilføjet ".fa".

Sekvensstart er nummeret på den base, som sekvensen starter med i den reformaterede FASTA-fil. Og tilsvarende for sekvensslut, som er nummeret på den base som sekvensen slutter med i den reformaterede FASTA-fil. I den reformaterede FASTA-fil med et helt kromosom vil sekvensstart være 1 og sekvensslut svare til længden på sekvensen dvs. antal baser.

Streng angiver hvilken DNA-streng der er tale om, plus strengen angives med tegnet "+" og tilsvarende minus strengen med tegnet "−".

De enkelte informationer skal adskilles med kolon ":", da dette tegn benyttes til at adskille de forskellige informationer i implementationen, således at det er muligt at udtrække de enkelte informationer fra identifieren, til

videre behandling i de enkelte programmer som pipelinen består af.

Et eksempel på en reformateret identifier for et helt kromosom:

```
>hsa-17:1:78774742:+
```

Et eksempel på en reformateret identifier for en delsekvens:

```
>hsa-17:27691241:27711334:+
```

Sekvensstart og sekvensslut positionerne refererer altid til positioner i den reformaterede FASTA-fil indeholdende plus strengen, også selvom en FASTA sekvens er en minus streng, dog er minus strengen vendt korrekt med 5'-enden som start på FASTA sekvensen og 3'-enden som slut på sekvensen. Dette er gjort, fordi så er der ingen steder, hvor der skal anvendes positioner eller korrigeres positioner, er nødvendigt at kende sekvenslængden i de reformaterde FASTA-filer.

DNA-sekvenserne i de downloadede filer er delt op på enkelt linier på 60 baser, og hver linie afsluttes med nylinietegnet "\n". Det betyder DNA-sekvensen ikke blot er en lang streng af baser, men baserne bliver hyppigt afbrudt af nylinietegn.

Ved at fjerne alle nylinietegn på nær den sidste, gøres det programmeringsteknisk meget simplere at udtrække delsekvenser fra FASTA filerne, når der ikke hele tiden skal tages højde for de ellers mange nylinietegn. Og udtrækning af delsekvenser foretages flere steder i den efterfølgende del af pipelinen. Den afsluttende "\n" skal være der, for ellers laver BLAT fejl i positionerne for de sekvenser, der bliver aligned.

Reformateringen foretages med programmet `oneLineRename.py`.

5.1.3 Maskering af DNA-sekvenserne

For at begrænse outputtet fra og kørselstiden for de forskellige alignments, maskeres exons for proteinkodende gener ud, evt. andre gener kan også maskeres ud. Da miRNA gener kan forekomme i proteinkodende geners introns, er det kun exons og ikke hele sekvensen for et gen, der skal maskeres ud.

Først genereres en maskeringsliste, som er en liste med de startpositioner og slutpositioner for delsekvenser, der skal maskeres, så de ikke indgår i den efterfølgende alignment. Oplysningerne om exons positioner udtrækkes fra Ensembl's online databaser (host: `ensembldb.ensembl.org`).

Det er også muligt at generer maskeringsliste med Ensembl's interaktive datamining system BioMart (`http://www.ensembl.org/biomart/martview/`).

Maskeringen kan enten være hard eller soft. Hard maskering er når baser i sekvenser, der skal maskeres konverteres til N'er. I de downloadede FASTA filer er baserne skrevet med store bogstaver (ACGT), soft maskering konverterer baserne til små bogstaver (acgt).

Til at hente exon oplysninger anvendes programmet `exonList.py.`, og til maskeringen anvendes `maskFASTAsequence.py`, hvis der er tale om filer med en DNA-sekvens. Er der tale om filer med scaffolds, så anvendes programmet `maskScaffolds.py`,

5.1.4 Opdeling af lange DNA-sekvenser i kortere sekvenser

De to sekvenser der indgår i en BLAT alignment kaldes henholdsvis database-sekvensen og query-sekvensen. Der er ikke en specifik grænse for hvor lange database-sekvenserne må være, det er muligvis afhængig af computerens hukommelse. Til gengæld må query-sekvensen maksimal være 200.000 baser lange (Kent, 2002) og for at få den bedste performance ved aligment af DNA-sekvenser mellem to arter bør query-sekvenserne deles op i sekvenser på maksimalt 25.000 baser (`http://genome.ucsc.edu/goldenPath/help/blatSpec.html`). Det betyder, at hvis de downloadede sekvenser er længere skal de deles op i mindre sekvenser, før de kan indgå som query-sekvenser i en BLAT alignmnent.

Programmet `splitFASTAsequence.py` foretager opsplitningen.

5.1.5 Registrering af DNA-sekvensernes start positioner

For de FASTA-filer, der indeholder DNA-sekvensen for et kromosom i en sekvens, er det hurtigt at finde delsekvenser, når alle "\n" er fjernet fra sekvensen.

Dette er ikke tilfældet med arter, hvor genomet endnu ikke er samlet til kromosom niveau, men kun som scaffolds, altså mindre sekvenser, og alle sekvenserne er samlet i en stor fil. Som f.eks. fugu med ialt 20.379 scaffolds i assembly 2.0 og 7.213 scaffolds i assembly 4.0.

For at finde delsekvenser fra disse arter, på en hurtig måde, er det derfor nødvendigt at kende start positionerne for alle sekvenserne i FASTA filen.

Disse positioner placeres i en database sammen med navnet og nummeret på de enkelte scaffolds. Efterfølgende er det hurtigt at finde delsekvenser fra artens genom. Databasen indeholder også information om sekvensernes længder.

Det er værd at bemærke, at der er to slags positioner for DNA-sekvenser, der er fil positioner, som beregnes ud fra starten af filen, og der er sekvens positioner, der beregnes ud fra fra starten af sekvensen for et helt kromosom eller scaffold.

Registreringen af DNA-sekvenserne start positioner, foretages med programmet `scaffolds2database`.

5.1.6 Valg af scaffolds

Det er ikke altid hensigtsmæssig at bruge alle scaffolds i en alignment. Det er derfor muligt at foretage et udvalg af scaffolds, der efterfølgende kan bruges

til alignment.

Først genereres en liste med numrene på de sekvenser, der skal indgå i en alignment. Det kan være fra oplysningerne i databasen beskrevet ovenfor. Eller fra en lokal installation af miRBase, hvor det er muligt at generer en liste over scaffolds med miRNA gener.

Efterfølgende anvendes listen til at danne en FASTA-fil med de ønskede DNA-sekvenser, som evt. er delt op i mindre sekvenser, hvis FASTA-filen skal ingå som en query-fil i en BLAT alignment.

Til at generer listerne med anvendes enten `scaffoldsList.py` eller `miRNA-scaffoldsList.py`. FASTA-filerne dannes med `scaffoldsFile.py` eller `split-ScaffoldsFile.py`.

5.2 Konserverede DNA-sekvenser

Konserverede DNA-sekvenser delen af pipelinen består af to BLAT alignments, som har til formål at finde de DNA-sekvenser, der er mest evolutionært konserverede mellem de valgte tre arter.

Først foretages en lokal alignment af DNA-sekvenserne fra to af arterne, herefter kaldet første alignment.

Dernæst benyttes de delsekvenser fra en af arterne, der blev alignet i den første alignment, til lokal alignment med den tredje arts DNA-sekvens, herefter kaldet anden aligment.

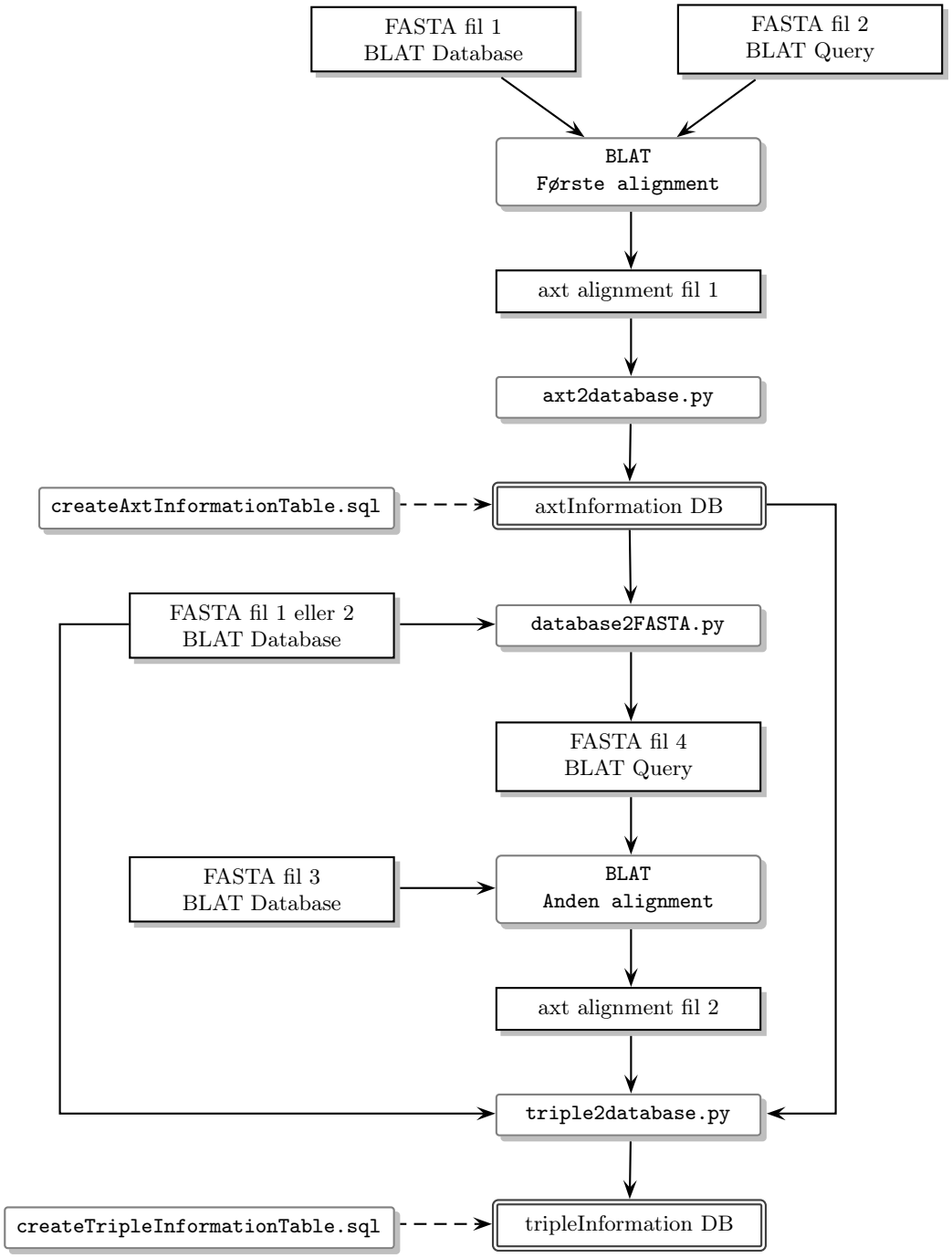
Til slut samles informationerne fra de to alignments til sæt af konserverede DNA-sekvenser. Resultatet placeres i en database, der benyttes til den videre analyse. Konserverede DNA-sekvenser delen af pipelinen er vist i figur 27.

Konserverede DNA-sekvenser delen af pipelinen består af følgende dele:

- Første alignment
- Resultatet fra første alignment placeres i database
- Generering af FASTA-fil til anden alignment
- Anden alignment
- Alignment resultaterne samles og placeres i database

5.2.1 Første alignment

Alignmentet mellem de to første arters DNA-sekvenser foretages med programmet BLAT. Den ene arts DNA-sekvens vælges til at være database, og den anden arts DNA-sekvens vælges til at være query, derfor skal denne arts DNA-sekvens være splittet op i delsekvenser, der ikke er mere end 25.000



Figur 27: Konserverede DNA-sekvenser pipelinen. De to parvise alignments.

nukleotider lange. Output filformatet sættes til axt for alignmentet. axt formatet er valgt, fordi den har de relevante informationer til videre analyse af alignments. axt-formatet er beskrevet i appendiks A Fil-formater.

5.2.2 Resultatet fra første alignment placeres i database

Resultatet fra første alignment som findes i en axt fil, skal placeres i en database.

Det er kun oplysningerne fra informationslinierne i axt filen, der placeres i databasen. Disse oplysninger kan dog ikke direkte placeres i databasen, nogle af oplysningerne skal først korrigeres, så de direkte kan anvendes senere i pipelinen.

Et eksempel på axt formatet:

```
8 hsa-17:27691241:27711334:+ 4239 4288 mmu-11:35615819:35635904:+ 1093 1141 + 3544
gtggctcacgcctgtaatcccagcactttgggaggccgaggcaggcggat
gtggcgcacgcctttaatcccagca-tttggaaggcagaggcaggcggat
```

En BLAT alignment er foretaget mellem to delsekvenser fra mennesket kromosom 17 og musen kromosom 11, men kun position 4239–4288 i menneskets delsekvens er blevet alignet med position 1093–1141 i musens delsekvens. Disse positioner er relative i forhold til input sekvenserne, de skal derfor omregnes til positioner i de hele kromosomer og dermed positioner i de reformaterede FASTA-filer, hvor første base har position 1. Da disse positioner er meget simplere at anvende senere i pipelinen.

For menneskets vedkommende bliver startpositionen $27691241 + 4239 - 1 = 27695479$ og slutpositionen $27691241 + 4288 - 1 = 27695528$. Tilsvarende beregninger gøres for musens vedkommende.

Hvis informationsliniens felt nr. 8 "+" havde været et minus "−", så havde musens start- og slutposition for alignmentet været relative i forhold til den omvendt komplementære input sekvens, og startpositionen omregnes til $35635904 - 1141 + 1 = 35634764$ og slutpositionen omregnes til $35635904 - 1093 + 1 = 35634812$. Så også disse positioner bliver til positioner i den reformaterde FASTA-fil. Oplysningen om at der er tale om den omvendt komplementære sekvens, altså minus sekvensen, gemmes også i databasen.

Det er derfor ikke nødvendigt at gemme selve DNA-sekvenserne i databasen, da de kan genfindes ud fra informationerne i databasen og de reformaterede FASTA-filer.

Resultatet placeres i en database, fordi det gør det mere fleksibelt at vælge de alignments, der skal arbejde videre med. Samtidig er det meget hurtigere at finde aligment informationerne om den art, der ikke indgår i den anden alignment, når disse informationer skal sættes sammen med alignment resultaterne fra den anden alignment, end hvis de skulle findes i axt-filen.

Programmet `axt2database.py` placerer resultatet i `AxtInformation` databasetabellen, der oprettes med `createAxtInformationTable.sql`.

5.2.3 Generering af FASTA-fil

Ud fra resultatet fra den første alignment genereres en FASTA-fil, der skal bruges til den anden alignment. Filen genereres ud fra oplysningerne i databasen med oplysningerne om den første alignment, og selve sekvenserne hentes fra den relevante reformaterede FASTA-fil. Filen indeholder de delsekvenser fra en af arterne, der kunne alignes i den første alignment, eller et udvalg af disse DNA-sekvenser. Identifieren for de enkelte sekvenser i den genererede FASTA-fil, er udvidet med oplysningen om postnummeret i databasen, hvor oplysningerne kommer fra.

Formatet på identifierne i den genereret FASTA-sekvens bliver:

```
>Art-Kromosomnummer:Sekvensstart:Sekvensslut:Streng:Postnummer
```

Et eksempel på identifierne i den genererede FASTA-fil:

```
>mmu-12:109053353:109053403:+:1
```

Postnummeret gør det muligt hurtigt at finde oplysningerne fra den art der ikke indgår i den anden alignment, når oplysningerne for de tre arter skal samles efter den anden alignment.

Den genererede FASTA-fil kan består af DNA-sekvenser, som både kan være plus og minus strenge. Hvis alle sekvenser skal være plus strenge, og alle alignede sekvenser skal med i FASTA-filen, skal sekvenserne for database arten benyttes. Da sekvenserne for denne art altid vil være plus sekvenser.

FASTA-filen genereres med programmet `database2FASTA.py`.

5.2.4 Anden alignment

Anden alignment er alignmentet mellem den tredje art, der ikke indgik i første alignment, og en af arterne fra den første alignment. Det er også BLAT programmet, der benyttes her til alignmentet som i første alignment. Den fil, der anvendes som database i denne BLAT alignment må kun indeholde plus sekvenser. Ellers vil implementationen for den følgende del af pipelinen ikke virke korrekt.

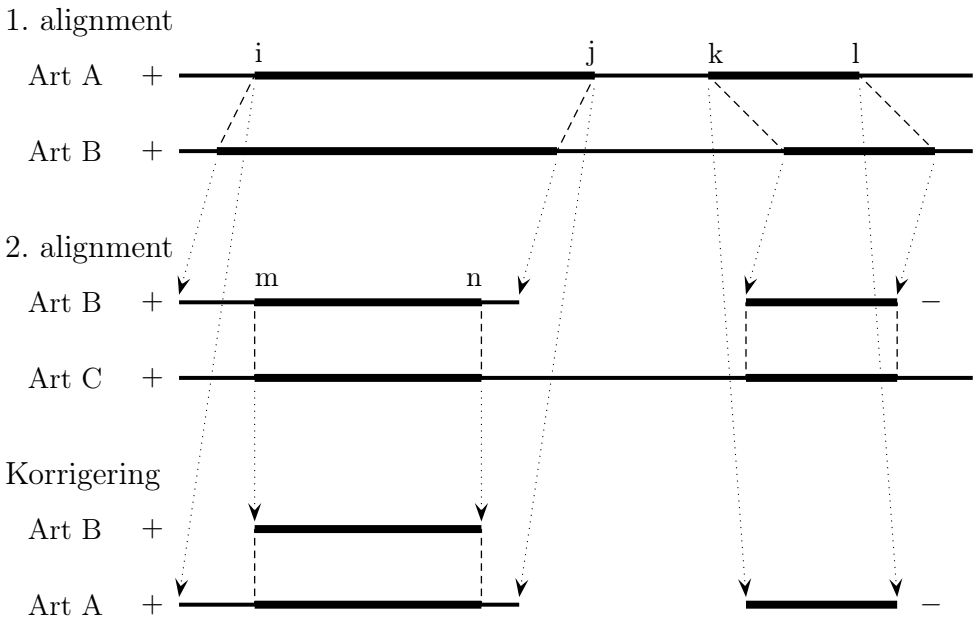
Det vil være logisk at anvende den tredje art, som database arten i alignmentet, og arten fra første alignment, som query art. Det vil derfor ikke være nødvendigt at splitte sekvenserne op i en af filerne, da ingen af sekvenserne i query arts FASTA-fil vil være på mere end 25.000 baser. I denne alignment skal output fil formatet også være axt format.

5.2.5 Alignment resultaterne samles og placeres i database

Resultatet fra den anden alignment placeres i en database. Lige som ved den første alignment, er det kun oplysningerne fra informationslinierne, der placeres i databasen. For de sekvenser der ingik i anden alignment, skal sekvens positionerne også korrigeres, som beskrevet for første alignment. Da query filen kan indeholde minus sekvenser, tages der højde for dette i korrektionen af sekvens positionerne.

Resultatet skal samles med resultatet fra den første alignment, således at informationerne fra den art, der ikke er med i den anden alignment kommer med i databasen. Informationern for denne art skal muligvis korrigeres. Der er to ting der kan korrigeres for længden af sekvensen og hvilken streng, der skal anvendes i det følgende af pipelinen.

Det er ikke sikkert, at det det hele af en sekvens fra den første alignment der kan alignes i den anden aligment, men kun en del af en sekvens, hvis det er tilfældet skal sekvensen for arten, der ikke er med i den anden alignment også afkortes. Dette er illustreret til venstre i figur 28.



Figur 28: De to parvise alignments og den efterfølgende korrektion, som anvendes før resultatet samles i en database. Til venstre er længde korrektionen og til højre er streng korrektionen. Se teksten for nærmere forklaring.

I første alignment kan delsekvens A_{ij+} fra art A alignes med delsekvens B_{ij+} fra art B. Sekvens B_{ij+} anvendes i anden alignment til alignment med Art C, men kun delen B_{mn+} af delsekvens B_{ij+} kan alignes med art C's delsekvens C_{mn+}. Derfor skal art A's delsekvens A_{ij+} afkortes, inden oplysningerne placeres i databasen. Den del af A_{ij+} der er interessant, er den del, der kan alignes til B_{mn+}.

Det var meningen at anvende BLAT til denne afkortning, men det viste sig at være en ikke særlig god ide. Det viser sig at benyttes BLAT til at genaligne to delsekvenser, der danner et alignment når to sekvenser alignes, så er det ikke sikkert at de to delsekvenser bliver genalignet i hele deres længde, men kun en del eller flere dele af delsekvenserne. Løsningen blev at anvende BLATz her, da programmet er mere sensitiv end BLAT, og hvis der er flere del alignements, er det muligt at vælge den med højeste score.

Den korrekte løsning på problemet er at anvende dynamisk programmering og global alignment, hvor gaps i enderne af den korteste streng ikke koster noget. At denne løsning ikke blev implementeret var, at det først sent i projektet blev opdaget, at der var et problem med at anvende BLAT på dette sted i pipelinen.

Det kan være, at det er den komplementære streng til en sekvens fra den første alignment, der kan alignes med den tredje arts sekvens i anden alignment, så skal strengen også ændres til den komplementære streng for arten der ikke er med i den anden alignment. Dette er illustreret til højre i figur 28

I første alignment kan delsekvens A_{kl+} fra art A alignes med delsekvens B_{kl+} fra art B. Sekvens B_{kl+} anvendes i anden alignment til alignment med Art C, men det er den omvendt komplementære sekvens B_{kl-} kan alignes med art C's delsekvens C_{kl+}. Derfor skal art A's delsekvens A_{kl+} konverteres til minus strengen A_{kl-}, inden resultatet placeres i databasen.

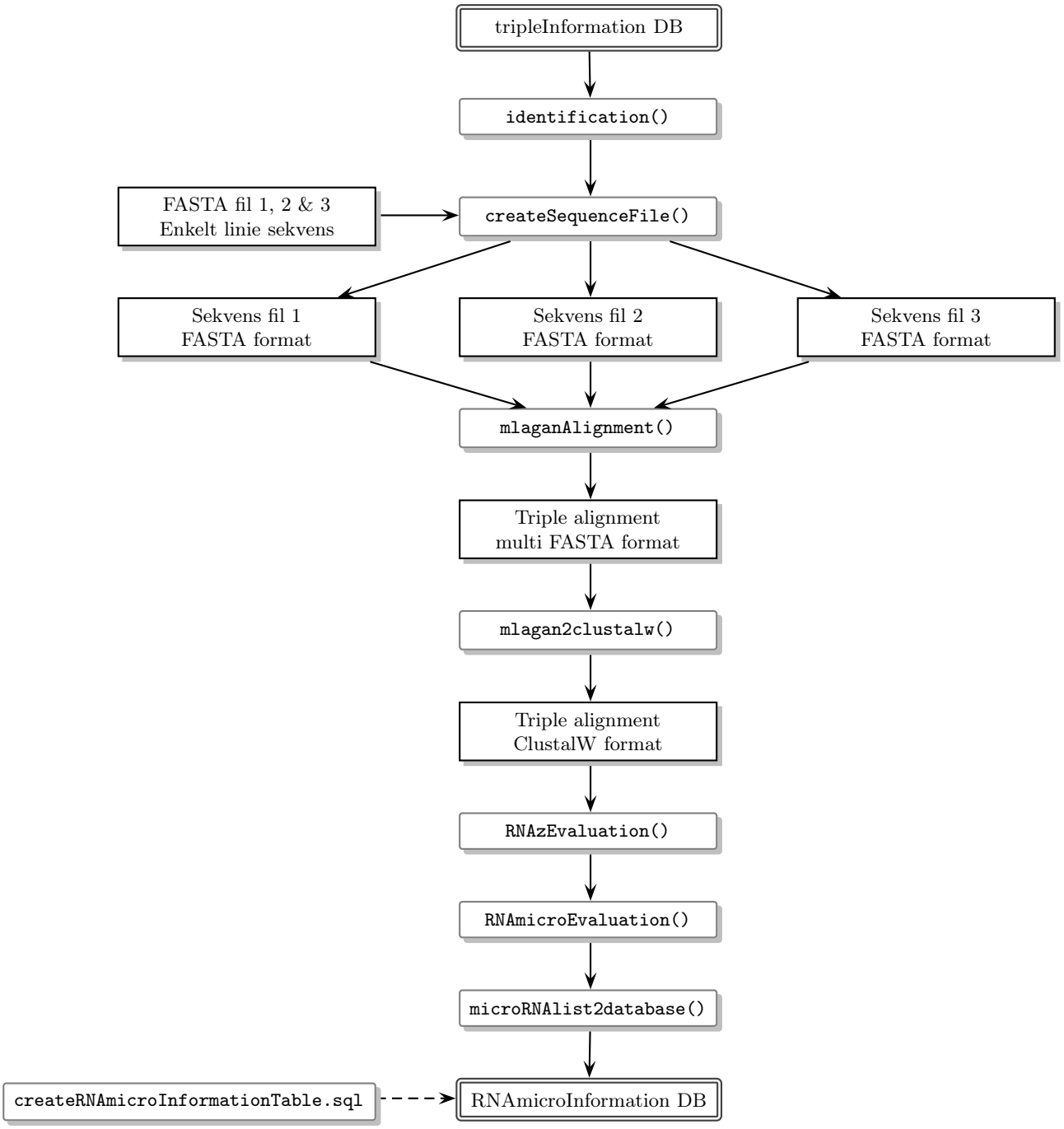
Programmet der samler resultaterne fra de to BLAT alignments er `triple-2database.py`, den placerer også resultatet i databasetabellen `tripleInformation`, der oprettes med `createTripleInformationTable.sql`.

5.3 Identifikation

Identifikation delen af pipelinen består i at finde de DNA-sekvenser der potentielt indeholder miRNA gener, blandt de sekvenser der blev fundet ved de to BLAT alignments. Pipelinedelen for identifikation er vist i figur 29.

Identifikations pipelinen består af følgende dele:

- Generer DNA-sekvens filer
- Multiple alignment
- Konverter til ClustalW-format



Figur 29: Identifikations pipelinen. Identifikationen af potentielle miRNA gener i multiple alignede DNA-sekvenser.

- ncRNA klassificering
- Identificer potentielle miRNA gener
- Identifikations resultaterne placeres i database

5.3.1 Generer DNA-sekvens filer

Der genereres en FASTA-fil for hver af de tre arter, der indgår i analysen. De tre filer indeholder hver en af de DNA-sekvenser, der kan alignes til hinanden, og blev fundet gennem de to BLAT alignments.

De generede FASTA-filer vil indeholde en DNA-sekvens på minimum 200 nukleotider, for at sikre at hele pre-miRNA sekvensområdet er med i alignmentet. Hvis BLAT alignmenterne resulterede i kortere sekvenser, bliver de forlænget til 200 nukleotider, og forlængelsen er lige fordelt på de to ender af sekvensen.

Det er ikke nødvendigt at konvertere DNA-sekvenserne til RNA-sekvenser, for det gør programmerne RNAz og RNAmicro selv.

5.3.2 Multiple alignment

Multiple alignmentet af de tre DNA-sekvenser foretages med MLAGAN. Outputtet er en fil på multi-FASTA-format, dvs. at alle tre DNA-sekvenser er i den samme fil på FASTA-format.

5.3.3 Konverter til ClustalW-format

RNAz kan ikke håndtere multi-FASTA-formatet, derfor konverteres multi-FASTA-format filen til ClustalW-formatet, som programmet godt kan håndtere. ClustalW-formatet er beskrevet i Appendiks A Fil-formater.

5.3.4 ncRNA klassificering

Multiple alignmentet søges igennem for at finde sekvensdele, der kan klassificeres som ncRNA sekvenser. Dertil bruges programmet RNAz. Alignmentet søges igennem med et sekvens vindue i passende størrelse og i passende trin. Der søges kun indtil, der findes en sekvensdel, der bliver klassificeret til at være ncRNA-sekvens. En sekvensvindue klassificeres som ncRNA, hvis RNAz giver den en score på 0,50 eller mere. Hvis der ikke findes nogen ncRNA-sekvens i alignmentet, søges den omvendte komplementære alignment også igennem.

Ovennævnte klassificering med at hele alignmentet bliver klassificeret som værende ncRNA sekvens eller ikke, er en ret simpel klassificering. En mere detaljeret klassificering kunne være at finde de områder i multiple alignmentet der blev klassificeret som ncRNA, og kun gå videre med de sekvensdele, i den videre analyse. Men programmet RNAmicro, der bruges

efterfølgende til at identificere miRNA gener, kan ikke bruges med "fra" og "til" parametre, hvis alignmentet har en længde på mere end 200 nukleotider, og kan derfor ikke anvendes til analyse af en begrænset område af en længere multiple alignment.

En mulig løsning på dette ville være at placere de dele af alignmentet, der blev klassificeret som ncRNA sekvenser, i filer med en maksimal længde på 200 nukleotider. Men det ville være en mere kompleks løsning, her er valgt at bruge den simple klassificering.

Det er ikke blevet undersøgt nærmere, hvilken betydning, det har for det samlede resultat, at en større del af multiple alignmentet bliver analyseret med RNAmicro, end den del som RNAz klassificere som ncRNA.

5.3.5 Identificer potentielle miRNA gener

De multiple alignments der er blevet klassificeret til at indeholde ncRNA sekvenser, søges igennem for at finde sekvensdele, der kan klassificeres som potentielle miRNA gener. Programmet RNAmicro bruges til at identificere potentielle miRNA gener. Hele alignmentet søges igennem med sekvens vinduer med passende størrelser og med trin af en størrelse betydeligt mindre end vindues størrelsen.

Der skal helst anvendes flere forskellige vindues størrelser, for at dække flere størrelser af pre-miRNAer. RNAmicro kan håndtere vindues størrelser fra 70 nt. til 130. nt, og vinduet flyttes automatisk i trin på 5 nt.

Den omvendt komplementære alignment søges også igennem. Hver sekvensvindue gives en score af RNAmicro, og hvis denne score er på 0,50 eller mere klassificeres den som en mulig miRNA gen.

Når alignmentet søges igennem med forskellige vindues størrelser og med små trin, vil der forekomme overlap mellem delsekvenser, der identificeres som mulige miRNA gener. Disse overlap bliver slået sammen, hvis de har samme score. Hvis der er overlap med forskellig score, vil den overlap med laveste score blive fjernet.

5.3.6 Identifikations resultaterne placeres i database

Oplysningerne om de delsekvenser, for de tre arter, der indgik i multiple alignmentet, og som blev klassificeres som potentielle miRNA gener placeres i en database. Med oplysninger om sekvens positioner og streng samt RNAmicro scoren. Selve sekvenserne bliver ikke gemt i databasen. De vil altid kunne findes udfra oplysningerne i databasen. Efterfølgende er det er muligt at analysere videre på resultatet.

Programmet `identification.py` foretager hele identifikationen af potentielle miRNA gener, og placere resultatet i databasetabellen `RNAmicro-Information`, der oprettes med `createRNAmicroInformationTable.sql`.

6 Programmerne

Pipelinen til identifikation af potentielle miRNA gener består af dels eksisterende programmer og dels nyudviklede programmer. Udover de programmer, der indgår i selve pipelinen, er der udviklet forskellige hjælpeprogrammer.

Først beskrives kort hvilke eksisterende programmer, der indgår i pipelinien. Derefter beskrives nogle generelle ting om de nyudviklede programmer. Efterfølgende beskrives de nyudviklede programmer, der indgår direkte i pipelinien i detaljer. Til slut beskrives hjælpeprogrammerne.

6.1 Eksisterende programmer

Eksisterende programmer er programmer, som andre har udviklet, og som ikke er udviklet til pipelinen, men indgår i pipelinen.

Et overordnet krav til disse programmer har været, at de skulle være gratis tilgængelige minimum til ikke kommercielt brug og kunne køre på DAIMIs Linux installation, som er Fedora (`http://fedoraproject.org`).

De eksisterende programmer er vist i tabel 2. Diskussion om hvorfor lige netop disse programmer er valgt fremfor andre med samme funktionalitet findes i afsnittet Strategien.

Navn	Version	Type	Reference
BLAT	33	parvis alignment	Kent (2002)
BLATz	1	parvis alignment	
MLAGAN	1.1	multiple alignment	Brudno et al. (2003)
RNAz	1.0pre	ncRNA identifikation	Washietl et al. (2005)
RNAmicro	1.1	miRNA identifikation	Hertel and Stadler (2006a)

Tabel 2: Eksisterende programmer benyttet i pipelinen. Programmer der er udviklet af andre, og som ikke er udviklet direkte til brug i pipelinen.

6.2 Nyudviklede programmer

De nyudviklede programmer, som indgår i pipelinen, er udviklet i programmeringssproget Python version 2.4.3 (`http://python.org`). Desuden er MySQL version 4.1.10a-max (`http://mysql.com`) benyttet som databasemanagementsystem. Til kommunikationen mellem Python programmerne og MySQL benyttes Python modulet MySQLdb version 2.0 (`http://sourceforge.net/projects/mysql-python`).

De nyudviklede programmer, kan deles i to grupper af kildefiler biblioteksmodulerne og de egentlige programmer.

Biblioteksmodulerne indeholder funktioner, som de øvrige programmer benytter sig af. Hvis mere end et program benytter en funktion, er den placeret i et af biblioteksmodulerne, hvorfra det enkelte program kan importere det fra, og rettelser skal kun foretages et sted. Der er dog enkelte steder, hvor dette princip er fraveget, hvis et program benytter en større og væsentlig funktion fra et andet program, så importeres den fra det andet program, så koden findes kun et sted. Faktisk kunne alle funktionerne placeres i et bibliotek, hvorfra de enkelte funktioner kunne importeres fra til det enkelte program. Dette er dog ikke gjort her for overskuelighedes skyld. Det er lettere at få et overblik over et program, når dens funktioner er samlet.

Alle egentlige programmer er opbygget efter en fælles model. De har alle en linie med følgende udformning:

```
if __name__ == "__main__":
```

Denne sætning vil kun blive udført, hvis den Python program-fil den findes i, udføres som "main" programmet, dvs. at det er den fil, som programmet startes fra. Det første der sker når et program køres er, at der tjekkes for om programmet er kaldt med det rette antal parametre. Resten foregår ved funktionskald, da den resterende del af koden er pakket ind i funktioner.

6.2.1 Biblioteksmodulerne

Der er to biblioteksmoduler `myDataLibrary.py` og `myLibrary.py` Biblioteksmodulerne indeholder alle de funktioner som benyttes af mere end et program, med programmer menes alle programmer der er udviklet i forbindelse med dette projekt.

myDataLibrary.py

Modulet indeholder funktioner, som indeholder data, der benyttes af de øvrige programmer. Disse funktioner er placeret i en separat fil, fordi de indeholder data, som evt. skal ændres. Det drejer sig om databasekald og en liste over de arter, hvor sekvenserne er i scaffolds, dvs. flere sekvenser i den samme fil.

myLibrary.py

Modulet indeholder alle øvrige funktioner.

6.3 Pipelineprogrammerne

De nyudviklede pipelineprogrammer er de programmer, der er udviklet til at indgå direkte i pipelinen.

6.3.1 Forbehandling

De følgende programmer indgår i forbehandlings pipelinen for FASTA-filer med en enkelt DNA-sekvens, se også figur 25.

oneLineRename.py

Programmet foretager ændringen af informationslinier i FASTA-filer, så identifierne bliver mere informative når filerne indgår i en BLAT alignment.

Desuden fjernes alle ny linie "\n", der er placeret i selve DNA-sekvensen. "\n" efter sekvensen bevares, hvilket er vigtigt, for ellers vil BLAT aligmentet give et forkert resultat. BLAT vil regne informationslinien med i sekvensen, det medfører at positionerne for de alignede sekvenser bliver forkerte.

Programmet tager som input en FASTA-fil og en identifiernavn. Programmet kan håndtere FASTA-filer downloaded fra Ensembl's ftp-site for DNA-sekvenser, eller som er formateret på samme måde. Eller også skal informationslinien være formateret på samme måde som output filens informationslinie er formateret `>identifiernavn:sekvensstart:sekvensslut:streng`. Programmet kan håndtere FASTA-filer med en eller flere FASTA-sekvenser.

Programmet producerer en outputfil, hvor filnavnet er bestemt af det identifiernavn, som programmet blev kaldt med, og får derfor navnet "identifiernavn.fa". Denne konstruktion gør det let senere at finde en delsekvensen udfra kendskab til identifiernavn, sekvensstart, sekvensslut og streng.

Informationslinierne får som sagt formatet:
`>identifiernavn:sekvensstart:sekvensslut:streng`

exonList.py

Programmet generer en fil med en liste af exon start- og slutpositioner, til brug for programmet `maskFASTAsequnce.py`. Oplysningerne om exons hentes fra Ensembls databasesystem `ensembldb.ensembl.org`, som tilgås via internettet.

Programmet tager databasenavn og kromosomnummer som input og evt. gentype (biotype i Ensembls terminologi) og evt. om det skal være known eller novel exons.

Ensembl har en lang række forskellige databaser. Hver database indeholder kun oplysninger om en enkelt art. Og for hver art er der flere versioner af databaser. Det betyder, at når databasenavn angives, så angives også hvilken art og database version, der ønskes oplysninger fra.

Gentype kan vælges til at være enten en bestemt gentype, alle gentyper, eller alle på nær miRNA gener. Hvis en exon liste med alle på nær miRNA exons ønskes, benyttes betegnelsen "not_mirna". Output filnavnet får navnet "databasenavn-kromosomnummer-exons.lst" Til at få overblik over hvilke gentyper, der findes oplysninger om i den database, som anvendes, er hjælpeprogrammet `geneInformation.py` blevet udviklet.

maskFASTAsequnce.py

Programmet maskerer en FASTA-fil, der er blevet reformateret med programmet `oneLineRename.py` eller har samme format, dvs. en identifier linie og sekvenslinien på en linie.

Input til programmet er filnavnet på den FASTA-fil, der skal maskeres, en fil med oplysninger om, hvor der skal maskeres i form af start- og slutpositioner og om maskeringen skal være "hard" eller "soft". Filen med maskeringsoplysningerne kan enten være genereret med `exonList.py` eller BioMart systemet. Programmet tager højde for, at der kan være tekst på første linie, som er tilfældet når BioMart benyttes.

Programmet foretager enten "hard mask" eller "soft mask". "Hard mask" er når hver base, der skal maskeres bliver konverteret til tegnet N. Ved "soft mask" bliver alle baser, der skal maskeres konverteret til små bogstaver. Programmet foretager en række test af maskeringspositionerne for at sikre, at de er gyldige.

splitFASTAsequnce.py

Programmet splitter en FASTA-sekvens i en fil op i mindre FASTA-sekvenser, og gemmer dem i en ny fil. Det er muligt at lave sekvenserne, så der er overlap mellem sekvenserne, for at forhindre de enkelte split får betydning for den senere analyse af sekvenserne.

Input til programmet er en inputfil, sekvenslængde og sekvensoverlap. Inputfilen skal være en outputfil fra `oneLineRename.py` eller den skal have samme format. Inputfilen må dog kun indeholde en FASTA-sekvens. Sekvenslængde er længden på sekvenserne i de nye FASTA-sekvenser. Sekvensoverlap angiver hvor stor overlap der skal være mellem de enkelte sekvenser. Sekvensoverlappet indgår som en del af længden på de nye sekvenser.

Ved at bruge identifier formatet `>identifiernavn:sekvensstart:sekvens-slut:streng` er det let at generer de nye identifier, for de opsplittede FASTA-sekvenser. Output filnavnet får navnet "split-inputfilnavnet".

De følgende programmer indgår i forbehandlings pipelinen for FASTA-filer med mere end en enkelt DNA-sekvens, se også figur 26.

oneLineRename.py

Det er den samme program, der anvendes i begge forbehandlingspipelines. Se beskrivelsen ovenfor.

scaffolds2database.py

Programmet placerer informationer om scaffolds i en database, disse informationer skal efterfølgende benyttes af andre programmer (se neden for) til

at udtrække scaffolds fra filen, der indeholder alle scaffolds. Informationerne der placeres i databasen er scaffoldnavn og -nummer, identifier- og sekvensposition samt sekvenslængde.

Informationerne om scaffolds placeres i databasetabellen `scaffolds`. MySQL-sætningen for at oprette scaffolds-tabellen findes i `createScaffoldsTable.sql` og beskrivelsen er vist i tabel 3.

`scaffolds` databasetabellen felter har følgende betydning:

`scaffoldID`	Unik identifikation
`scaffoldName`	Scaffold navn
`scaffoldNumber`	Scaffold nummer
`identifierPosition`	Identifierens position i filen
`sequencePosition`	Sekvensens position i filen
`sequenceLength`	Sekvensens længde

Programmet tager som input en fil med scaffolds, filen skal være en outputfil fra `oneLineRename.py`, eller den skal have samme format.

Field	Type	Null	Key	Default	Extra
scaffoldID	int(11)		PRI	NULL	auto_increment
scaffoldName	varchar(50)				
scaffoldNumber	int(10) unsigned		UNI	0	
identifierPosition	int(10) unsigned			0	
sequencePosition	int(10) unsigned			0	
sequenceLength	int(10) unsigned			0	

Tabel 3: MySQL beskrivelsen af databasetabellen `scaffolds`.

scaffoldsList.py

Programmet genererer en liste af tal, som er numrene på de scaffolds, der ønskes at arbejde videre med. Listen af tal benyttes efterfølgende af `scaffoldsFile.py` eller `splitScaffoldsFile.py` til at genererer en fil med de ønskede scaffolds.

Input til programmet er en outputfilnavn samt en betingelsessætning, som er en standart MySQL WHERE betingelse, der angiver hvilke scaffolds fra `scaffolds` tabellen,der ønskes. WHERE betingelsen skal være i citationstegn, for at programmet skal opfatte hele betingelsen som en parameter. I øvrigt skal betingelsen starte med ordet WHERE. Udelades betingelsen, fåes en liste med numrene på alle scaffolds.

miRNAscaffoldsList.py

Programmet gør noget af det samme som `scaffoldsList.py`, men i stedet for at trække data ud af `scaffolds` tabellen, laver den er forespørgsel til en

lokale installation af miRBase. Programmet genererer en liste med numrene på alle de fugu scaffolds, der har kendte miRNA gener ifølge miRBase. Pt. virker programmet kun til fugu scaffolds.

maskScaffolds.py

Programmet maskere en FASTA-fil bestående af scaffolds. Programmet gør næsten det samme som `exonList.py` og `maskFASTAsequnce.py` gør tilsammen. Grunden til at de to programmers samlede funktion er slået sammen i et program, når det handler om scaffolds, er det fordi, ligesom det er upraktisk at have alle scaffolds i filer for sig, er det upraktisk først at hente oplysningerne om exons positioner for alle scaffolds i filer for hver scaffold. Dette vil også skabe mange filer. Derfor er programmet konstrueret således, at den behandler de enkelte scaffolds en efter en. Først hentes oplysninger om en scaffolds position i `scaffolds` tabellen. Derefter hentes oplysningerne om exons placering i denne scaffold fra `ensembldb.ensembl.org` via internettet. Disse oplysninger bliver så brugt til at maskere, den aktuelle scaffold. Programmet kan både foretage "soft" og "hard" maskering, som beskrevet for `exonList.py`. Med programmet er det dog kun muligt at maskere en gentype af gangen, og det er ikke muligt at vælge om det skal være "known" eller "novel" exons.

Programmet tager som input filnavnet på den fil, der skal maskeres, navnet på den Ensembl database, som exon oplysningerne skal hente fra, samt maskerings måde og gentype.

scaffoldsFile.py

Programmet generer en fil med scaffolds ud fra en liste med scaffolds navne eller numre.

Inputparametrene til programmet er lookupfilnavn, listefilnavn og outputfilnavn. Lookupfilnavn er navnet på den fil, som blev benyttet som inputfil til `scaffolds2database.py` eller en maskeret version af filen. Listefilnavn indeholder listen med scaffold-numre eller -navne. Outputfilnavn er navnet på den genererede fil med de valgte scaffolds.

Skulle listen indeholde et nummer eller navn, som ikke findes i tabellen `scaffolds`, skriver programmet det på skærmen, og fortsætter med at finde de resterende scaffolds.

splitScaffoldsFile.py

Programmet gør det samme som `scaffoldsFile.py`, men derudover splitter den sekvenser, der er for lange til at kunne anvendes i en BLAT alignment som query sekvenser. Programmet benytter `splitFASTAsequence()` til at splitte sekenserne op i mindre sekvenser. Derfor har programmet også to ekstra input parametre sekvenslængde og overlap. Da `splitFASTAsequence()`

kun kan splitte en FASTA-fil med en DNA-sekvens, bliver hver sekvens, der skal splittes først skrives ud i en temporær fil, hvorefter opsplitningen kan foregå. Efterfølgende bliver indholdet af den temporære splitfil tilføjet den endelige outputfil.

6.3.2 Konserverede DNA-sekvenser

De følgende programmer indgår i konserverede DNA-sekvenser pipeline delen, se også figur 29.

axt2database.py

Dette program tager output-filen på axt-format fra første BLAT alignmentet og gemmer relevante informationer i tabellen axtInformation. MySQL-sætningen for at oprette axtInformation-tabellen findes i `createAxtInformationTable.sql` og beskrivelsen er vist i tabel 4. Det er kun informationer fra informationslinien, der gemmes og ikke sekvenserne. Start og slutpositioner for sekvenserne bliver korrigeret, hvis det er nødvendigt som beskrevet i pipeline afsnittet. Det er ikke nødvendigvis alle alignments, der skal i databasen. Det er muligt at sætte en minimum længde og minimum score for hver alignment.

`axtInformation` databsetabellens felter har følgende betydning:

`axtInformationID`	Unik identifikation
`alignmentNumber`	Nummeret på alignmentet i axt-filen
`databaseName`	Database-filens navn
`databaseStart`	Startposition i database-filens sekvens
`databaseEnd`	Slutposition i database-filens sekvens
`queryName`	Query-filens navn
`queryStart`	Startposition i query-filens sekvens
`queryEnd`	Slutposition i query-filens sekvens
`queryStrand`	query strengen
`BLATscore`	BLAT score i axt-filen

Som det kan ses er der ingen streng angivelse for database-sekvensen. Det skyldes, at database-filens sekvens altid vil være plus-sekvensen, når der anvendes sekvens-filer downloaded fra Ensembl. Start- og slutpositioner er sekvenspositioner for de alignede delsekvenser.

database2FASTA.py

Programmet genererer en FASTA-fil, udfra informationerne i `axtInformation` tabellen, med de DNA-sekvenser fra den første alignment, der skal indgå

Field	Type	Null	Key	Default	Extra
axtInformationID	int(11)		PRI	NULL	auto_increment
alignmentNumber	int(10) unsigned			0	
databaseName	varchar(50)			NULL	
databaseStart	int(10) unsigned			0	
databaseEnd	int(10) unsigned			0	
queryName	varchar(50)			NULL	
queryStart	int(10) unsigned			0	
queryEnd	int(10) unsigned			0	
queryStrand	char(2)			NULL	
BLATscore	int(11)			0	

Tabel 4: MySQL beskrivelsen af databasetabellen `axtInformation`.

i den anden BLAT alignment. Alle DNA-sekvenserne skal enten være fra database-filen eller query-filen fra den første alignment. Da `axtInformation` tabellen ikke indeholder sekvenserne, hentes de fra den relevante reformatertede FASTA-fil.

De genererede FASTA-sekvenser få i deres identifier tilføjet axtInformationID'et fra den post i tabellen, de kommer fra. Dette nummer bruges efter anden alignment til at finde informationerne for den art, der ikke indgår i den anden alinment, når informationerne for de tre arter samles efter anden alignment. Derfor har disse FASTA-sekvensers identifiers have følgende mulige formater, enten `>databaseName:databaseStart:databaseEnd:+:axtInformationID` eller `>queryName:queryStart:queryEnd:queryStrand:axtInformationID`.

triple2database.py

Dette program tager output-filen på axt-format fra den anden BLAT alignment og gemmer informationerne i `tripleInformation` tabellen. MySQL-sætningen for at oprette `tripleInformation` tabellen findes i `createTripleInformationTable.sql` og beskrivelsen er vist i tabel 5.

`tripleInformation` databasetabellens felter har følgende betydning:

`tripleInformationID`	Unik identifikation
`axtInformationID`	1. alignments unikke identifikation
`alignmentNumber`	Nummeret på alignmentet i axt-filen
`databaseName`	Database-filens navn
`databaseStart`	Startposition i database-filens sekvens
`databaseEnd`	Slutposition i database-filens sekvens
`Strand`	Database strengen
`queryName`	Query-filens navn
`queryStart`	Startposition i query-filens sekvens
`queryEnd`	Slutposition i query-filens sekvens

`queryStrand`	query strengen
`firstName`	Filnavnet for sekvensen fra 1. alignment
`firstStart`	Startposition for sekvensen fra 1. alignment, korrigeret
`firstEnd`	Slutposition for sekvensen fra 1. alignment, korrigeret
`firstStrand`	Strengen for sekvensen fra 1. alignment, korrigeret
`BLATscoreDQ`	BLAT score i axt-filen
`BLATscoreFQ`	BLAT score i forbindelse med korrektionen af sekvensen fra 1. alignment

Som for `axt2database.py` programmet er det også her muligt at sætte en minimum længde og minimum score for hver alignment. Der korrigerer også start- og slutpositioner på sekvenserne, hvis det er nødvendig. Ud over informationerne fra den anden alignment, skal informationerne for den sekvens, der var med i den første alignment men ikke med i den anden alignment også med i `tripleInformation` tabellen. Dette er beskrevet nærmere i pipeline afsnittet.

Field	Type	Null	Key	Default	Extra
tripleInformationID	int(11)		PRI	NULL	auto_increment
axtInformationID	int(10) unsigned			0	
alignmentNumber	int(10) unsigned			0	
databaseName	varchar(50)				
databaseStart	int(10) unsigned			0	
databaseEnd	int(10) unsigned			0	
databaseStrand	char(2)				
queryName	varchar(50)				
queryStart	int(10) unsigned			0	
queryEnd	int(10) unsigned			0	
queryStrand	char(2)				
firstName	varchar(50)				
firstStart	int(10) unsigned			0	
firstEnd	int(10) unsigned			0	
firstStrand	char(2)				
BLATscoreDQ	int(11)			0	
BLATscoreFQ	int(11)			0	

Tabel 5: MySQL beskrivelsen af databasetabellen `tripleInformation`.

6.3.3 Identifikation

Modsat de andre dele af pipelinen, består identifikationsdelen kun af et program. Derfor bliver de væsentligste funktioner vist i pipelinefiguren, se figur 29.

identification.py

Programmet finder de dele af de alignbare sekvenser for de tre valgte arter, som indeholder potentielle miRNA gener.

Input til programmet er oplysninger om i hvilken rækkefølge MLAGAN skal aligne sekvenserne, en standart MySQL WHERE betingelse, der angiver hvilke multiple alignments, der skal undersøges for potentielle miRNA gener, samt en sti til hvor programmet skal finde de relevante FASTA-filer, hvor DNA-sekvenserne skal hentes fra, da de ikke er gemt i `tripleInformation` tabellen.

Først trækkes informationer ud af `tripleInformation` tabellen om de tre sekvenser der skal alignes. For hver sekvens oprettes en FASTA-fil med sekvensen. De tre sekvenser alignes med multiplealignment programmet M-LAGAN. Outputfilen fra multiplealignmentet er på multi-FASTA format. Dette format kan ikke læses af hverken RNAz eller RNAmicro, derfor konverteres outputfilen til ClustalW format. Dernæst analyseres ClustalW format filen med programmet RNAz, der analyserer multiple alignmentet med en vindues størrelse på 100 nt. og i trin på 10 nt. Hvis RNAz finder et sted i multiple alignmentet, der kan klassificeres som potentiel ncRNA-sekvens, analyseres hele alignmentet efterfølgende med programmet RNAmicro, der anvender vindues størrelserne 75, 95 0g 115 nt. Oplysningerne om de dele af alignmentet, der bliver klassificeret som potentielle miRNA gener placeres derefter i `RNAmicroInformation` databasetabellen.

MySQL-sætningen for at oprette `RNAmicroInformation` tabellen findes i `createRNAmicroInformation.sql` og beskrivelsen er vist i tabel 6.

`RNAmicroInformation` databasetabellens felter har følgende betydning:

`RNAmicroInformationID`	Unik identifikation
`tripleInformationID`	tripleInformations identifikation
`speciesOneName`	1. arts navn
`speciesOneStart`	Startposition i 1. arts sekvens
`speciesOneEnd`	Slutposition i 1. arts sekvens
`speciesOneStrand`	Strengen for 1. arts sekvens
`speciesTwoName`	2. arts navn
`speciesTwoStart`	Startposition i 2. arts sekvens
`speciesTwoEnd`	Slutposition i 2. arts sekvens
`speciesTwoStrand`	Strengen for 2. arts sekvens
`speciesThreeName`	3. arts navn
`speciesThreeStart`	Startposition i 3. arts sekvens
`speciesThreeEnd`	Slutposition i 3. arts sekvens
`speciesThreeStrand`	Strengen for 3. arts sekvens
`RNAmicroScore`	RNAmicro score

Programmet skriver også en logfil. Alle post-id skrives til logfilen en på hver linie. På en linie skrives også hvis multiple alignmentet resulterede i ingen alignment, om RNAz klassificerede alignmentet som RNA eller om RNAz kom med en exception.

Field	Type	Null	Key	Default	Extra
RNAmicroInformationID	int(11)		PRI	NULL	auto_increment
tripleInformationID	int(10) unsigned			0	
speciesOneName	varchar(50)			NULL	
speciesOneStart	int(10) unsigned			0	
speciesOneEnd	int(10) unsigned			0	
speciesOneStrand	char(2)			NULL	
speciesTwoName	varchar(50)			NULL	
speciesTwoStart	int(10) unsigned			0	
speciesTwoEnd	int(10) unsigned			0	
speciesTwoStrand	char(2)			NULL	
speciesThreeName	varchar(50)			NULL	
speciesThreeStart	int(10) unsigned			0	
speciesThreeEnd	int(10) unsigned			0	
speciesThreeStrand	char(2)			NULL	
RNAmicroScore	float unsigned			0	

Tabel 6: MySQL beskrivelsen af databasetabellen `RNAmicroInformation`.

6.4 Hjælpeprogrammer

splitTestFASTAsequence.py

Programmet generer test FASTA-filer til splitFASTAsequnce.py, så det er betydelig lettere at se om splitFASTAsequnce.py splitter FASTA-sekvenserne korrekt.

geneInformation.py

Programmet generer en liste over de forskellige gentyper, samt deres antal fordelt på "known" eller "novel" gener, der findes på et kromosom. Oplysningerne om gentyper hentes fra Ensembls databasesystem `ensembldb.ensembl.org`, som tilgås via internettet. Ved at anvende programmet er det muligt at se, hvordan de forskellige gentyper staves.

Input til programmet er databasenavn, som også angiver art, og kromosomnavn.

7 Test

For at teste pipelinen, er det nødvendig at finde relevant testmateriale, således at det er muligt at vurdere om pipelinen virker, som det var tiltænkt.

Først beskrives hvordan testmaterialet er blevet valgt, og de programmer der er blevet udviklet til dette formål. Efterfølgende beskrives selve testen af pipelinen og resultatet heraf, samt de programmer, der er blevet udviklet til at analysere resultatet.

7.1 Valg af testmateriale

Ved at se på, hvor mange af allerede kendte miRNAer, der kan findes ved at anvende pipelinen, kan det vurderes, hvor god den er til at finde mulige microRNA gener.

Oplysningerne om allerede kendte miRNAer findes i miRBase::Sequences (`http://microrna.sanger.ac.uk/sequences/`) (Griffiths-Jones et al., 2006, 2008) tidligere kaldet The miRNA Registry (Griffiths-Jones, 2004). Det er oplysninger fra release 9.0 oktober 2006, der er benyttet i dette projekt. Databasen indeholder oplysninger om kendte miRNA for både dyr (32 arter), planter (9 arter) og viruser (8 arter) ialt 4361 miRNAer.

I figur 30 er vist antallet af miRNAer for de 24 arter af hvirveldyr (Vertebrata), der er med i miRBase release 9.0. Kun hvirveldyr er medtaget her, fordi valget af testmateriale er rettet mod hvirveldyrarterne. Da disse arter er de mest interessante, og så vil det være muligt at indrage mennesket i testen.

7.1.1 Valg af arter

Når der skal vælges arter til komparativ genom analyse, til at finde gener, så er det ikke ligegyldigt, hvilke arter der anvendes. Er arterne for nært beslægtede, vil for meget af deres DNA-sekvenser kunne alignes med hinanden, så det der søges efter vil forblive skjult. Er arterne for fjernt beslægtede, er DNA-sekvenserne ændret så meget at homologe sekvenser ikke længere kan identificeres. Det er derfor nødvendigt at finde arter, der har en passende evolutionær afstand, der gør det muligt at finde det der søges efter (Boffelli et al., 2004).

Ud fra figur 30 kan det ses at mennesket er den art hos hvirveldyrene med flest kendte miRNAer. Denne art vil derfor være et naturligt valg som en af de tre arter. Dernæst vil musen være et fornuftigt valg, da det er den art med næstflest kendte miRNAer, og de to arter er ofte blevet anvendt i komparative genomanalyser. Den evolutionære afstand mellem menneske og mus er 91 millioner år (Ureta-Vidal et al., 2003).

Oprindelig var det tænkt at bruge fugu som den tredje art, men det var før at der var oplysninger om fugus miRNAer i miRBase. De fugu miRNAer, der

```
Metazoa
  +-- Vertebrata
        +-- Amphibia
        |     +--  Xenopus laevis                    (7)
        |     +--  Xenopus tropicalis               (177)
        |
        +-- Aves
        |     +--  Gallus gallus                    (152)
        |
        +-- Mammalia
        |     +-- Carnivora
        |     |     +--  Canis familiaris             (6)
        |     |
        |     +-- Metatheria
        |     |     +--  Monodelphis domestica      (107)
        |     |
        |     +-- Primates
        |     |     +-- Atelidae
        |     |     |     +--  Ateles geoffroyi      (45)
        |     |     |     +--  Lagothrix lagotricha  (48)
        |     |     |
        |     |     +-- Cebidae
        |     |     |     +--  Saguinus labiatus     (42)
        |     |     |
        |     |     +-- Cercopithecidae
        |     |     |     +--  Macaca mulatta        (71)
        |     |     |     +--  Macaca nemestrina     (75)
        |     |     |
        |     |     +-- Hominidae
        |     |     |     +--  Gorilla gorilla       (86)
        |     |     |     +--  Homo sapiens         (474)
        |     |     |     +--  Pan paniscus          (89)
        |     |     |     +--  Pan troglodytes       (83)
        |     |     |     +--  Pongo pygmaeus        (84)
        |     |     |
        |     |     +-- Lemuridae
        |     |           +--  Lemur catta           (16)
        |     |
        |     +-- Rodentia
        |     |     +--  Mus musculus               (373)
        |     |     +--  Rattus norvegicus          (234)
        |     |
        |     +-- Ruminantia
        |     |     +--  Bos taurus                  (98)
        |     |     +--  Ovis aries                   (4)
        |     |
        |     +-- Suina
        |           +--  Sus scrofa                  (54)
        |
        +-- Pisces
              +--  Danio rerio                     (337)
              +--  Fugu rubripes                   (131)
              +--  Tetraodon nigroviridis          (132)
```

Figur 30: Antal miRNAer hos hvirveldyr (Vertebrata) arterne i miRBase release 9.0.

siden er kommet i miRBase er fundet vha. homologi søgning fra verificerede miRNAer fra zebrafisk, det vil derfor være mere relevant at bruge denne art,

som den tredje art i testen. Et problem med fugu og zebrafisk er at de måske er evolutionnært for langt fra menneske og mus, til at kunne benyttes i en BLAT alignment. Zebrafisks evolutionære afstand til menneske og mus er 450 millioner år (Ureta-Vidal et al., 2003). Et alternativ er derfor rotten, hvor der også er fundet ret mange miRNAer. Dens evolutionære afstand til musen er 41 millioner år og til mennesket 91 millioner år ligesom musen (Ureta-Vidal et al., 2003).

Ved at foretage en BLAT-alignment af alle pre-miRNAer fra menneske og zebrafisk og fra menneske og rotte, er det muligt at få en ide, om det er muligt, at benytte zebrafisk, som den tredje art. Resultatet af to sådanne alignments er vist i figur 31 og 32, der viser længderne på alignments, og længderne af zebrafisks og rottens pre-miRNAer er også vist.

Til udarbejdelsen af data til figur 31 og 32 blev programmerne `premiRNA-sequences.py`, `premiRNAlength4Gnuplot.py` og `axtLinelenght4Gnuplot.py` udviklet. Selve figurene er genereret med programmet Gnuplot. Programmerne der er blevet udviklet i forbindelse med valg af testmateriale er beskerevet nærmere i afsnittet "Programmer til valg af testmateriale" på side 82.

Ud fra figur 31 og 32 kan det ses, at BLAT alignment længderne ligger i to grupper, en med længder på ca. 20-40 baser og en anden med længder ca. 60-120 baser.

Ud fra figur 31 ses det tydeligt, at pre-miRNAernes længder hos zebrafisk ligger i området ca. 75-135 nt. Hvorimod alignment længderne ligger i to grupper en med længder på ca. 20-40 nt. og en anden med længder på ca. 55-90 nt.

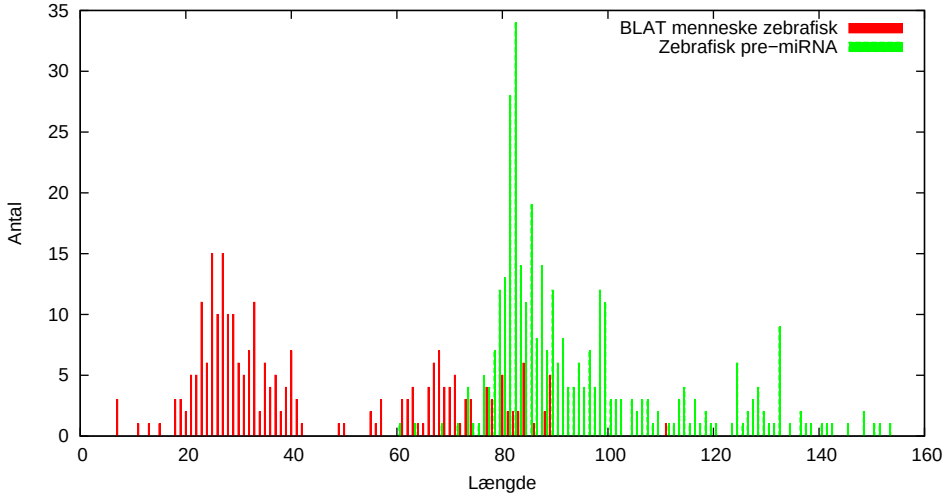
Grunden til, at alignment længderne danner de to grupper er, at nogle af de dannede alignments er alignments af hele pre-miRNAer (gruppen 55-90 nt.), og andre der kun består af alignments af de dele af pre-miRNAerne, som udgør stem regionerne (gruppen 20-40 nt.), hvorimod de dele der udgør loop regionen ikke bliver alignet.

Ud fra figur 32 kan det ses et tilsvarende resultat for rotten pre-miRNAerne, der har længder på ca. 60-115 nt. Alignment længderne deles også i to grupper, dog forekommer der i gruppen med de lange alignments længere alignments end hos zebrafisken, op til ca. 115 nt.

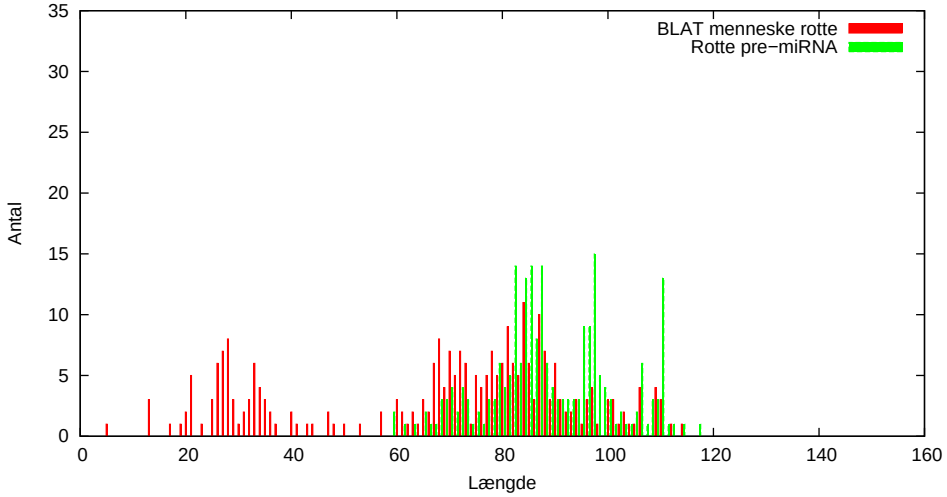
Der er dog også en tydelig forskel mellem de to sæt alignments. Hos zebrafisken er flertallet af alignments af den korte slags, mens det hos rotten er de lange alignments der dominerer.

En anden ting er, at selvom der kendes ca. 100 flere miRNAer hos zebrafisk end hos rotten, så giver de to sæt alignments næsten det samme antal alignments. Der er derfor mange af zebrafiskens pre-miRNAer der ikke alignes til menneskets pre-miRNAer.

Det skal iøvrigt bemærkes, at hvis det kun er stem regionerne, der alignes mellem to pre-miRNAer, så kan det resulterer i to alignments, nogle gange er det kun den ene stemregion, der bliver alignet. Hvis det er hele pre-miRNAerne, der alignes, så giver det kun en alignment. Det er ikke nærmere



Figur 31: Sammenligning af BLAT aligment længder, fra alignment af pre-miRNAer fra menneske og zebrafisk (254 alignments) og pre-miRNA længderne for zebrafisk (337 pre-miRNA).



Figur 32: Sammenligning af BLAT aligment længder, fra alignments af pre-miRNAer fra menneske og rotte (274 alignments) og pre-miRNA længderne for rotte (234 pre-miRNA).

undersøgt, hvor mange pre-miRNA alignments der resulterer i to alignments.

Resultatet af disse alignments må blive, at hvis der skal være mulighed for mange alignments, så vil det være mest fornuftigt at vælge rotten som

tredje art. Zebrafisken er evolutionært for langt fra menneske og mus, når BLAT anvendes som alignment program.

7.1.2 Valg af kromosomer

Det er formodentlig ikke nødvendig at anvende hele genomet for de valgte arter i første omgang, men nøjes med et kromosom fra hver art, evt. kun et udsnit af et kromosom fra hver art. Det er dog ikke uden betydning hvilke kromosomer der vælges.

Der er mindst to ting der skal tages højde for, når der vælges kromosomer:

1 Antallet af allerede kende miRNA gener på et kromosom. Der skal gerne være mange, så der er noget at lede efter, hvis det skal kunne findes.

2 Beslægtethed mellem miRNA generne på kromosomerne der sammenlignes. Mange af miRNA generne skal være beslægtede (være homologe), for ellers vil de ikke kunne findes ved komparativ genomanalyse.

En tredje ting der kan tages med i valg af kromosom:

3 Længden af kromosomerne. Hvis to kromosomer har ca. samme antal miRNA gener, kan det være en fordel at vælge den korteste kromosom, for så vil mængden af data, der skal behandles ikke være så stort. Og hvis det er muligt at finde delområder på kromosomerne med mange beslægtede miRNA gener, kan datamængden måske reduceres endnu mere.

I tabel 7, 8 og 9 er vist antallet af miRNAer pr. kromosom for menneske, mus og rotte, desuden er tætheden af miRNAer pr. 1.000.000 basepar vist. Programmerne `chromosomeLength.py` og `miRNAdensity.py` blev udviklet for at kunne udarbejde tabel 7, 8 og 9.

Det kan ses, at antallet af miRNAer pr. kromosom varierer meget fra 0 til 66 hos mennesket. Det højeste antal hos musen er 50, og hos rotten er det højeste antal 35.

De mest interessante kromosomer hos mennesket, når der ses på antal miRNAer er kromosom 14, 19 og X med henholdsvis 49, 66 og 51 miRNAer. Hvis tætheden også tages i betragtning bliver kromosom 17 også interessant, den har 27 miRNAer, og tætheder er højere end for kromosom X.

For musens vedkommende er det kromosom 2, 12 og X, som er mest interessante, når der ses på antallet af miRNAer, de har henholdsvis 45, 50 og 43 miRNAer. Kromosom 11 bliver også interessant, når tætheden tages i betragtning, den har 27 miRNAer. Selvom mitokondrie kromosomet MT har den højeste tæthed er den ikke særlig interessant, da den kun har et miRNA. Det er den korte længde af kromosomet, der giver den høje tæthed.

Kromosom	Antal miRNAer	Kromosom længde	miRNA tæthed
1	32	247.249.719	0.129
2	13	242.951.149	0.054
3	20	199.501.827	0.100
4	15	191.273.063	0.078
5	17	180.857.866	0.094
6	11	170.899.992	0.064
7	27	158.821.424	0.170
8	15	146.274.826	0.103
9	22	140.273.252	0.157
10	13	135.374.737	0.096
11	16	134.452.384	0.119
12	18	132.349.534	0.136
13	11	114.142.980	0.096
14	49	106.368.585	0.461
15	13	100.338.915	0.130
16	8	88.827.254	0.090
17	27	78.774.742	0.343
18	4	76.117.153	0.053
19	66	63.811.651	1.034
20	11	62.435.964	0.176
21	5	46.944.323	0.107
22	10	49.691.432	0.201
X	51	154.913.754	0.329
Y	0	57.772.954	-
MT	0	16.571	-

Tabel 7: Antal miRNAer, kromosom længde i basepar og miRNA tæthed pr. 1.000.000 basepar for hvert kromosom hos hos mennesket.

Hos rotten er det kromosom 1, 6, 10 og 22 med henholdsvis 22, 35, 21 og 22 miRNAer, der er de mest interessante, når der ses på antallet af miRNAer. Tages tætheden også i betragtning er kromosom 1 måske ikke så interessant, da det er et meget langt kromosom.

Det er ikke nok at se på antal miRNAer og deres tæthed for at udvælge det mest egnet kromosom fra hver art til testmateriale. Det er nødvendigt at miRNAerne fra de valgte kromosomer er beslægtede, for at de skal kunne findes ved at aligne kromosomernes DNA-sekvenser.

I miRBase er slægtskabet mellem de forskellige miRNAer angivet ved, at de er grupperet i familier. Ved at sammenligne antallet af fælles familier mellem de valgte arters kromosomer, vil det være muligt at finde de kro-

Kromosom	Antal miRNAer	Kromosom længde	miRNA tæthed
1	21	197.069.962	0.107
2	45	181.976.762	0.247
3	14	159.872.112	0.088
4	20	155.029.701	0.129
5	11	152.003.063	0.072
6	19	149.525.685	0.127
7	16	145.134.094	0.110
8	11	132.085.098	0.083
9	16	124.000.669	0.129
10	9	129.959.148	0.069
11	27	121.798.632	0.222
12	50	120.463.159	0.415
13	15	120.614.378	0.124
14	17	123.978.870	0.137
15	11	103.492.577	0.106
16	14	98.252.459	0.142
17	8	95.177.420	0.084
18	8	90.736.837	0.088
19	6	61.321.190	0.098
X	43	165.556.469	0.260
Y	0	16.029.404	-
MT	1	16.299	61.353

Tabel 8: Antal miRNAer, kromosom længde i basepar og miRNA tæthed pr. 1.000.000 basepar for hvert kromosom hos hos musen.

mosomer med flest fælles familier. Disse kromosomer vil formodentlig være blandt de best egnede til at teste pipelinen.

Tabel 10 viser antallet af fælles miRNA familier kromosomvis og parvis mellem de tre arter menneske, mus og rotten. For at kunne udarbejde tabellen blev programmerne `countCommonmiRNAfamilies.py` og `topCommonmiRNA-familiesCount.py`.

Det ses tydeligt, at der er tre sæt af kromosomer, der har mange fælles miRNA familier. Samlet antal miRNAer og samlede længden af af kromosomerne for de tre sæt af kromosomer er vis i tabel 11. Ud fra dette ser det ud til at kromosomsættet hsa 14, mmu 12 og rno 6 er det mest interessante at foretage testen af pipelinen på, især fordi det er det sæt med flest miRNAer.

En anden ting der gør kromosomsættet hsa 14, mmu 12 og rno 6 interessant er, at de fleste miRNAer, der findes på disse kromosomer, findes inden for et begrænset område af kromosomerne. Det gør det muligt at ud-

Kromosom	Antal miRNAer	Kromosom længde	miRNA tæthed
1	22	267.910.886	0.082
2	8	258.207.540	0.031
3	12	171.063.335	0.070
4	9	187.126.005	0.048
5	10	173.096.209	0.058
6	35	147.636.619	0.237
7	12	143.002.779	0.084
8	11	129.041.809	0.085
9	5	113.440.463	0.044
10	21	110.718.848	0.190
11	6	87.759.784	0.068
12	4	46.782.294	0.086
13	13	111.154.910	0.117
14	3	112.194.335	0.027
15	9	109.758.846	0.082
16	2	90.238.779	0.022
17	7	97.296.363	0.072
18	9	87.265.094	0.103
19	7	59.218.465	0.118
20	1	55.268.282	0.018
X	22	160.699.376	0.137
MT	0	16.300	-

Tabel 9: Antal miRNAer, kromosom længde i basepar og miRNA tæthed pr. 1.000.000 basepar for hvert kromosom hos hos rotten.

vælge disse områder, så datasættet bliver meget mindre end, hvis det var hele kromosomer, der blev benyttet i testen. Samtidig bevares de fleste af miRNAerne i datasættet, til at vurdere pipelinens effektivitet.

For hsa 14 ligger 45 miRNAer ud af 49 inden for ca. 1 mbp., for mmu 12 ligger 46 miRNAer ud af 50 inden for ca. 1 mbp., og for rno 6 ligger 31 ud af 35 inden for ca. 250.000 bp.

Mere detaljerede informationer om miRNAerne vedrørende de tre kromosomer findes på specialets hjemmeside. Bl.a. miRNA genernes placering på kromosomerne og deres indbyrdes afstand, deres placering i forhold til andre gener – mellem gener, mellem exons eller i 3'UTR områder.

Kromosom hsa	mmu	Antal	Kromosom hsa	rno	Antal	Kromosom mmu	rno	Antal
X	X	23	X	X	19	12	6	21
14	12	19	14	6	18	X	X	19
17	11	17	17	10	15	11	10	18
7	6	11	11	1	9	2	3	11
1	1	10	1	13	8	1	13	10
3	9	7	19	1	7	7	1	10
20	2	6	7	4	6	9	8	10
19	13	5	19	17	5	6	4	7
1	4	5	12	7	5	8	19	7
21	16	5	3	8	5	4	5	7

Tabel 10: Antal fælles miRNA familier mellem mennesket (hsa), musen (mmu) og rotten (rno) parvis og kromosomvis. Kun de 10 med flest fælles familer er medtaget.

Kromosom hsa	mmu	rno	Antal miRNAer	Samlet antal baser
X	X	X	116	481.169.599
14	12	6	134	374.468.363
17	11	10	75	311.292.222

Tabel 11: Samlet antal miRNA og længde på kromosomerne for de tre sæt kromosomer med flest miRNA familier for mennesket (hsa), musen (mmu) og rotten (rno).

7.1.3 Programmer til valg af testmateriale

I det følgende beskrives de programmer, der er blevet udviklet i forbindelse med valg af testmateriale. Programmerne der anvendes i forbindelse med en lokal installation af miRBase databasen, er udviklet til at anvende miRBase release 9.0, men skulle formodentlig kunne anvendes op til release 11.0, men det er kun testet i begrænset omfang.

createmiRNAdb90.sh

Efter at have downloaded database filerne, der skal anvendes til en lokal installation af miRBase databasen, benyttes `createmiRNAdb90.sh` til at pakke filerne ud, oprette de nødvendige databasetabeller, og til sidst placere miRBase data i de respektive tabeller. Kan ikke anvendes fuld ud til de nyeste releases af miRBase, da der er kommet et par ekstra tabeller med i.

premiRNAsequences.py

Trækker pre-miRNA-sekenser ud af en lokal miRBase database, og gemmer dem i en FASTA-fil. Der kan dannes en FASTA-fil indeholdene alle pre-miRNAer for en art eller for en enkelt kromosom. Identifieren for hver sekvens er formatteret på samme måde som de reformaterede FASTA-filer i pipelinen. Sekvenserne i miRBase databasen er RNA-sekvenser, programmet kan konvertere dem til DNA-sekvenser. Input til programmet er art, der angives ved den artskode, som miRBase identificere en art med. F.eks. menneske "hsa", evt. en kromosomnummer, og om sekvenserne skal konverteres til DNA-sekvenser.

premiRNAlength4Gnuplot.py

Tæller antal pre-miRNAer, der har samme længde. Resultatet placeres i en fil, hvor der på hver linie indeholder længde og antal sorteret efter længde. Oplysningerne hentes fra lokal miRBase installation. Optællingen foretages i MySQL forespørgslen. Input til programmet er miRBase artskode.

axtLinelenght4Gnuplot.py

Tæller antal alignments, der har en bestemt længde i en axt formateret fil. Til optællingen anvendes en Python dictionary. Dette betyder, at der ikke er en maksimum længde, der skal tages højde for. Input til programmet er axt-filens navn.

chromosomeLength.py

Henter oplysninger om kromosom længderne for en art fra en Ensembl database, og placerer dem i en lokal databasetabel `chromosomeLength`, se tabel 12, der oprettes, hvis den ikke allerede findes. Databasetabellen placeres sammen med den lokalt installerede miRBase, da informationerne i tabellen skal anvendes sammen med oplysninger fra miRBase databasen. Input til programmet er Ensembl databasenavn og miRBase artskoden for arten.

Field	Type	Null	Key	Default	Extra
chromosomeLengthID	int(10) unsigned		PRI	NULL	auto_increment
species	varchar(5)				
chromosome	varchar(5)				
length	int(10) unsigned			0	

Tabel 12: MySQL beskrivelsen af databasetabellen `chromosomeLength`.

miRNAdensity.py

Beregner tætheden af miRNA gener på en arts kromosomer. Programmet henter oplysningerne fra en lokal miRBase installation samt databasetabellen med kromosomlængder, derfor skal artens kromosomlængder være i tabellen for at programmet kan køre. Alle beregninger foregår i MySQL forespørgslen. Resultatet skrives til skærmen, og kan være sorteret efter kromosomnavn, antal miRNA gener, kromosomlængde eller tæthed i stigende eller faldende orden. Input til programmet er miRBase artskode for arten, og oplysninger om den ønskede sortering.

countCommonmiRNAfamilies.py

Tæller hvor mange fælles miRNA familier to arter har, optællingen er for hver kromosompar, bestående af en kromosom for hver art. Resultatet placeres i en databasetabel, se tabel 13, der først oprettes, og er navngivet efter de to arters miRBase kode. Der oprettes en databasetabel for hver artspar, der analyseres. Input til programmet er de to arters miRBase artskode.

```
+----------+------------------+------+-----+---------+----------------+
| Field    | Type             | Null | Key | Default | Extra          |
+----------+------------------+------+-----+---------+----------------+
| ID       | int(10) unsigned |      | PRI | NULL    | auto_increment |
| hsa      | varchar(20)      |      |     |         |                |
| mmu      | varchar(20)      |      |     |         |                |
| families | int(10) unsigned |      |     | 0       |                |
+----------+------------------+------+-----+---------+----------------+
```

Tabel 13: MySQL beskrivelsen af databasetabelstrukturen som `countCommonmiRNAfamilies.py` bruger. Eksemplet er fra menneske "hsa" og mus "mmu". Felterne hsa og mmu er til kromosomnavnene.

topCommonmiRNAfamiliesCount.py

Trækker alle oplysningerne ud af databasetabeller oprettet med programmet `countCommonmiRNAfamilies.py`. Resultatet skrives til skærmen, med de kromosompar med flest fælles familier først. Input til programmet er de to arters miRBase artskoder og en evt. en begrænsning på, hvor mange poster, der skal skrives ud.

countCommonmiRNAfamilies1Chromosome.py

Gør noget af det samme som `topCommonmiRNAfamiliesCount.py`. Men resultatet begrænses til kun at gælde en kromosom i en art og alle kromosomer i den anden art. Derved er det muligt at se, hvordan miRNA familierne på en kromosom i en art er spredt ud over kromosomerne i den anden art. Input til programmet er miRBase artskode og kromosomnavnet for den ene art og kun miRBase artskode for den anden art.

miRNAcount.py

Tæller antal miRNA gener pr. kromosom for en art. Oplysninger hentes fra en lokal miRBase database. Resultatet skrives ud på skærmen, og kan sorteres efter kromosomnavn eller miRNA gen antal i stigende eller faldende orden. Input til programmet er miRBase artskoden, samt hvordan der skal sorteres.

De følgende tre programmer trækker informationer om miRNAer for en art og en kromosom ud af en lokal miRBase installation. De skriver alle resultatet ud på skærmen. Input til programmerne er miRBase artskode og kromosomnavn, hvis ikke andet er angivet.

miRNAinformation.py

Finder følgende oplysninger om hver miRNA: Navn, familie, start- og slut position for pre-miRNA, streng samt pre-miRNAlængde.

miRNAdistance.py

Beregner afstanden mellem miRNA generne på den samme DNA-streng. I input til programmet, skal der også angives hvilken streng afstandene skal beregnes. Alle beregninger foretages i MySQL forespørgslen.

miRNAoverlap.py

Finder oplysninger om hvorvidt miRNA generne overlapper med andre typer gener eller er placeret mellem andre gener. Hvis et miRNA gen overlapper med et andet gen, oplyses det om de ligger på samme streng, hvilken del af det andet gen det ligger på (exon, intron eller 3'UTR) samt navnet på det andet gen.

7.2 Test af pipelinen

Til test af pipelinien er valgt et område på 5 millioner basepar (mbp.) for hver af de tre kromosomer hsa 14, mmu 12 og rno 6, således at flest mulige miRNA gener kommer med i datasættet. De DNA sekvensområder som testmaterialet dækker er vist i tabel 14, desuden er der vist hvor stor del af kromosomernes miRNA gener, der ligger i området.

DNA-sekvenserne der anvendes i testen er downloaded fra Ensembl release 40 August 2006. Fordi det er den samme Ensembl release, der ligger til grund for oplysningerne i miRBase release 9.0. Oplysningerne om kromosom længder og exons er hentet fra samme Ensemble release. Ved at bruge samme Ensembl release sikres det at gen positionerne er de samme, uanset

Art	Kromosom	Område	Antal miRNAer
Menneske	hsa 14	99.000.000-104.000.000	46/49
Mus	mmu 12	109.000.000-114.000.000	47/50
Rotte	rno 6	132.000.000 137.000.000	34/35

Tabel 14: DNA-sekvens områderne for testmaterialet, samt antal miRNA gener i området ud af alle miRNA gener på det enkelte kromosom.

om der refereres til miRBase, en Ensembl database eller downloadede DNA-sekvenser. De Ensembl databaser, der er anvendt til oplysninger om exons og kromosomlængder, er vist i tabel 15

Art dansk	Art latin	Assembly	Ensembl database
Menneske	Homo sapiens	NCBI36	homo_sapiens_core_40_36b
Mus	Mus musculus	NCBIM36	mus_musculus_core_40_36a
Rotte	Rattus norvegicus	RGSC3.4	rattus_norvegicus_core_40_34j

Tabel 15: Database og assembly oplysninger om de valgte arter menneske, mus og rotte.

En samlet oversigt af miRNAerne fra de tre kromosomer er vist i tabel 16 sorteret familievis. Hver række indeholder de homologe gener for de tre arter. Dette er dog afveget to steder, hvor der hos mennesket findes to homologe gener til et gen i henholdsvis mus og rotte. Det er de steder, hvor gen navnene ender på "-1" eller "-2" hos mennesket. Alle miRNA generne på de tre kromosomer ligger på plus-strengen på nær fire hsa-mir-208, hsa-mir-624, mmu-mir-680-3 og mmu-mir-681. Alle miRNA gener der indgår i testen ligger på plus-strengen.
Det er ikke alle miRNA gener på de tre valgte kromosomer, der kan findes gennem pipelinen:

- De der ligger udenfor testmaterialets område
- De der er blevet maskeret væk
- De der ikke har homologe gener i alle tre arter i testsekvenserne

De miRNA gener, der ligger udenfor testmateriale området er markeret med "u" i tabel 16. Det drejer sig om ialt syv miRNA gener. For mennesket og musen er det 3 miRNA gener hver og en enkelt for rottens vedkommende. De tre hos mennesket og to af dem hos musen forekommer kun hos den enkelte art. De to sidste er homologe, men findes kun hos mus og rotte.

Tabel 16: Oversigt over miRNA gener hos hsa 14, mmu 12 og rno 6. Med angivelse af resultaterne for test af BLAT, RNAz og RNAmicro. Se teksten for forklaring på bogstav symbolernes betydning.

							BLAT-test				
Familie	hsa-miRNA	Ekskl.	mmu-miRNA	Ekskl.	rno-miRNA	Ekskl.	h:m	m:r	r:h	RNAz	RNAmicro
mir-127	hsa-mir-127	x	mmu-mir-127	x	rno-mir-127	x	p	p	p	1,00	1,00
mir-134	hsa-mir-134		mmu-mir-134		rno-mir-134		p	p	p	1,00	0,47
mir-136	hsa-mir-136		mmu-mir-136	x	rno-mir-136	x	p	p	p	0,93	1,00
mir-153			mmu-mir-153	u	rno-mir-153	u		p		0,93	1,00
mir-154	hsa-mir-154		mmu-mir-154		rno-mir-154		p	p	p	0,90	1,00
			mmu-mir-300		rno-mir-300			p		0,96	0,96
	hsa-mir-323		mmu-mir-323		rno-mir-323		p	p	p	0,99	0,93
	hsa-mir-369		mmu-mir-369		rno-mir-369		p	p	p	0,88	1,00
	hsa-mir-377		mmu-mir-377		rno-mir-377		p	p	p	0,94	0,77
	hsa-mir-381		mmu-mir-381		rno-mir-381		p	p	p	0,90	1,00
	hsa-mir-382		mmu-mir-382		rno-mir-382		p	p	p	0,96	1,00
	hsa-mir-409		mmu-mir-409		rno-mir-409		p	p	p	0,97	1,00
	hsa-mir-410		mmu-mir-410				p			0,94	1,00
	hsa-mir-487a									0,98	1,00
	hsa-mir-487b	r5	mmu-mir-487b	r4	rno-mir-487b	r4	p	p	p	0,94	1,00
	hsa-mir-494		mmu-mir-494		rno-mir-494		p	p	p	0,90	1,00
	hsa-mir-496		mmu-mir-496				p			0,92	1,00
	hsa-mir-539		mmu-mir-539		rno-mir-539		2s	p	2s	0,92	1,00

Fortsættes på næste side.

Tabel 16: *Fortsat fra foregående side.*

							BLAT-test				
Familie	hsa-miRNA	Ekskl.	mmu-miRNA	Ekskl.	rno-miRNA	Ekskl.	h:m	m:r	r:h	RNAz	RNAmicro
	hsa-mir-655									1,00	1,00
	hsa-mir-656									1,00	1,00
mir-203	hsa-mir-203		mmu-mir-203		rno-mir-203		p	p	p	1,00	1,00
mir-208	hsa-mir-208	u									
mir-299	hsa-mir-299		mmu-mir-299		rno-mir-299		p	p	p	0,56	0,11
mir-329			mmu-mir-329		rno-mir-329		-	p	-	0,98	0,99
	hsa-mir-329-1						-		-	0,98	1,00
	hsa-mir-329-2						-		-	0,98	1,00
	hsa-mir-495		mmu-mir-495				2s		f/−	1,00	1,00
			mmu-mir-543		rno-mir-543			p	f/−	0,98	1,00
mir-337	hsa-mir-337		mmu-mir-337		rno-mir-337		p	p	−/p	0,93	1,00
mir-341			mmu-mir-341	r76	rno-mir-341	r77		p		1,00	1,00
mir-342	hsa-mir-342		mmu-mir-342		rno-mir-342		p	p	p	0,91	1,00
mir-345	hsa-mir-345	r40	mmu-mir-345		rno-mir-345		−	p	−	0,96	1,00
mir-368	hsa-mir-368						f		f	0,96	1,00
			mmu-mir-376a		rno-mir-376a		f	p	f	1,00	1,00
	hsa-mir-376a-1						f		f	1,00	1,00
	hsa-mir-376a-2						f		f	1,00	1,00
	hsa-mir-376b		mmu-mir-376b		rno-mir-376b		f	p	p/f	0,89	1,00
			mmu-mir-376c		rno-mir-376c		f	p	f	0,99	1,00
mir-370	hsa-mir-370	r100	mmu-mir-370	r100	rno-mir-370	r100	p	p	p	0,95	0,99

Fortsættes på næste side.

Tabel 16: *Fortsat fra foregående side.*

							BLAT-test				
Familie	hsa-miRNA	Ekskl.	mmu-miRNA	Ekskl.	rno-miRNA	Ekskl.	h:m	m:r	r:h	RNAz	RNAmicro
mir-379	hsa-mir-379		mmu-mir-379		rno-mir-379		p	p	p	0,99	0,98
	hsa-mir-380		mmu-mir-380				p			0,99	0,01
	hsa-mir-411		mmu-mir-411				p			1,00	1,00
	hsa-mir-758		mmu-mir-758				p			1,00	1,00
mir-412	hsa-mir-412		mmu-mir-412		rno-mir-412		p	p	p	1,00	0,53
mir-431	hsa-mir-431	x	mmu-mir-431	x	rno-mir-431	x	p	p	p	1,00	0,82
mir-432	hsa-mir-432	r97									
mir-433	hsa-mir-433	x	mmu-mir-433	x	rno-mir-433	x	p	p	p	1,00	0,03
mir-485	hsa-mir-485		mmu-mir-485		rno-mir-485		p	p	p	0,92	1,00
mir-493	hsa-mir-493	r65			rno-mir-493	r65			p	0,90	1,00
mir-540			mmu-mir-540		rno-mir-540			p		0,94	1,00
mir-541			mmu-mir-541		rno-mir-541			p		0,94	0,98
mir-668	hsa-mir-668		mmu-mir-668				p			1,00	0,87
mir-680			mmu-mir-680-3	u							
mir-770	hsa-mir-770		mmu-mir-770				–			1,00	1,00

Fortsættes på næste side.

Tabel 16: *Fortsat fra foregående side.*

							BLAT-test				
Familie	hsa-miRNA	Ekskl.	mmu-miRNA	Ekskl.	rno-miRNA	Ekskl.	h:m	m:r	r:h	RNAz	RNAmicro
Ingen	hsa-mir-453										
	hsa-mir-544	r100									
	hsa-mir-624	u									
	hsa-mir-625	u, r100									
	hsa-mir-654										
			mmu-mir-434	x, r100							
			mmu-mir-665								
			mmu-mir-666	r18							
			mmu-mir-667								
			mmu-mir-673								
			mmu-mir-679								
			mmu-mir-681	u							

Det betyder sandsynligvis, at ingen af de miRNA gener der ikke er med i testmaterialet, ville blive fundet hvis testen havde omfattet hele kromosomerne hsa 14, mmu 12 og rno 6. Da ingen af dem ikke findes hos alle tre arter, når der kun ses på de tre kromosomer.

De miRNA gener der er blevet maskeret ved exon maskeringen er markeret med "x" i tabel 16. Der er tolv exon maskerede miRNAer. Alle de exon maskerede miRNA gener ligger 3'UTR regioner. Tre af de maskerede gener findes hos alle tre arter. For et gen der findes hos alle tre arter, er det kun hos mus og rotte men ikke mennesket, at genet er maskeret. Desuden er der et miRNA gen mere maskeret hos musen.

Hvilke miRNA gener der blev exonmaskerede gennem pipelinen, blev bestemt ved at exonmaskere, som i pipelinen, en ikke repeatmaskeret udgave af DNA-sekvensen for hver kromosom, og derefter udtrække miRNA-sekvenserne fra DNA-sekvensen med programmet `sampleSequencesmiRNA.py`. Efterfølgende kunne det ses hvilke miRNA-sekvenser, hvor nukleotiderne var blevet udskiftet med N'er.

Hvis der ikke er homologe miRNA gener i alle tre arter, vil disse miRNA gener ikke kunne findes gennem pipelinen. Det kan dog være muligt at finde miRNA gener, selvom der ud fra oplysningerne i tabel 16 ikke findes homologe gener i alle tre arter, ved at finde nye miRNA gener, der ikke findes i miRBase release 9.0.

De repeatmaskerede miRNA gener er markeret med "r" og et tal, i tabel 16. Tallet angiver hvor mange procent af det enkelte gen, der er maskeret, da det er meget forskelligt hvor meget de enkelte gener er maskeret. I alt er 15 miRNA gener helt eller delvist maskeret. For de fleste gener er der den samme maskering i de homologe gener. Et enkelt sted er det kun menneskets miRNA gen, der er maskeret.

Hvilke miRNA gener der er repeatmaskeret i de repeatmaskerede udgaver af DNA-sekvenserne for de enkelte kromosomer, blev bestem ved at anvende de repeatmaskerede udgaver af DNA-sekvenserne, der kan downloades fra Ensembl. Først blev alle ny linie "\n" tegn fjernet fra sekvenserne, som i pipelinen med programmet `oneLineRename.py`, derefter blev miRNA-sekvenserne trukket ud fra DNA-sekvensen med programmet `sampleSequences-miRNA.py`. Efterfølgende kunne det ses hvilke miRNA-sekvenser, hvor nukleotiderne var blevet udskiftet med N'er.

7.2.1 Test af BLAT, RNAz og RNAmicro

Før testen af pipelinen blev der foretaget en test af programmerne BLAT, RNAz og RNAmicro, hvor inputdata kun består af pre-miRNA sekvenserne fra kromosomerne hsa 14, mmu 12 og rno 6, for at få en ide om hvilke pre-miRNAer, der vil passere igennem pipelinen, og hvilke der risikere at forsvinde undervejs.

BLAT

Først blev pre-miRNA sekvenserne for de tre kromosomer samlet i hver deres FASTA-fil. Sekvenserne blev udtrukket fra en lokal installation af miRBase, og konverteret til DNA sekvenser. Dette blev gjort med programmet `premiRNAsequences.py`.

Der blev foretaget seks BLAT alignments, således at hver kromosoms pre-miRNA sæt indgik i en alignment både som database- og som query-fil overfor hver af de to andre arters datasæt. Parametrene blev sat til `-minIdentity=70` og `-out=axt`, til alle øvrige benyttedes default.

Output filerne blev herefter gennemgået manuelt, og resultatet er vist i tabel 16 i de tre "BLAT-test" kolonner.

Bogstaverne i de tre kolonner har følgende betydning:

h:m	Menneske mus alignments resultater
m:r	Mus rotte alignments resultater
r:h	Rotte menneske alignments resultater
p	Alignment i hele pre-miRNA længden eller næsten (samme række)
ns	Alignment kun i del af pre-miRNA længden, "n" angiver antallet af ikke overlappende alignments
f	Som p, men alignment med en pre-miRNA inden for familien (forskellige rækker)
–	Forventet eller måske forventet en alignment, men ingen fundet
/	Resultat er afhængig af hvilken art, der er database og query

De pladser i de tre kolonner, som er tomme, er hvor der ikke blev fundet en alignment, og heller ikke var forventet at finde en alignment.

Ud fra tabel 16 kan det ses, at langt de fleste pre-miRNA sekvenser alignes i hele deres længde. For enkelte er det kun stem regionerne, der alignes. Inden for mir-368 familien er mange alignments mellem de forskellige medlemmer af familien. Det er kun få steder, hvor der ikke forekommer en alignment, hvor det kunne forventes, at der ville være en alignment. Blandt alignments mellem menneske og rotte er der enkelte steder forskel i resultatet, som følge af hvilken art, der vælges til database- og query art.

RNAz

For hver række i tabel 16, der indeholder to eller tre miRNAer, blev en ClustalW alignment af disse miRNAer downloaded fra miRBase og gemt i en fil, og brugt til inputdata til RNAz. Enkelte af de rækker med kun en miRNA, blev slået sammen med andre rækker indenfor samme familie, for at få en alignment og dermed en RNAz score. Det drejer som følgende miRNAer: 1) hsa-mir-487a blev tilføjet de tre mir-487b, scoren for de tre mir-487b blev bestemt af en alignment kun bestående af disse tre sekvenser.

2) hsa-mir-368 blev tilføjet mmu-mir-376c og rno-mir-376c, scoren for de to mir-376c blev bestemt af en alignment kun bestående af disse to sekvenser. 3) hsa-mir-655 og hsa-mir-656 blev slået sammen til en alignment. 4) hsa-mir-329-1 og hsa-mir-329-2 blev også slået sammen til en alignment.

Alle alignments er analyseret med RNAz i hele deres længde på en gang, de er ikke blevet analyseret ved at tage delsekvenser af de forskellige alignments. Resultatet er vist i kolonnen "RNAz" i tabel 16. Maksimum score er 1,00 og RNAz klassificere en alignment som ncRNA, hvis scoren er $\geq$ 0,50. Alle alignments bliver klassificeret som ncRNA i testen. Alignmentet med den laveste score er alignmentet, der indeholder mir-299'erne, scoren er 0,56. Alle de øvrige alignments har en score $\geq$ 0,88. hvis minimum scoren vælges til 0,50 vil alle pre-miRNAerne blive klassificeret korrekt af RNAz.

RNAmicro

De samme ClustalW-filer, der blev brugt til analyse af RNAz blev brugt som inputdata til RNAmicro.

Alle alignments er analyseret med RNAmicro i hele deres længde på en gang. De alignments der ikke opnåede maksimum scoren 1,00 ved denne analyse og havde en alignment længde $\geq$ 70 nt., blev efterfølgende analyseret med alle vindues størrelser fra 70 nt. op til alignment længden, dog max 130 nt., for at finde den højeste RNAmicro score for en alignment. Dette blev gjort, fordi det viste sig, at for enkelte alignments ville RNAmicro kun give en høj score inden for er smalt område, hvad angår vindues størrelse. Sidstnævnte analyse blev foretaget med programmet `RNAmicroTestRange.py`.

Resultatet er vist i kolonnen "RNAmicro" i tabel 16. Maksimum score er 1,00 og RNAmicro klassificere en alignment som miRNA, hvis scoren er $\geq$ 0,50. Langt de fleste alignments har en score på 1,00, men enkelte har også en meget lav score. Mir-380 alignmentet har en score på kun 0,01, mir-433 alignmentet har en score på 0,03, og mir-299 alignmentet en score på 0,11. Dette betyder, at det ikke kan forventes, at RNAmicro vil klassificere alle kendte pre-miRNAer korrekt, selv om der findes homologe miRNAer i alle tre arters testmateriale.

Grunden til, at der er blevet brugt ClustalW alignments i testen af RNAz og RNAmicro i stedet for MLAGAN, som anvendes i pipelinen, er den at ClustalW alignments kan downloades fra miRBase. Det var derfor den hurtigste måde at skaffe de anvendte alignments.

Alle inputfilerne der blev brugt til test af de tre programmer og outputfilerne fra BLAT alignments findes på specialets hjemmeside.

7.2.2 Gennemførsel af testen

Første del af pipelinen – forbehandling – blev gennemført på FASTA-filer, der indeholdt de komplette DNA-sekvenser, for de tre valgte kromosomer

hsa 14, mmu 12 og rno 6. Herunder blev exon for protein-kodende gener maskeret, mens gener for ikke-kodende gener ikke blev maskeret, for at se hvordan pipelinen ville håndtere disse gener.

Forbehandlingen blev foretaget på henholdsvis et sæt DNA-sekvenser, der ikke var blevet maskeret med RepeatMasker, og et sæt som var blevet maskeret med RepeatMasker. Grunden til at der blev anvendt to sæt data var, at 15 miRNA gener berørtes af repeatmaskering med RepeatMasker, i de tre valgte kromosomer. Det ville derfor være en fordel, hvis pipelinen kunne anvendes på DNA-sekvenser, der ikke var repeatmaskeret. Da kun 5 mbp. for hver art i de to datasæt skulle anvendes i den resterende del af pipelinetesten, blev disse DNA-delsekvenser kopieret ud af de forbehandlede DNA-sekvenser med programmet `FASTAsubsequence.py`.

Først blev pipelinen testet med de valgte 5 mbp., der ikke var maskeret med RepeatMasker, for at inkludere de miRNA gener, der ellers ville være maskeret i testen. Dette viste sig dog at være en dårlig ide, da mange repeterende sekvenser endte med at blive klassificeret som potentielle miRNA gener af RNAmicro, og mange af dem med en meget høj score.

Derefter blev pipelinen testet med de samme 5 mbp., men denne gang med repetemaskerede sekvenser. Dette gav et helt anden resultat. De to tests er sammenlignet i tabel 17.

Der er meget stor forskel på hvor mange alignments og poster i databasen, de to test resulterer i. Når de repeterende sekvenser ikke er maskeret, bliver der mange flere alignments. Selve kørslen af testen tager også meget mere tid. 54t. 7m. for de umaskerede sekvenser og 1t. 20m. for de maskerede sekvenser. Tiderne skal tages med en vis forbehold, da der ikke er registreret hvilke andre programmer, der evt. kørte på den computer, der blev anvendt til testen.

Som det ses ud af tabel 17, så blev en alignment fra den første BLAT alignment ikke med taget i `axtInformation` databasetabellen. Dette skyldes at en alignment, havde en negativ score.

Det ses også ud fra tabel 17 at der er forskel på antal alignments i den anden BLAT alignment og antal poster i `tripleInformation` databasetabellen, det burde være det samme antal, de to steder. At de ikke er ens skyldes, den måde hvorpå de to sekvenser i den anden BLAT alignment bliver samlet med den sekvens fra den første alignment, der ikke indgik i den anden alignment. Som omtalt under beskrivelsen af pipelinen skal denne sekvens evt. forkortes. Det er her at problemet opstår, da BLATz bruges til dette formål, og i visse situationer resultere anvendelsen af BLATz i, at der ikke bliver nogen alignment, hvilket betyder at der ikke bliver noget resultat, som skal placeres i `tripleInformation` databasetabellen. Det ser dog ikke ud til det har nogen betydning for, hvor mange mange miRNA gener, der kommer helskindet igennem pipelinen.

Ud fra tabel 17 kan det ses at flere af posterne i `RNAmicroInformation` databasetabellen vedrører de samme sekvenser. Antallet af sekvenser er min-

Antal	RM	ikke RM
BLAT 1. alignment	1.642	18.111
axtInformation poster	1.642	18.110
BLAT 2. alignment	538	37.661
tripleInformation poster	494	16.390
RNA klassificerede	281	3.329
RNAmicroInformation poster	152	719
hsa, sekvens positioner	74	156
hsa, kendte miRNAer poster	85	90
hsa, nye miRNAer poster	8	10
hsa, kendte miRNAer fundet, begge test	30	30
hsa, kendte miRNAer fundet, en test	1	4
Tid	1t. 20m.	54t. 7m.

Tabel 17: Sammenligning af pipelinen ved brug af inputdata, hvor repeterende sekvenser er maskeret (RM) eller ikke er maskeret (ikke RM).

dre end antal poster i `RNAmicroInformation` databasetabellen. Der er flere årsager til dette. I forbindelse med den første BLAT alignment bliver en af sekvenserne splittet op i mindre sekvenser, hvor der laves overlap i sekvenserne, så slutdelen af en sekvens også er start sekvensen i den næste sekvens. Derved optræder sekvensen to gange i sekvensmaterialet. De sekvenser der findes ved de to BLAT alignments, bliver forlænget til at have en længde på minimum 200 nt., hvorved der kan forekomme overlap. Dette kan forekomme, hvis det kun er stem regionerne i en pre-miRNA-sekvens, der bliver alignet, og derved resulterer i to alignments. En sekvens i en art godt kan alignes til flere steder i en af to andre arters sekvenser. Desuden kan en arts sekvens og den omvendt komplementære sekvens, godt alignes til et sted i en af andre to arters sekvenser. RNAmicro tester både på de fundne multiple alignments og de omvendt komplementære alignments. Hvis de begge score $\geq 0{,}50$ resulterer det i to poster i databasen, men regnes kun for en position.

Hvis der ses på overlap hos menneskets sekvenser, så resulterer de 152 poster i 74 sekvens positioner for de repeatmaskerede sekvenser, og det gælder for næsten alle sekvenser, at hvor der er overlap, så har sekvenserne en start- og slut position inden for en afstand på 20 nt. eller mindre.

Dette gælder ikke for den ikke repeatmaskerede test. Der er en række eksempler på løbende overlap, hermed forstås at startpositionen for de enkelte sekvenser, der overlapper, forskydes med nogle få nukleotider til den ene side, hvis sekvenserne sorteres efter startposition. Den mest omfangsrige eksempel, vedrører 134 sekvenser med lænder på 75-200 nt., og sekvenserne dækker et sekvensområde på 451 nt. 108 af sekvenserne havde en RNAmicro score på $\geq 0{,}99$.

RNAz klassificere en langt højere andel af de fundne alignments, som RNA-sekvenser, når der anvendes repeatmaskerede sekvenser end ikke repeatmaskerede 56,9% mod 20,3%. Det samme gør sig gældende for RNAmicro, langt højere andel af de sekvenser, som RNAz klassificerer som RNA-sekvenser klassificere RNAmicro som miRNA-sekvenser 54,1% mod 21,6%.

Et tilsvarende resultat ses, når der ses på kendte miRNA gener hos mennesket, udgør de en meget større andel af posterne i `RNAmicroInformation` databasetabellen i testen med de repeatmaskerede sekvenser end de ikke repeatmaskerede 55,9% mod 12,5%. Det betyder, at hvis pipelinen anvendes , så vil der være langt større sandsynlighed for at de fundne sekvenser er miRNA gener, hvis der anvendes repeat maskerede sekvenser som inputdata, end hvis inputdata ikke var repeat maskerede.

En interessant ting er, at mir-770 kun blev fundet i den repeat maskerede test. I testen uden repeat maskering, blev mir-770 ikke fundet i den første BLAT alignment. Det er dog ikke nærmere undersøgt, hvorfor den ikke blev alignet. Men maskeringen kan tilsyneladende have indflydelse på hvilke alignments, der findes med BLAT.

Konklusionen må blive, at det er en klar fordel at anvende DNA-sekvenser, der er repeat maskerede. Da der vil være mere sandsynligt, at de sekvenser der findes gennem pipelinen, er ægte miRNA gener.

7.2.3 Pipelineresultatet

Til den detaljerede analyse af pipelinen er anvendt resultatet, hvor der er anvendt repeat maskerede sekvenser, for at undgå problemerne med repeat sekvenserne, som beskrevet ovenfor.

Tabel 18 giver et detaljeret beskrivelse at resultatet for de kendte miRNA gener på de tre anvendte kromosomer hsa 14, mmu 12 og rno 6, som test sekvenserne kommer fra. miRNAerne er opstillet på samme måde, som i tabel 16.

De to BLAT kolonner viser resultatet af de to BLAT alignments, der blev anvendt til at finde konserverede DNA-sekvenser mellem de tre anvendte arter. Parametrene til BLAT i de to alignments var `-minIdentity=70 -out=axt`, øvrige parametre var default, dvs. at seed strategien var to perfekt match, og en ordlængde på 11 nt. I den første alignment var menneske database-sekvenser og musen query-sekvensen. I den anden BLAT alignment var rotten database-sekvenser og musen query-sekvensen. Alle musesekvenser der blev fundet i den første alignment, blev anvendt som query-sekvenser i den anden alignment.

"h:m" viser resultatet for den første alignment, og "r:m:h" viser resultatet for den anden alignment samlet med resultatet af den første alignment. Bogstaverne "p", "f" og "s" har samme betydning som beskrevet for tabel 16. "+h" (menneske), "+m" (mus) og "+r" (rotte) angiver, at der er fundet nye homologe sekvenser til de kendte miRNA gener for det pågældende art. Hvis

Tabel 18: Oversigt over miRNA gener hos hsa 14, mmu 12 og rno 6. Med angivelse af resultaterne fra pipelinetesten. Se teksten for forklaring på bogstav symbolernes betydning.

				BLAT		RNAmicro		Pipeline
Familie	hsa-miRNA	mmu-miRNA	rno-miRNA	h:m	r:m:h	ma h:m:r	ok h:m:r	resultat
mir-127	hsa-mir-127	mmu-mir-127	rno-mir-127					x
mir-134	hsa-mir-134	mmu-mir-134	rno-mir-134	p	p	0,79	0,60	(✓)
mir-136	hsa-mir-136	mmu-mir-136	rno-mir-136					x
mir-153		mmu-mir-153	rno-mir-153					u
mir-154	hsa-mir-154	mmu-mir-154	rno-mir-154	p	p	1,00	1,00	✓
		mmu-mir-300	rno-mir-300	+1h	+1h	1,00	1,00	✓, +hsa
	hsa-mir-323	mmu-mir-323	rno-mir-323	p	p	0,99		✓
	hsa-mir-369	mmu-mir-369	rno-mir-369	p	p	1,00	1,00	✓
	hsa-mir-377	mmu-mir-377	rno-mir-377	p	p	1,00		✓
	hsa-mir-381	mmu-mir-381	rno-mir-381	p	p	1,00	1,00	✓
	hsa-mir-382	mmu-mir-382	rno-mir-382	p	p	1,00	0,95	✓
	hsa-mir-409	mmu-mir-409	rno-mir-409	p	p	1,00	1,00	✓
	hsa-mir-410	mmu-mir-410		p	+r	1,00	0,98	✓, +rno
	hsa-mir-487a							BLAT
	hsa-mir-487b	mmu-mir-487b	rno-mir-487b	p	p			rm5, Rmi
	hsa-mir-494	mmu-mir-494	rno-mir-494	p	p	1,00	1,00	✓
	hsa-mir-496	mmu-mir-496		p	+r	1,00	1,00	✓, +rno
	hsa-mir-539	mmu-mir-539	rno-mir-539	2s	1s	1,00	1,00	✓

Fortsættes på næste side.

Tabel 18: *Fortsat fra foregående side*

Familie	hsa-miRNA	mmu-miRNA	rno-miRNA	BLAT h:m	BLAT r:m:h	RNAmicro ma h:m:r	RNAmicro ok h:m:r	Pipeline resultat
	hsa-mir-655							BLAT
	hsa-mir-656							BLAT
mir-203	hsa-mir-203	mmu-mir-203	rno-mir-203	p	p	1,00	1,00	✓
mir-208	hsa-mir-208							u
mir-299	hsa-mir-299	mmu-mir-299	rno-mir-299	p	p	1,00	1,00	✓
mir-329		mmu-mir-329	rno-mir-329	f	f	1,00	0,66	✓
	hsa-mir-329-1			f	f	1,00	0,66	✓
	hsa-mir-329-2			f	f	1,00		✓
	hsa-mir-495	mmu-mir-495		2s,f,+h	+h,+2r	1,00	1,00	✓, +rno, +hsa
		mmu-mir-543	rno-mir-543	f,+h	f,+h	1,00	1,00	✓, +hsa
mir-337	hsa-mir-337	mmu-mir-337	rno-mir-337	p	p	1,00	1,00	✓
mir-341		mmu-mir-341	rno-mir-341					rm77, BLAT
mir-342	hsa-mir-342	mmu-mir-342	rno-mir-342	p	p	1,00	1,00	✓
mir-345	hsa-mir-345	mmu-mir-345	rno-mir-345					rm40, BLAT
mir-368	hsa-mir-368			f	f,+r	1,00, (0,90)	1,00	✓, (+rno)
		mmu-mir-376a	rno-mir-376a	f	f	1,00	1,00	✓
	hsa-mir-376a-1			f	f,+r	1,00	1,00	✓
	hsa-mir-376a-2			f	f,+r	1,00	1,00	✓
	hsa-mir-376b	mmu-mir-376b	rno-mir-376b	p,f	p,f,+r	1,00, (0,92)	1,00	✓, (+rno)
		mmu-mir-376c	rno-mir-376c	f	f	1,00	1,00	✓
mir-370	hsa-mir-370	mmu-mir-370	rno-mir-370					rm100

Fortsættes på næste side.

Tabel 18: *Fortsat fra foregående side*

				BLAT		RNAmicro		Pipeline
Familie	hsa-miRNA	mmu-miRNA	rno-miRNA	h:m	r:m:h	ma h:m:r	ok h:m:r	resultat
mir-379	hsa-mir-379	mmu-mir-379	rno-mir-379	p	p	1,00	0,99	✓
	hsa-mir-380	mmu-mir-380		p	+r	1,00	0,92	✓, +rno
	hsa-mir-411	mmu-mir-411		p	+r	1,00	1,00	✓, +rno
	hsa-mir-758	mmu-mir-758		p	+r	1,00		✓, +rno
mir-412	hsa-mir-412	mmu-mir-412	rno-mir-412	p	p	1,00	0,73	✓
mir-431	hsa-mir-431	mmu-mir-431	rno-mir-431					x
mir-432	hsa-mir-432							rm97
mir-433	hsa-mir-433	mmu-mir-433	rno-mir-433					x
mir-485	hsa-mir-485	mmu-mir-485	rno-mir-485	p	p	1,00	1,00	✓
mir-493	hsa-mir-493		rno-mir-493	+1m	+1m			rm65, Rmi
mir-540		mmu-mir-540	rno-mir-540					BLAT
mir-541		mmu-mir-541	rno-mir-541					BLAT
mir-668	hsa-mir-668	mmu-mir-668		p	+r	1,00		✓, +rno
mir-680		mmu-mir-680-3						u
mir-770	hsa-mir-770	mmu-mir-770		1s	+2r	1,00	1,00	✓, +rno

Fortsættes på næste side.

Tabel 18: *Fortsat fra foregående side*

Familie	hsa-miRNA	mmu-miRNA	rno-miRNA	BLAT h:m	BLAT r:m:h	RNAmicro ma h:m:r	RNAmicro ok h:m:r	Pipeline resultat
Ingen	hsa-mir-453			+2m	+1m,+1r			Rmi
	hsa-mir-544							rm100
	hsa-mir-624							u
	hsa-mir-625							u, rm100
	hsa-mir-654			+m	+m,+r			Rmi
		mmu-mir-434						x, r100
		mmu-mir-665		+1h	+1h,+1r	1,00	1,00	✓, +hsa, +rno
		mmu-mir-666						rm18, BLAT
		mmu-mir-667						BLAT
		mmu-mir-673						BLAT
		mmu-mir-679						BLAT
		mmu-mir-681						u

der er tilføjet et tal angiver det at der er tale om en eller to delsekvenser.

Resultatet i "h:m" kolonnen er fremkommet ved først at analysere `axtInformation` databasetabellen med programmet `miRNAaligned.py` for hver af de to arter. Resultatet herfra blev efterfølgende analyseret med `miRNAfoundmiRNA.py` for at finde hvilke miRNA gensekvenser, der blev alignet med hvilke miRNA gensekvenser, og hvilke kendte miRNA gensekvenser, der blev alignet med sekvenser, der endnu ikke er klassificeret som miRNA gensekvenser. Til sidst blev en manuel analyse foretaget, for at klassificere de enkelte miRNA alignments som "p", "f", "s", "+h", "+m" eller "+r".

Resultatet i "r:m:h" fremkom på tilsvarende måde. Først analyseredes `tripleInformation` databasetabellen med programmet `miRNAaligned.py` for hver af de tre arter. Resultatet herfra blev ligeledes efterfølgende analyseret med `miRNAfoundmiRNA.py` for at finde hvilke miRNA gensekvenser, der blev alignet med hvilke miRNA gensekvenser, og hvilke kendte miRNA gensekvenser, der blev alignet med sekvenser, der endnu ikke er klassificeret som miRNA gensekvenser. Til sidst blev der også foretaget en manuel analyse, for at klassificere de enkelte miRNA alignments som"p", "f", "s", "+h", "+m" eller "+r".

De to RNAmicro kolonner viser resultatet af RNAmicro analysen af de multiple alignede konserverede sekvenser, der blev fundet gennem de to alignments, og den efterfølgende positive klassifikation af RNAz som ncRNA-sekvenser.

MLAGAN blev brugt med parameteren `-tree ''(hsa (mmu rno))''`, der angiver, at først skal musen og rottens sekvens alignes, derefter skal menneske sekvensen alignes til alignmentet af de to andre sekvenser. Til RNAz blev anvendt en vindues størrelse på 100 nt. og i trin på 10. Vindues størrelsen blev valgt for at kunne dække længden på en pre-miRNA. Trinene er ret korte, det skyldes at den ikke var blevet ændret fra da RNAz skulle finde positioner i de multiple alignede sekvenser med ncRNAer, som senere blev opgivet, pga. RNAmicro ikke kunne anvende disse positioner.

RNAmicro blev brugt med vindues størrelser på 75 nt., 95 nt. og 115 nt. Disse blev valgt ud fra simple test på multiple alignments af pre-miRNAer dovnloaded fra miRBase. Der blev ikke foretaget en grundig test for at finde de helt rigtige vindues størrelser. Hertel and Stadler (2006a) brugte i sine egne test vindues størrelser på 70 nt., 100 nt. og 130 nt.

"ma h:m:r" angiver resultatet for de fundne og multiple alignede sekvenser. "ok h:m:r" angiver resultatet for de omvendt komplementære sekvenser. Det er RNAmicro scoren, der er angivet med to decimaler.

Alle miRNA gener i testområdet ligger på plus-strengen, og det er plus-strenge, der er anvendt som input til pipelinen. Det har medført at alle "ma h:m:r" resultater vedrører plus-strenge, og "ok h:m:r" vedrører minus-strenge. Der var dog en undtagelse mir-299, hvor der både fantes en multiple alignment med alle plus-strenge, og en hvor rottens streng var minus-strengen.

Begge alignments fik samme RNAmicro score.

Resultatet i de to RNAmicro kolonner er fremkommet ved at analysere `RNAmicroInformation` databasetabellen med programmet `miRNAfound.py` for hver af de tre arter. Resultatet herfra blev efterfølgende analyseret med `miRNAfoundmiRNA.py` for at finde hvilke miRNA sekvenser, der blev alignet med hvilke miRNA-sekvenser, og hvilke kendte miRNA gensekvenser, der blev alignet med sekvenser, der endnu ikke er klassificeret som miRNA gensekvenser. Til sidst blev en manuel analyse foretaget, for at få placeret de enkelte RNAmicro scorer i tabellen.

Det endelige resultat af pipelinen findes i kolonnen "Pipeline Resultat". Kolonnen angiver om pre-miRNAerne blev fundet, fundet ved hjælp af nye homologe sekvenser eller ikke fundet. Betegnelserne i kolonnen skal forstås på følgende måde:

✓	Fundet
+nnn	Fundet ved hjælp af nye homologe sekvenser hsa: menneske, mmu: mus, rno: rotte
BLAT	Ikke fundet i de to BLAT alignments
Rmi	RNAmicro score $< 0{,}50$
u	Udenfor testområdet
x	Exon maskeret
rm00	RepeatMask maskeret, tallet angiver antal % maskeret
()	RNAmicro score $< 0{,}99$

På specialets hjemmeside findes filer med outputtet fra programmerne, der er nævnt ovenfor.

Resultatet i tabel 18 er samlet i tabel 19 for at give et samlet overblik over resultatet for kendte miRNAer. Tabel 20 giver en overblik af resultatet for alle fundne sekvenser.

7.2.4 Analyse af pipelineresultatet

I testen klassificeres en alignment som en potentiel miRNA, hvis RNAmicro scoren er $\geq 0{,}99$. Dette er en høj grænse. Hertel and Stadler (2006a) bruger selv en grænse på 0,90. Scoren er sat så høj, fordi at hvis den sænkes til 0,90, så ville ikke flere kendte miRNAer blive klassificeret som miRNAer. Der er dog et tilfælde med en ny rotte miRNA gen, men det er et special tilfælde, som er diskuteret nærmere i afsnittet om nye miRNAer,og skal derfor ikke regnes med. Sættes grænsen til 0,90 så ville fem mere af de alignments, hvor ingen af sekvenserne kommer fra kendte miRNA gener.

Effektiviteten af pipelinen er beregnet ud fra, hvor mange af de kendte miRNA, der er i input datasættet, som findes gennem pipelinen. Effektiviteten er beregnet ud fra tallene i tabel 19 og er vist i tabel 21. Der er anvendt tre måder til at bestemme effektiviteten.

miRNA	hsa 14	mmu 12	rno 6	I alt
Antal på kromosom	49	50	35	134
Udenfor test (u)	3	3	1	7
Antal i test	46	47	34	127
Maskeret i test (x, r)	9	8	8	25
Være muligt at finde i test	37	39	26	102
Fundet, score $\geq$ 0,99 i test	30	31	22	83
Ikke fundet, score $\geq$ 0,99 i test	7	8	4	18
Fundet, score 0.50–0.98 i test	1	1	1	3
Ikke 3 * homologi	3	6	2	11
Nye homologe til kendte, score $\geq$ 0,99 i test	3	0	9	12
Nye homologe til kendte, score 0.50–0.98 i test	0	0	1	1

Tabel 19: Samlet resultat for pipelinetesten vedrørende kendte miRNAer.

Fundet	hsa 14	mmu 12	rno 6	I alt
Sekvenser	74	72	74	220
Sekvenser, score $\geq$ 0,99	48	46	46	140
Kendte miRNA, score $\geq$ 0,99	30	31	22	83
Nye homologe til kendte, score $\geq$0,99	3	0	9	12
Ikke kendte miRNA, alle 3 sekvenser, score $\geq$ 0,99	14	14	14	45
snoRNA, score $\geq$ 0,99	1	1	1	3
Negative, score $<$ 0,99	25	25	26	76
Ikke kendte miRNA, alle 3 sekvenser, score 0.90–0.98	5	5	5	15
Kendte miRNA, score 0.90–0.98	0	0	0	0
Nye homologe til kendte, score 0.90–0.98	0	0	1	1

Tabel 20: Samlet resultat for pipelinetesten vedrørende alle fundne sekvenser.

miRNA	hsa 14	mmu 12	rno 6	Samlet
Alle i test	65,2	66,0	64,7	65,4
Kun umaskerede	81,1	79,5	84,6	81,4
Kun homologe	88,2	94,0	91,2	91,2

Tabel 21: Effektiviteten af pipelinen vist for hver art og samlet.

Den første er, hvor alle kendte miRNAer i testmaterialet er medtaget i beregningerne, det giver en effektivitet på 65%. Dette kan siges at være den overordnede effektivitet.

Den anden måde er kun at se på de miRNAer, der ikke er blevet ma-

skeret, eller ikke berørt af maskering, forstået på den måde, at hvis blot en af miRNAerne i en række i tabel er maskeret så tælles den med i alle tre arter, som maskeret. Da det på grund af maskeringen i en art ikke vil være muligt at finde homologe sekvenser i alle tre arter. F.eks. er hsa-mir-136 ikke exon-maskeret mens mmu-mir-136 og rno-mir-136 er det, derfor er hsa-mir-136 talt som en maskeret miRNA i tabel 19. Det samme gælder for mir-345, hvor kun hsa-mir-345 er repeat maskeret.

Et problem er, at alle repeat maskeringerne ikke er 100% maskeringer. Det har derfor været nødvendigt at se på, hvor maskeringen er placeret i de enkelte sekvenser. Desuden skal der tages højde for at sekvenspositionerne for pre-miRNAer i miRBase er plus flankerende sekvenser. Det har betydet at mir-487b ikke er talt som maskeret, fordi de 4-5% maskering ligger udenfor selve pre-miRNA sekvensen. Det samme gør sig gældende for mmu-mir-666. For de øvrige miRNAer, som er berørt af repeat-maskering, er klassificeret som maskeret. Ved kun at se på de ikke maskerede miRNAer, bliver effektiviteten af pipelinen 81%.

Den tredje måde at beregne effektiviteten på er kun at se på de miRNAer, hvor der findes homologe sekvenser i alle tre arter, enten som kendte miRNAer eller nye homologes sekvenser. For det kan jo tænkes, at de homologe miRNAer for en eller to arters vedkommende ikke ligger inden for det anvendte testområde, det vil derfor ikke være muligt at opnå 3*homologi, dvs. finde homologe sekvenser i alle tre arter, og dermed ikke kunne findes med pipelinen. Hvis der kun ses på de miRNAer, der er umaskerede og har 3*homologi, så bliver effektiviteten 91%.

RNAz var ikke årsag til, at nogen af miRNA generne i testmaterialet, ikke blev fundet gennem pipelinen.

Det er lidt overraskende at mir-487b ikke findes gennem pipelinen, for i den indledende test, se tabel 16, fik den en score på 1,00. Men det skyldes de få nukleotider, der er maskeret i sekvensen, som endda ligger udenfor den egentlige pre-miRNA. Ved at undersøge det nærmere, viste det sig at hvis den maskerede nukleotid, der ligger nærmest pre-miRNA sekvensen ikke var maskeret, så ville RNAmicro godt klassificere mir-487b som miRNA.

I tabel 18 så har kolonnen "ma h:m:r" 36 rækker med en score $\geq$ 0,99, og kolonnen"ok h:m:r" 25 rækker med en score $\geq$ 0,99. Det betyder at RNAmicro ofte ikke kan sige hvilken streng en miRNA gen er på.

7.2.5 Nye miRNAer og miRBase release 11.0

I testen af pipelinen blev fundet en række nye miRNA gener, der er homologe med allerede kendte miRNA gener. De var nye i forhold til miRBase release 9.0, som er blevet anvendt i denne test.

De er beskrevet i det følgende. De er navngivet efter de kendte homologes navne. Desuden er de nye miRNA gener forsøgt fundet i den nyeste release af miRBase – release 11.0.

Blandt de nye miRNA gener, der er homologe med allerede kendte miRNA gener har en RNAmicro score på $\geq$ 0,99. De er vist i tabel 22. Der er tale om tre nye menneske og ni nye rotte miRNA gener. Ni af de tolv er med i miRBase release 11.0.

Navn	Kromosom	Start	Slut	Streng	miRBase
hsa-mir-300	hsa 14	100577426	100577554	+/-	11.0
hsa-mir-543	hsa 14	100568044	100568186	+/-	11.0
hsa-mir-665	hsa 14	100411097	100411223	+/-	11.0
rno-mir-380	rno 6	134390959	134391089	+	11.0
rno-mir-410	rno 6	134425051	134425173	+	11.0
rno-mir-411	rno 6	134389314	134389443	+/-	11.0
rno-mir-495	rno 6	134399186	134399329	+/-	11.0
rno-mir-496	rno 6	134420333	134420469	+/-	
rno-mir-665	rno 6	134177271	134177399	+/-	
rno-mir-668	rno 6	134416200	134416313	+	
rno-mir-758	rno 6	134391982	134392081	+	11.0
rno-mir-770	rno 6	134141329	134141471	+/-	11.0

Tabel 22: Nye miRNA gener fundet gennem pipelinen med en RNAmicro score på $\geq$ 0,99, og som er homologe med kendte miRNA gener.

Der blev fundet en rotte sekvens, se tabel 23, som er homolog med flere miRNA gener i mir-368 familien. forklaringen på at sekvensen ikke scorer mere end 0,92 kan forklares med at ca. 40% af sekvensen består af N'er. Dette skyldes, at den del af sekvensen endnu ikke er blevet bestemt.

Navn	Kromosom	Start	Slut	Streng	Score
p376-rno-mir	rno 6	134401792	134401891	+	0,92

Tabel 23: Ny miRNA gen fundet gennem pipelinen med en RNAmicro score på 0,92, og som er homologe med kendte miRNA gener. At scoren ikke er $\geq$ 0,99 kan forklares med at ca. 40% af sekvensen ikke er bestemt endnu.

Der blev desuden fundet fem nye miRNA gener, se tabel 24, der er homologe med kendte miRNA gener, disse havde dog ikke en score der var høj nok til at blive klassificeret som miRNA gener af RNAmicro. Tre fra mus, som også er kommet med i miRBase release 11.0 og to fra rotte. At mmu-mir-493 ikke blev klassificeret af RNAmicro, kan forklares med, at hsa-mir-493 og rno-mir-493 begge 65% repeatmaskeret. Det er ikke forsøgt at finde en nærmere forklaring hvorfor de øvrige ikke er blevet klassificeret af RNAmicro. De angivne positioner er positionerne fra den anden BLAT alignment.

Navn	Kromosom	Start	Slut	Streng	miRBase
mmu-mir-453	mmu 12	110183425	110183481	+	11.0
mmu-mir-493	mmu 12	110027990	110028065	+	11.0
mmu-mir-654	mmu 12	110171019	110171099	+	11.0
rno-mir-453	rno 6	134417046	134417102	+	
rno-mir-654	rno 6	134403976	134404056	+	

Tabel 24: Nye miRNA gener fundet gennem de to BLAT alignments, og som er homologe med kendte miRNA gener. De er dog af forskellige grunde ikke blevet klassificeret som miRNA gener af RNAmicro.

Ved at sammenligne resultatet af pipelinetesten og positionerne i miRBase release 11.0 for miRNA generne for de tre anvendte kromosomer, kunne det konstateres at blandt de fundne sekvenser, var en mere miRNA gen, men med en meget lav RNAmicro score, se tabel 25. Bemærk at rno-mir-1197 ikke er med i miRBase release 11.0.

Navn	Kromosom	Start	Slut	Streng	Score	miRBase
hsa-mir-1197	hsa 14	100561661	100561755	+/-	0,52/0,57	11.0
mmu-mir-1197	mmu 12	110160154	110160248	+/-	0,52/0,57	11.0
rno-mir-1197	rno 6	134391536	134391630	+/-	0,52/0,57	

Tabel 25: mir-1197 er en ny miRNA miRBase 11.0, den blev også fundet gennem pipelinen, med RNAmicro scoren er ikke høj nok til at blive klassificeret som miRNA når scoren er sat til $\geq 0{,}99$.

7.2.6 Potentielle miRNA gener eller falske

I alt blev der fundet 14 multiple alignments med en RNAmicro score $\geq 0{,}99$, som ikke indeholdte nogen kendte miRNAer, se tabel 20. Mere detaljerede informationer om sekvenserne er vis i tabel 26. De er navngivet efter deres id i `RNAmicroInformation` databasetabellen, hvis navnet indeholder to numre, er det fordi både multiple alignmentet og den omvendt komplementære alignment har scoret højt nok, til at blive klassificeret som miRNA af RNAmicro. Resultatet fra pipeline testen findes i de to kolonner "Streng" og "Score" under "Pipeline test".

Der er ikke foretaget test af de fundne sekvenser, med andre programmer for at se, om de ville klassificere dem som miRNA gener.

Der er foretaget en test, og det er med RNAmicro. For det har vist sig, at RNAmicro nogle gange giver forskellig score, afhængig af om der bruges se-

				Pipeline test		2. RNAmicro test	
Navn	Kromosom	Start	Slut	Streng	Score	Streng	Score
p-4-5	hsa 14	99935758	99935888	+/-	1,00/1,00	+/-	1,00/1,00
	mmu 12	109364008	109364147	+/-	1,00/1,00	+/-	1,00/1,00
	rno 6	133260002	133260139	+/-	1,00/1,00	+/-	1,00/1,00
p-7	hsa 14	100114002	100114094	+	1,00		
	mmu 12	109507158	109507256	+	1,00		
	rno 6	133290583	133290682	-	1,00		
p-21	hsa 14	100401265	100401378	-	1,00	-	1,00
	mmu 12	110024384	110024485	-	1,00	-	1,00
	rno 6	134153984	134154081	-	1,00	-	1,00
p-30-31	hsa 14	100421476	100421589	+/-	0,99/0,99		
	mmu 12	110043801	110043913	+/-	0,99/0,99		
	rno 6	134187664	134187778	+/-	0,99/0,99		
p-32	hsa 14	100436683	100436795	+	1,00		
	mmu 12	110056548	110056658	+	1,00		
	rno 6	134200618	134200729	+	1,00		
p-53	hsa 14	100566116	100566240	+	1,00	+	1,00
	mmu 12	110163485	110163608	+	1,00	+	1,00
	rno 6	134395190	134395313	+	1,00	+	1,00
p-122	hsa 14	100944624	100944738	-	1,00		
	mmu 12	110570781	110570895	-	1,00		
	rno 6	134802594	134802702	-	1,00		
p-123-124	hsa 14	100994123	100994245	+/-	1,00/1,00	+/-	1,00/0,97
	mmu 12	110635914	110636031	+/-	1,00/1,00	+/-	1,00/0,97
	rno 6	134900361	134900478	+/-	1,00/1,00	+/-	1,00/0,97
p-129	hsa 14	101043449	101043572	+	0,99		
	mmu 12	110680854	110680976	+	0,99		
	rno 6	134945251	134945373	+	0,99		
p-132	hsa 14	101676550	101676679	+	1,00	+	1,00
	mmu 12	111186358	111186479	+	1,00	+	1,00
	rno 6	135460822	135460943	+	1,00	+	1,00
p-134	hsa 14	102009751	102009874	+	1,00	+	0.65
	mmu 12	111409605	111409728	+	1,00	+	0.65
	rno 6	135703058	135703181	+	1,00	+	0.65
p-135	hsa 14	102129672	102129760	+	1,00	+	1,00
	mmu 12	111488018	111488106	+	1,00	+	1,00
	rno 6	135784030	135784118	+	1,00	+	1,00
p-141-142	hsa 14	103070209	103070302	+/-	1,00/1,00		
	mmu 12	112130009	112130103	+/-	1,00/1,00		
	rno 6	136464491	136464585	+/-	1,00/1,00		
p-151-152	hsa 14	103705958	103706032	+/-	1,00/0,99		
	mmu 12	112618709	112618774	+/-	1,00/0,99		
	rno 6	136997892	136997957	+/-	1,00/0,99		

Tabel 26: Nye potentielle miRNA gener fundet gennem pipelinen med en RNAmicro score på $\geq 0{,}99$, som ikke er homologe med kendte miRNA gener. Eller er det RNAmicro fejl klassificeringer?

kvenser fra `tripleInformation`- eller fra `RNAmicroInformation` databasetabellen. Forstået på den måde, at de sekvenser, der findes i `RNAmicroInformation` er delsekvenser fra `tripleInformation`, som er blevet klassificeret som mulige miRNAer med en score $\geq$ 0,50 af RNAmicro. Hvis de delsekvenser fra `RNAmicroInformation` genscores med RNAmicro var det forventet, at de ville få samme score, men det er ikke altid tilfældet, hvis de får en anden score er det oftest en mindre score. Det tyder på, at RNAmicro's score er afhængig af delsekvenserne, der ligger udenfor, det område som RNAmicro scorer.

Testen blev foretaget med de samme vindues størrelser 75 nt., 95 nt. og 115. nt. som i pipeline testen. Resultatet fra denne anden RNAmicro test er vist for de 14 alignments i tabel 26 i de to kolonner "Streng" og "Score" under "2. RNAmicro test". Det er den højeste score blandt de tre vindues størrelser, der er angivet. Det blev ikke registreret om det var hele eller kun en del af alignmentet, der fik den høje score. Det kan ses, at nogle holder scoren, seks af sekvenserne gør det, enkelte får en mindre score, mens resten syv alignments bliver slet ikke klassificeret som miRNAer.

Den samme test blev også udført på de alignments med kendte miRNAer. Af resultaterne i kolonne "ma h:m:r" som har en score $\geq$ 0,99 i tabel 18 fik tre en score $\leq$ 0,99. I kolonne "ok h:m:r" fik tretten en score $\leq$ 0,99, hvor de før fik en score $\geq$ 0,99. Det resulterede dog kun i at to mir-323 og mir-412 ikke ville blive klassificeret som miRNAer, hvis denne test blev gjort til en del af pipelinen.

7.2.7 Programmer til analyse af pipeline

Her beskrives de programmer der er blevet udviklet i forbindelse med analysen af pipelinen.

FASTAsubsequence.py

Kopier en delsekvens fra en FASTA-fil, og placere delsekvensen i en ny FASTA-fil. Den nye FASTA-fils identifier er formateret som de reformaterede FASTA-filer i pipelinen. FASTA-filen der kopieres fra må kun have en sekvens, og den skal være på en linie, og startpositionen for sekvensen skal være 1. Programmet kan ikke trække en delsekvens ud af en delsekvens. Input til programmet er sekvensfilnavn, outputfilnavn, start- og slutpositioner for delsekvensen. Hvis startpositionen er mindre end 1 sættes den til 1, og hvis slutpositionen er højere end den i FASTA-filen, sættes den til slutpositionen i FASTA-filen.

sampleSequencesmiRNA.py

Kopier delsekvenser fra en FASTA-fil indeholdende en hel kromosom. Delsekvensernes positioner er positioner for pre-miRNAerne, og som hentes fra

lokal miRBase installation. Start- og slutpositionerne kan sættes til positioner henholdsvis før og efter de positioner, der findes i miRBase databasen. Programmet bruger `FASTAsubsequence.py` til selve kopieringen, derfor skal betingelserne beskrevet der overholdes. De enkelte delsekvenser samles i en FASTA-fil. Input til programmet er sekvensfilnavn, miRBase artkode for art, kromosomnavn og distance, der angiver hvor meget sekvens, der skal tilføjes før og efter pre-miRNA positionerne.

RNAmicroTestRange.py

Kører en RNAmicro test på en multiple alignment på ClustalW-format, dog skal hver sekvens i ClustalW-filen være på en linie. Der testes med en vindues størrelse på 70 nt. op til alignment længden, dog maksimalt 130 nt. Er alignmentet $\leq$ 70 nt. testes hele alignmentet på en gang. Resultaterne skrives ud til skærmen.

RNAzClassificationCount.py

`identification.py` registrer hvilke multiple alignments RNAz klassificere som RNA i en log-fil. Programmet tæller, hvor mange der er klassificeret som RNA af RNAz. Input til programmet er log-filen.

miRNAaligned.py

Undersøger hvilke alignments, efter den første BLAT alignment eller den anden BLAT alignment plus samling af de tre artes sekvenser, der helt eller delvist dækker positionerne for kendte pre-miRNAer. Pre-miRNA positionerne hentes fra en lokal miRBase database. Oplysningerne om de alignede sekvensers positioner hentes fra henholdsvis `axtInformation` og `tripleInformation` databasetabellerne. Programmet kan undersøge en art, et kromosom og en databasetabel af gangen, derfor er input miRBase artskode, kromosomnavn og en kode for hvilke oplysninger fra tabellen, der skal anvendes. “1db” og “1gu” er henholdsvis database- og query arten i førte BLAT alignemnt. “2db” og “2gu” er henholdsvis database- og query arten i anden BLAT alignemnt. “2fi” er arten der var med i første men ikke anden BLAT alignment.

Outputtet fra programmet skrives til skærmen. Det er miRNA navn, post id, Illustration for placerings forholdet mellem alignment-sekvens og pre-miRNA-sekvens, længde på alignment, lænden på det område som både alignment-sekvens og pre-miRNA-sekvens dækker, længde på pre-miRNA, hvilken art/arter oplysningerne hører sammen med.

For at få et hurtigt overblik over hvordan forholdet mellem alignment-sekvens og pre-miRNA-sekvens er, er der lavet nogle små illustrationer, der viser det. “|” viser start- eller slut position for pre-miRNAen.Der anvendes følgende illustrationer:

`--|-->|` Alignment-sekvensen har en lavere startposition end pre-miRNA-sekvensen, dens slutpositionen ligger mellem eller er lig med pre-miRNA-sekvensens start- eller slutposition.

`|<---->|` Alignment-sekvensen har start- og slutposition liggende på eller mellem pre-miRNA-sekvensens start- eller slutposition.

`|<--|--` Alignment-sekvensen har en startposition der ligger mellem eller er lig med pre-miRNA-sekvensens start- eller slutposition, dens slutpositionen ligger ligger højere end pre-miRNA-sekvensens slutposition.

`<-|--|->` Alignment-sekvensen har start- og slutposition, der ligger henholdvis før og efter pre-miRNA-sekvensens start- og slutposition.

miRNAfound.py

Gør det samme som `miRNAaligned.py`, men henter oplysningerne om sekvens positionerne i `RNAmicroInformation`. Outputtet indeholder yderligere følgende punkter: Streng oplysninger for alignment-sekvens og pre-miRNA-sekvens og RNAmicro scoren for alignmentsekvensen.

miRNAfoundmiRNA.py

Outputtet fra `miRNAaligned.py` og `miRNAfound.py` vedrører kun en art, og det er ikke nemt at se hvilke kendte miRNAer der er blevet alignet med hvilke kendte miRNAer, eller blevet alignet med en sekvens, der ikke er en kendt miRNA. Programmet tager som input filer, der indeholder outputtet fra en af de to programmer og en af databaserne, og samler oplysningerne så det er nemt at se hvilke kendte miRNAer, der er blevet alignet med hinanden, og hvilke der er blevet alignet med sekvenser, der ikke er kendte miRNAer.

miRNAcandidateSequences.py

Generer en FASTA-fil med tre sekvenser, der kan alignes og der er blevet klassificeret af RNAmicro som potentielle miRNA gener. Programmet henter oplysningerne fra `RNAmicroInformation` databasetabellen. Input er post-id og sti til FASTA-filerne med DNA-sekvenserne for hele kromosomerne.

miRNAcandidateSingleSequence.py

Gør det samme som `miRNAcandidateSequences.py`, men placerer de tre sekvenser i hver deres fil.

miRNAcandidateAlignment.py

Gør næsten det samme som `miRNAcandidateSequences.py`, men den foretager en alignment ved hjælp af MLAGAN af de tre sekvenser, og gemmer resultatet i en ClustalW formateret fil. Programmet gør brug af to funktioner i `identification.py`.

8 Konklusion

I dette projekt er der blevet udviklet en pipeline til identifikation af potentielle miRNA gener, ved hjælp af komparativ genomanalyse.

Først blev strategien beskrevet, efterfølgende blev den implementeret ved hjælp af eksisterende programmer og nyudviklede programmer. Til sidst blev pipelinen testet, for at se hvor effektiv den er.

En væsentlig ting, ved pipelinen var at den skulle være hurtig og fleksibel. Pipelinen er gjort hurtig ved at anvende en hurtig alignment program BLAT, ved at anvende DNA-sekvenser, hvor repeat regionerne er maskeret og ved at maskerer exon for protein kodende gener og ikke protein-kodende gener.

Pipelinen er gjort fleksibel i den forstand, at den skal være uafhængig af eksisterende alignments. Så det er muligt selv at vælge hvilke arter, som pipelinen skal anvendes på. Pipelinen blev også gjort fleksibel, ved at kunne håndtere filer med kun en FASTA-sekvens i og filer med flere FASTA-sekvenser som f.eks. fugu-scaffolds, der alle ligger i en FASTA-fil. Den er gjort hurtig ved at anvende en hurtig alignment program BLAT,

Ved at anvende BLAT som alignment program, sættes der en grænse for hvor stor den evolutionære afstand må være mellem arterne, for at der kan opnås en ordentlig resultat. At vælge de tre arter menneske, mus og rotte, viste sig at være et fornuftigt valg, hvorimod at anvende zebrafisk i stedet for rotten, ville være en dårlig ide. Hvis der ønskes en større evolutionær afstand mellem arterne, er det let at skifte til mere sensitive alignment programmer som BLASTz eller BLATz, da pipelinen anvender axt formatet som input fra alignment programmet. Dette er også med til at gøre pipelinen fleksibel.

Der bør ikke anvendes DNA-sekvenser, hvor repeat regionerne ikke er maskerede, da RNAz og RNAmicro ofte vil klassificere dem som henholdsvis ncRNA og miRNAer.

Effektiviteten af pipelinen er blevet bestemt til at være 65%, med hensyn til at finde de miRNAer, der er i testmaterialet. Ses der kun på de miRNAer, der ikke er maskeret og har homologe sekvenser i alle tre arter, er effektiviteten 91%.

Der blev fundet en række nye miRNAer, i forhold til miRBase release 9.0, nogle af disse er også fundet af andre, da de er kommet med i miRBase release 11.0.

I forbindelse med testen af pipelinen, blev der også fundet 14 alignments, der blev klassificeret som miRNAer, og hvor ingen af sekvenserne der indgik var kendte miRNAer. Der har ikke været tid til en nærmere analyse af disse sekvenser, for at forsøge at fastslå om det er tale om reelle miRNA gener eller fejl klassifikationer. Ved at genteste disse kortere alignments med RNAmicro, var der kun seks alignments der stadig kunne klassificeres som miRNAer.

Den samme test blev foretaget på de alignments, som indeholdt kendte miRNAer. Det resulterede i, at to miRNAer blev ikke blev klassificeret som miRNAer.

8.1 Forbedringer til pipelinen

Der er et sted i pipelinen, der skal laves om. Det er hvor de tre sekvenser, der er blevet fundet ved de to BLAT alignments skal samles og placeres i en databasetabel. Sekvensen for den art, der ikke er med i den anden alignment, skal nogle gange forkortes. Den måde det gøre nu med BLATz bevirker, at der ikke placeres et resultat i databasetabellen. I stedet for BLATz, skal der udvikles et program, der anvender dynamisk programmering, og hvor det ikke koster noget at placere gaps i enderne af den korteste sekvens.

Parametrene som RNAz og RNAmicro anvender er hardkodet ind i pipeline programmerne, de burde gives som parametre til pipeline programmerne.

De programmer, der anvender en lokal installation af miRBase databasen, bør ændres så det bliver lettere at skifte mellem forskellige lokale installationer af miRBase.

A Fil formater

Her beskrives de fil-formater, der er benyttet i forbindelse med pipelinen.

A.1 FASTA-format

FASTA-formatet er det format, som DNA-sekvenserne der anvendes i dette projekt findes på.

Et eksempel på FASTA-format:

```
>mmu-12:109053353:109053403:+
GGAAGATGAGGTCATTAGGCTGATCACAATGGAGCAGGTCTCATTCTCATT
```

FASTA-formatet. Dette format har en meget simpel specifikation bestående af to dele: definitionslinien og sekvenslinierne.

Definitions linien er en enkelt linie, der starter med en obligatorisk større end ">" tegn efterfulgt af en identifier og derefter en beskrivelse. Der må ikke være et mellemrum mellem ">" og identifieren. Identifieren må heller ikke indeholder mellemrum, da det er afgrænsningen mellem identifier og beskrivelsen. Beskrivelsen må ikke indeholde ny linie tegnet "\n", ellers er der ingen begrænsning på hvilke tegn, der må bruges i beskrivelsen.

Sekvenslinierne kan indeholde enten DNA-, RNA- eller proteinsekvenser og har et meget simpelt format, et tegn for hver nukleotid eller aminosyre, og der er ingen begrænsning på deres længde og deres antal, men normalt er længden på hver linie 50 til 80 tegn.

En FASTA-fil kan indeholde en eller flere FASTA-sekvenser. Hver linie der starter med ">" indikerer starten på en ny sekvens. En FASTA-fil må ikke indeholde tomme linier. (Korf et al., 2003; Markel and León, 2003)

A.2 axt-format

Axt-formatet er det format, der benyttes i som output-format ved BLAT alignments i pipelinen.

Et eksempel på axt formatet:

```
8 hsa-17:27691241:27711334:+ 4239 4288 mmu-11:35615819:35635904:+ 1093 1141 + 3544
gtggctcacgcctgtaatcccagcactttgggaggccgaggcaggcggat
gtggcgcacgcctttaatcccagca-tttggaaggcagaggcaggcggat
```

I en fil på axt-format, fylder hver alignment tre linier og hver alignment er adskilt med en tom linie:

Linie 1: Informationslinien
Linie 2: Database alignment sekvens med gaps
Linie 3: Query alignment sekvens med gaps

Informationslinien indeholder ni felter adskilt med blank tegn. Felterne har følgende betydning:

1) Alignment nummer, starter fra 0
2) Database identifier
3) Database alignmentets startposition i database input sekvensen, hvor første base har nummer 1
4) Database alignmentets slutposition i database input sekvensen, slut basen er inkluderet
5) Query identifier
6) Query alignmentets startposition i query input sekvensen
7) Query alignmentets slutposition i query input sekvensen
8) Query streng, hvis den er '-' er query alignmentets startposition og slutposition relativ til den omvendt komplementære query sekvens
9) Alignment score

A.3 Clustal-format

Clustal-formatet benyttes til multiple aligned sekvenser. Clustal-formatet består af en header linie efterfulgt af sekvens data i blokke.
Et eksempel på axt formatet:

```
CLUSTAL W (1.83) multiple sequence alignment


hsa-mir-9-1      CGGGGUUGGUUGUUAUCUUUGGUUAUCUAGCUGUAUGAGUGGUGUGGAGUCUUCAUAAAG 60
mmu-mir-9-1      CGGGGUUGGUUGUUAUCUUUGGUUAUCUAGCUGUAUGAGUGGUGUGGAGUCUUCAUAAAG 60
dre-mir-9-1      -GGGGUUGGCUGUUAUCUUUGGUUAUCUAGCUGUAUGAGUGUUAUUCAUUCUUCAUAAAG 59
fru-mir-9-1      -GGGGUUGUCUGUUAUCUUUGGUUAUCUAGCUGUAUGAGUGACGUACAAUCUUCAUAAAG 59
                  *******  ******************************   *  * ***********

hsa-mir-9-1      CUAGAUAACCGAAAGUAAAAAUAACCCCA 89
mmu-mir-9-1      CUAGAUAACCGAAAGUAAAAAUAACCCCA 89
dre-mir-9-1      CUAGAUAACCGAAAGUAACAAGAAUCCC- 87
fru-mir-9-1      CUAGAUAACCGAAAGUAACAAGAAUCCC- 87
                 ****************** ** ** ***
```

Første linie skal starte med CLUSTAL, resten er ikke obligatorisk, men beskriver gerne program "W" og versionsnummer "1.83". Første linie skal efterfølges af en eller flere tomme linier.

Derefter kommer en eller flere blokke af sekvens data. Hver blok indeholder en linie for hver sekvens i alignmentet. Hver linie starter med sekvens navn efterfulgt af sekvensen/delsekvensen og evt. til slut antallet af nukleotider i sekvensen/delsekvensen. Disse oplysninger er adskilt med blank tegn.

Under sekvenserne kan der være en linie, der angiver hvor konserverede de enkelte kolonner er. Symbolerne der anvendes er "*" der betyder fuld konserveret, ":" der betyder stærkt konserveret, "." der betyder svagt konserveret.

Hver blok er adskilt med en tom linie.

Litteratur

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402.

Ambros, V., Bartel, B., Bartel, D. P., Burge, C. B., Carrington, J. C., Chen, X., Dreyfuss, G., Eddy, S. R., Griffiths-Jones, S., Marshall, M., Matzke, M., Ruvkun, G., and Tuschl, T. (2003). A uniform system for microRNA annotation. *RNA*, 9:277–279.

Bartel, D. P. (2004). MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116:281–297.

Batzoglou, S. (2005). The many faces of sequence alignment. *Briefings in Bioinformatics*, 6:6–22.

Bentwich, I. (2005). Prediction and validation of microRNAs and their targets. *FEBS Letters*, 579:5904–5910.

Bentwich, I., Avniel, A., Karov, Y., Aharonov, R., Gilad, S., Barad, O., Barzilai, A., Einat, P., Einav, U., Meiri, E., Sharon, E., Spector, Y., and Bentwich, Z. (2005). Identification of hundreds of conserved and nonconserved human microRNAs. *Nature Genetics*, 37:766–770.

Berezikov, E., Guryev, V., van de Belt, J., Wienholds, E., Plasterk, R. H., and Cuppen, E. (2005). Phylogenetic shadowing and computational identification of human microRNA genes. *Cell*, 120:21–24.

Berezikov, E. and Plasterk, R. H. (2005). Camels and zebrafish, viruses and cancer: a microRNA update. *Human Molecular Genetics*, 14:R183–R190.

Berezikov, E., van Tetering, G., Verheul, M., van de Belt, J., van Laake, L., Vos, J. and Verloop, R., van de Wetering, M., Guryev, V., Takada, S., van Zonneveld, A. J., Mano, H., Plasterk, R., and Cuppen, E. (2006). Many novel mammalian microRNA candidates identified by extensive cloning and RAKE analysis. *Genome Research*, 16:1289–1298.

Boffelli, D., Nobrega, M. A., and Rubin, E. M. (2004). Comparative genomics at the vertebrate extremes. *Nature Reviews Genetics*, 5:456–465.

Bonnet, E., Wuyts, J., Rouze, P., and Van de Peer, Y. (2004). Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences. *Bioinformatics*, 20:2911–2917.

Borchert, G. M., Lanier, W., and Davidson, B. L. (2006). RNA polymerase III transcribes human microRNAs. *Nature Structural & Molecular Biology*, 13:1097–1101.

Borer, P. N., Dengler, B., Tinoco, I. J., and Uhlenbeck, O. C. (1974). Stability of ribonucleic acid double-stranded helices. *Journal og Molelular Biology*, 86:843–853.

Bray, N. and Pachter, L. (2004). MAVID: constrained ancestral alignment of multiple sequences. *Genome Research*, 14:693–699.

Brennecke, J., Stark, A., Russell, R. B., and Cohen, S. M. (2005). Principles of microRNA-target recognition. *PLoS Biol*, 3:e85.

Brudno, M., Do, C. B., Cooper, G. M., Kim, M. F., Davydov, E., NISC Comparative Sequencing Program, Green, E. D., Sidow, A., and Batzoglou, S. (2003). LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Research*, 13:721–731.

Cai, X., Hagedorn, C. H., and Cullen, B. R. (2004). Human microRNAs are processed from capped, polyadenylated transcripts that can also function as mRNAs. *RNA*, 10:1957–1966.

Chiaromonte, F., Weber, R. J., Roskin, K. M., Diekhans, M., Kent, W. J., and Haussler, D. (2003). The share of human genomic DNA under selection estimated from human-mouse genomic alignments. *Cold Spring Harbor Symposia on Quantitative Biology*, 68:245–254.

Coventry, A., Kleitman, D. J., and Berger, B. (2004). MSARI: multiple sequence alignments for statistical detection of RNA secondary structure. *Procedings of the National Academy of Science United States of America*, 101:12102–12107.

Cullen, B. R. (2004). Transcription and processing of human microRNA precursors. *Molecular Cell*, 16:861–865.

Dezulian, T., Remmert, M., Palatnik, J. F., Weigel, D., and Huson, D. H. (2005). Identification of plant microRNA homologs. *Bioinformatics*, 22:359–360.

di Bernardo, D., Down, T., and Hubbard, T. (2003). ddbRNA: detection of conserved secondary structures in multiple alignments. *Bioinformatics*, 19:1606–1611.

Doench, J. G. and Sharp, P. A. (2004). Specificity of microRNA target selection in translational repression. *Genes & Development*, 18:504–511.

Dowell, R. D. and Eddy, S. R. (2004). Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5:71.

Du, T. and Zamore, P. D. (2005). microPrimer: the biogenesis and function of microRNA. *Development*, 132:4645–4652.

Dwyer, R. A. (2003). *Genomic Perl from bioinformatics basics to working code*. Cambride University Press.

Eddy, S. R. (2004). What is dynamic programming? *Nature Biotechnology*, 22:909–910.

Frazer, K. A., Elnitski, L., Church, D. M., Dubchak, I., and Hardison, R. C. (2003). Cross-species sequence comparisons: a review of methods and available resources. *Genome Research*, 13:1–12.

Gardner, P. P. and Giegerich, R. (2004). A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5:140.

Ghosh, Z., Chakrabarti, J., and Mallick, B. (2007). miRNomics-the bioinformatics of microRNA genes. *Biochemical and Biophysical Research Communication*, 363:6–11.

Griffiths-Jones, S. (2004). The microRNA Registry. *Nucleic Acids Research*, 32:D109–D111.

Griffiths-Jones, S., Grocock, R. J., van Dongen, S., Bateman, A., and Enright, A. J. (2006). miRBase: microRNA sequences, targets and gene nomenclature. *Nucleic Acids Research*, 34:D140–D144.

Griffiths-Jones, S., Saini, H. K., van Dongen, S., and Enright, A. J. (2008). miRBase: tools for microRNA genomics. *Nucleic Acids Research*, 36:D154–D158.

He, L. and Hannon, G. J. (2004). MicroRNAs: small RNAs with a big role in gene regulation. *Nature Reviews Genetics*, 5:522–531.

Helvik, S. A., Snøve, O. J., and Sætrom, P. (2007). Reliable prediction of Drosha processing sites improves microRNA gene prediction. *Bioinformatics*, 23:142–149.

Hertel, J. and Stadler, P. F. (2006a). Hairpins in a haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics*, 22:e197–e202.

Hertel, J. and Stadler, P. F. (2006b). RNAmicro: upgraded version 1.1. Documentation.

Higgs, P. G. (2000). RNA secondary structure: physical and computational aspects. *Quartely Reviews of Biophysics*, 33:199–253.

Hofacker, I. L. (2003). Vienna RNA secondary structure server. *Nucleic Acids Research*, 31:3429–3431.

Hofacker, I. L., Fekete, M., and Stadler, P. F. (2002). Secondary structure prediction for aligned RNA sequences. *Journal of Molecular Biology*, 319:1059–1066.

Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatshefte fur Chemie*, 125:167–188.

Hutvagner, G., McLachlan, J., Pasquinelli, A. E., Balint, E., Tuschl, T., and Zamore, P. D. (2001). A cellular function for the RNA-interference enzyme Dicer in the maturation of the let-7 small temporal RNA. *Science*, 293:834–838.

Jensen, P. and Prentø, P. (2003). *Cellebiologi. Cellens organisation og livsprocesser*. Gads Forlag, 2. edition.

Kent, W., Baertsch, R., Hinrichs, A., Miller, W., and Haussler, D. (2003). Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. *Proceedings of the National Academy of Science of the United States of America*, 100:11484–11489.

Kent, W. J. (2002). BLAT–the BLAST-like alignment tool. *Genome Research*, 12:656–664.

Khvorova, A., Reynolds, A., and Jayasena, S. D. (2003). Functional siRNAs and miRNAs exhibit strand bias. *Cell*, 115:209–216.

Kim, V. N. (2005). MicroRNA biogenesis: coordinated cropping and dicing. *Nature Reviews Molecular Cell Biology*, 6:376–385.

Kjeldsen, E. and Nørby, S. (2006). Menneskets genom. In Nørby, S. and Nielsen, P. K. A., editors, *Medicinsk Genetik*, chapter 1, pages 13–50. FADL.

Knudsen, B. and Hein, J. (1999). RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, 15:446–454.

Knudsen, B. and Hein, J. (2003). Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, 31:3423–3428.

Korf, I., Yandell, M., and Bedell, J. (2003). *BLAST*. O'Reilly.

Krol, J., Sobczak, K., Wilczynska, U., Drath, M., Jasinska, A., Kaczynska, D., and Krzyzosiak, W. J. (2004). Structural features of microRNA (miRNA) precursors and their relevance to miRNA biogenesis and small interfering RNA/short hairpin RNA design. *The Journal of Biological Chemistry*, 279:42230–42239.

Lagos-Quintana, M., Rauhut, R., Lendeckel, W., and Tuschl, T. (2001). Identification of novel genes coding for small expressed RNAs. *Science*, 294:853–858.

Lau, N. C., Lim, L. P., Weinstein, E. G., and Bartel, D. P. (2001). An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans. *Science*, 294:858–862.

Lee, R. C. and Ambros, V. (2001). An extensive class of small RNAs in Caenorhabditis elegans. *Science*, 294:862–864.

Lee, R. C., Feinbaum, R. L., and Ambros, V. (1993). The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *Cell*, 75:843–854.

Lee, Y., Ahn, C., Han, J., Choi, H., Kim, J., Yim, J., Lee, J., Provost, P., Rådmark, O., Kim, S., and Kim, V. N. (2003). The nuclear RNase III Drosha initiates microRNA processing. *Nature*, 425:415–419.

Lee, Y., Kim, M., Han, J., Yeom, K. H., Lee, S., Baek, S. H., and Kim, V. N. (2004). MicroRNA genes are transcribed by RNA polymerase II. *The EMBO Journal*, 23:4051–4060.

Lewis, B. P., Burge, C. B., and Bartel, D. P. (2005). Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell*, 120:15–20.

Lim, L. P., Glasner, M. E., Yekta, S., and Burge, C. B. Bartel, D. P. (2003a). Vertebrate microRNA genes. *Science*, 299:1540.

Lim, L. P., Lau, N. C., Weinstein, E. G., Abdelhakim, A., Yekta, S., Rhoades, M. W., Burge, C. B., and Bartel, D. P. (2003b). The microRNAs of Caenorhabditis elegans. *Genes & Development*, 17:991–1008.

Lindow, M. and Gorodkin, J. (2007). Principles and limitations of computational microRNA gene and target finding. *DNA and Cell Biology*, 26:339–351.

Lindow, M., Jacobsen, A., Nygaard, S., Mang, Y., and Krogh, A. (2007). Intragenomic matching reveals a huge potential for miRNA-mediated regulation in plants. *PLoS Computational Biology*, 3:e238.

Liu, J., Valencia-Sanchez, M. A., Hannon, G. J., and Parker, R. (2005). MicroRNA-dependent localization of targeted mRNAs to mammalian P-bodies. *Nature Cell Biology*, 7:719–723.

Liu, X., Fortin, K., and Mourelatos, Z. (2008). MicroRNAs: biogenesis and molecular functions. *Brain Pathology*, 18:113–121.

Ma, B., Tromp, J., and Li, M. (2002). PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 8:440–445.

Margulies, E. H. and Birney, E. (2008). Approaches to comparative sequence analysis: towards a functional view of vertebrate genomes. *Nature Reviews Genetics*, 9:303–313.

Markel, S. and León, D. (2003). *Sequence analysis in a nutshell : a guide to common tools and databases*. O'Reilly.

Mathews, D. H. (2006a). Predicting RNA secondary structure by free energy minimization. *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*, 116:160–168.

Mathews, D. H. (2006b). Revolutions in RNA secondary structure prediction. *Journal of Molecular Biology*, 359:526–532.

Mathews, D. H., Disney, M. D., Childs, J. L., Schroeder, S. J., Zuker, M., and Turner, D. H. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Science of the United States of America*, 101:7287–7292.

Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940.

Mathews, D. H., Schroeder, S. J., Turner, D. H., and Zuker, M. (2006). Predicting RNA secondary structure. In Gesteland, R. F., Cech, T. R., and Atkins, J. F., editors, *The RNA world : the nature of modern RNA suggests a prebiotic RNA*, chapter 22, pages 631–657. Cold Spring Harbor Laboratory Press, Third edition.

Mathews, D. H. and Turner, D. H. (2006). Prediction of RNA secondary structure by free energy minimization. *Current Opinion in Structual Biology*, 16:270–278.

Mattick, J. S. (2007). A new paradigm for developmental biology. *Journal of Experimental Biology*, 210:1526–1547.

Mattick, J. S. and Makunin, I. V. (2006). Non-coding RNA. *Human Molecular Genetics*, 15:R17–R29.

Miller, W., Makova, K. D., Nekrutenko, A., and Hardison, R. C. (2004). Comparative genomics. *Annual Review of Genomics and Human Genetics*, 5:15–56.

Mount, D. W. (2001). *Bioinformatics: sequence and genome analysis*. Cold Spring Harbor Laboratory Press.

Needleman, S. B. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453.

NG Kwang Loong, S. and Mishra, S. K. (2007). De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics*, 23:1321–1330.

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, 24:1565–1567.

Nussinov, R. and Jacobson, A. B. (1980). Fast algorithm for predicting the secondary structure of single-stranded rna. *Proceedings of the National Academy of Science of the United States of America*, 77:6903–6913.

Nørby, S. (2003). Nukleinsyrernes struktur og funktion. In Kielberg, V., Nørby, S., and Rasmussen, L., editors, *DNA og RNA : en håndbog*, pages 27–44. Gad, 1 edition.

Pasquinelli, A. E., Reinhart, B. J., Slack, F., Martindale, M. Q., Kuroda, M. I., Maller, B., Hayward, D. C., Ball, E. E., Degnan, B., Muller, P., Spring, J., Srinivasan, A., Fishman, M., Finnerty, J., Corbo, J., Levine, M., Leahy, P., Davidson, E., and Ruvkun, G. (2000). Conservation of the sequence and temporal expression of let-7 heterochronic regulatory RNA. *Nature*, 408:86–89.

Pearson, W. R. (2000). Flexible sequence similarity searching with the FASTA3 program package. *Methods in Molecular Biology*, 132:185–219.

Pearson, W. R. (2001). Protein sequence comparison and protein evolution. Tutorial - ISMB2000.

Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Science of the United States of America*, 85:2444–2448.

Pennacchio, L. A. and Rubin, E. M. (2003). Comparative genomic tools and databases: providing insights into the human genome. *Journal of Clinical Investigation*, 111:1099–1106.

Reinhart, B. J., Slack, F. J., Basson, M., Pasquinelli, A. E., Bettinger, J. C., Rougvie, A. E., Horvitz, H. R., and Ruvkun, G. (2000). The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans. *Nature*, 403:901–906.

Reinhart, B. J., Weinstein, E. G., Rhoades, M. W., Bartel, B., and Bartel, D. P. (2002). MicroRNAs in plants. *Genes & Development*, 16:1616–1626.

Rhoades, M. W., Reinhart, B. J., Lim, L. P., Burge, C. B., Bartel, B., and Bartel, D. P. (2002). Prediction of plant microRNA targets. *Cell*, 110:13–520.

Rivas, E. and Eddy, S. R. (2001). Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics*, 2:8.

Rodriguez, A., Griffiths-Jones, S., Ashurst, J. L., and Bradley, A. (2004). Identification of mammalian microRNA host genes and transcription units. *Genome Research*, 14:1902–1910.

Sankoff, D. (1985). Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45:810–825.

Schwartz, S., Kent, W. J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R. C. Haussler, D., and Miller, W. (2003). Human-mouse alignments with BLASTZ. *Genome Research*, 13:103–107.

Seitz, H., Royo, H., Bortolin, M. L., Lin, S. P., Ferguson-Smith, A. C., and Cavaille, J. (2004). A large imprinted microRNA gene cluster at the mouse Dlk1–Gtl2 domain. *Genome Research*, 14:1741–1748.

Sewer, A., Paul, N., Landgraf, P., Aravin, A., Pfeffer, S., Brownstein, M. J., Tuschl, T., van Nimwegen, E., and Zavolan, M. (2005). Identification of clustered microRNAs using an ab initio prediction method. *BMC Bioinformatics*, 7:267.

Sheng, Y., Engstrom, P. G., and Lenhard, B. (2007). Mammalian microRNA prediction through a support vector machine model of sequence and structure. *PLoS ONE*, 2:e946.

Slowinski, J. B. (1998). The number of multiple alignments. *Molecular Phylogenetics and Evolution*, 10:264–266.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:197–197.

Sun, Y. F. and Fan, X. D. amd Li, Y. D. (2003). Identifying splicing sites in eukaryotic RNA: support vector machine approach. *Computers in Biology and Medicine*, 33:17–29.

Szymanski, M., Barciszewska, M. Z., Erdmann, V. A., and Barciszewski, J. (2005). A new frontier for molecular medicine: noncoding RNAs. *Biochimica et Biophysica Acta – Reviews on Cancer*, 1756:65–75.

Sætrom, P. and Snøve, O. J. (2007). Robust machine learning algorithms predict microRNA genes and targets. *Methods in Enzymology*, 427:25–49.

Tinoco, I. J. and Bustamante, C. (1999). How RNA folds. *Journal of Molecular Biology*, 293:271–281.

Ureta-Vidal, A., Ettwiller, L., and Birney, E. (2003). Comparative genomics: genome-wide analysis in metazoan eukaryotes. *Nature Reviews Genetics*, 4:251–262.

Washietl, S. (2006). *RNAz 1.0 Predicting structural noncoding RNAs*. University Vienna.

Washietl, S., Hofacker, I. L., and Stadler, P. F. (2005). Fast and reliable prediction of noncoding RNAs. *Proceedings of the National Academy of Science of the United States of America*, 102:2454–2459.

Weber, M. J. (2005). New human and mouse microRNA genes found by homology search. *FEBS Journal*, 272:59–73.

Wienholds, E. and Plasterk, R. H. (2005). MicroRNA function in animal development. *FEBS Letters*, 579:5911–5922.

Wightman, B., Ha, I., and Ruvkun, G. (1993). Posttranscriptional regulation of the heterochronic gene lin-14 by lin-4 mediates temporal pattern formation in C. elegans. *Cell*, 75:855–862.

Woese, C. R., Kandler, O., and Wheelis, M. L. (1990). Towards a natural system of organisms: Proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Science of the United States of America*, 87:4576–4579.

Woodson, S. A. (2000). Recent insights on RNA folding mechanisms from catalytic RNA. *Cellular and Molecular Life Science*, 57:796–808.

Xia, T., SantaLucia, J. J., Burkard, M. E., Kierzek, R., Schroeder, S. J., Jiao, X., Cox, C., and Turner, D. H. (1998). Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, 37:14719–14735.

Xue, C., Li, F., He, T., Liu, G. P., Li, Y., and Zhang, X. (2005). Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, 6:310.

Yang, Z. R. (2004). Biological applications of support vector machines. *Briefings in Bioinformatics*, 5:328–338.

Yi, R., Qin, Y., Macara, I. G., and Cullen, B. R. (2003). Exportin-5 mediates the nuclear export of pre-microRNAs and short hairpin RNAs. *Genes & Development*, 17:3011–3016.

Zhang, B. H., Pan, X. P., Wang, Q. L., Cobb, G. P., and Anderson, T. A. (2005). Identification and characterization of new plant microRNAs using est analysis. *Cell Research*, 15:336–360.

Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31:3406–3415.

Zuker, M., Mathews, D. H., and Turner, D. H. (1999). Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide. In Barciszewski, J. and Clark, B. F. C., editors, *RNA Biochemistry and Biotechnology*, pages 11–43. Kluwer Academic Publishers.

Zuker, M. and Sankoff, D. (1984). RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46:591–621.

Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9:133–148.