ПЛОВДИВСКИ УНИВЕРСИТЕТ „ПАИСИЙ ХИЛЕНДАРСКИ”
БИОЛОГИЧЕСКИ ФАКУЛТЕТ

КАТЕДРА”ФИЗИОЛОГИЯ НА РАСТЕНИЯТА И МОЛЕКУЛЯРНА БИОЛОГИЯ”

Георги Иванов Минков

АВТОРЕФЕРАТ

на дисертационен труд за придобиване на образователна и научна степен “доктор”

ТЕМА:

„Биоинформатични методи и софтуерни решения за *de novo* идентификация и анализ на растителни миРНК гени чрез данни от масово паралелно секвениране“

Област на висше образование: 4. Природни науки, математика и информатика

Професионално направление: 4.3. Биологически науки

Докторска програма: Биоинформатика

Научен ръководител: Доц. д-р Веселин Баев

Пловдив, 2017

Дисертантът е докторант на самостоятелна подготовка към катедра „Физиология на растенията и молекулярна биология“, ПУ „Паисий Хилендарски“

Дисертационният труд е обсъден и предложен за защита на заседание на катедра „Физиология на растенията и молекулярна биология“ при Биологическия факултет на Пловдивския университет „П.Хилендарски“, проведено на 10.10. 2017г.

Защитата на дисертационния труд ще се състои на 07.10.2017г. от 11.00 часа в гр. Пловдив, Биологически факултет на Пловдивския университет „П.Хилендарски“ на открито заседание на научното жури.

1. **Увод**

МикроРНК (миРНК) са клас малки, ендогенни, некодиращи РНК с важна роля в почти всички биологични процеси. миРНК контролират пост-транскрипциония сайлънсинг на много гени и регулират много ключови процеси, включително развитие, растеж и отговор към стрес. миРНК в растенията се изследват от преди десетина години. Въпреки това, само някои техни аспекти – преди всичко тяхната биогенеза – са добре познати.

По-пълно познание на функцията на миРНК би позволило на учените да контролират генната експресия в живи клетки чрез екзогенни методи. Изследването на миРНК също така може да ни даде повече информация за различни еволюционни събития. По този начин можем да придобием по-добра представа за важни етапи от еволюцията, като например появата на сухоземните растения.

Настоящият дисертационен труд е добър пример за многопрофилния подход в сферата на биоинформатиката. В търсене на по-добро теоретично разбиране на миРНК биогенезата, тук бяха използвани серия биологични, информатични и математични подходи. Основните данни са резултат от масово паралелно секвениране, обработени със съществуващи online и локални биоинформатични софтуерни продукти и биологични бази данни.

В настоящия дисертационен труд е разработен нов софтуерен пакет за идентификация на растителни миРНК гени, наречен miRDEG. В неговата разработка се използват два езика за програмиране, теория на обектно-ориентирано програмиране, теория и оптимизация на математически графи, както и иновативен подход към обработка на РНК вторична структура.

miRDEG използва широк диапазон от характеристики за de novo идентификация на миРНК. Кандидат миРНК могат да се търсят както без вторична структура, така и без консервативност в други, неизследвани видове. miRDEG работи сравнително бързо, използва се лесно и поддържа серия операционни системи, поне частично. В процеса на разработка бяха създадени и няколко помощни модули с по-широко биоинформатично приложение.

Голямото предимство на miRDEG е неговият обектно-ориентиран подход. В биоинформатиката, ООП най-често се използва за създаване на графични интерфейси (където няма практична алтернатива). miRDEG използва обектно-ориентиран работен програмен код, както и серия обектно-ориентирани структури от данни. Това му дава високо ниво на модуларност, позволяващо добавянето на нови филтри, премахването на стари такива, пълна промяна на метода на филтриране и т.н. За това е възможно да се извадят цели вътрешни модули от miRDEG, които да се използват в изцяло нови програми без допълнителна обработка.

По време на създаването си miRDEG се промени драстично с добавянето на нови филтри. С нарастващото ни разбиране за миРНК биогенезата, софтуерният пакет има капацитета да продължи да се променя. За първи път с него са намерени серия нови миРНК в геномите на Brachypodium distachyon и Solanum lycopersicum. С допълнителни настройки на входните данни и създаването на нови модули, много вероятно е да се намерят и други нови миРНК в други видове. Важна и уникална особеност на софтуера е, че интегрира два типа данни от масово-паралелно секвениране – малки РНК и деградомни секвенции. Чрез анализа на деградомните секвенции, miRDEG успешно може да разграничи реално експресираните миРНК локуси, при условие, че дадена миРНК е дуплицирана в няколко геномни локуса. Този подход за идентификация на нови растителни миРНК гени се използва за първи път, което прави miRDEG уникален по своята същност, както и универсален за растителните видове.

2. **Цели и задачи**

Целта на настоящия дисертационен труд е да се разработи софтуер за идентификация и анализ на растителни миРНК гени, базиран на интеграция на данни от масово-паралелно секвениране на малки РНК и деградомни секвенции.

Така поставена цел определя следните основни задачи:

1. Да се разработи алгоритъм за предвиждане на растителни миРНК гени, базиран на клъстериране и анализ на малки РНК от масово-паралелно секвениране.
2. Прилагане на разработения алгоритъм за предвиждане на нови миРНК в генома на *Brachypodium distachyon.*
3. Разработване на софтуерен пакет за предвиждане на растителни миРНК, базиран на интеграция на данни от масово-паралелно секвениране на малки РНК и деградомни секвенции.
4. Разработване на модул за анотиране на РНК фуркетни структури чрез методите на ООП.
5. Прилагане на разработения софтуерен пакет за предвиждане на миРНК в генома на *Solanum lycopersicum.*

3. Материали и методи

3.1. Данни от масово-паралелно секвениране на малки РНК

В дисертационния труд са използвани малки РНК от *Brachypodium distachyon* от публично достъпната база данни NCBI GeneBank – Gene Expression Omnibus (GEO) със следните номера на достъп – GSM506620, GSM506621, GSM406303, GSM406302.

За данни от малки РНК както и деградомни секвенции на *S. lycopersicum* бяха използвани публично достъпни библиотеки както следва:

- GSM452714, GSM452715, GSM452716, GSM452717, GSM452718, GSM452719, GSM452720, GSM452721 (малки РНК)
- GSM1047560, to GSM1047562, GSM1047561, GSM10475630 (деградомни секвенции)

3.2. Външни бази данни

3.2.1. База данни SOL Genomics Network

3.2.2. База данни за миРНК – mirBase

3.3. Външни софтуерни продукти

3.3.1. Пакет от софтуерни продукти Bowtie (картиране на секвенции, мапинг)

3.3.2. fastaFromBed, (софтуерен пакет BEDTools)

3.3.3. Софтуер за вторична РНК структура Vienna RNA

3.3.4. Платформа за контрол на работния процес Galaxy

3.3.5. Среда за софтуерна разработка Eclipse

3.4. Информатични и математични инструменти

3.4.1. Език за програмиране Java

3.4.2. Език за програмиране Perl

3.4.3. Теория на обектно-ориентираното програмиране

3.4.4. Теория на математическите графи

3.5. Използван хардуер

Сървърни платформи: Dell 48 Core, 128 GB RAM, HDD Array 24TB, CentOS

Десктоп платформи: Intel Core i7-4770 @ 3.40 Ghz, 16 GB RAM, Windows 7 Professional

4. **Резултати и обсъждане**

В настоящата дисертация са разработени следните софтуерни приложения:

Clustering module е самостоятелна програма, предназначена за идентификация на растителни миРНК чрез изследване на резултати от масово-паралелно секвениране (на малки РНК) и представлява предшественик на програмата *clusterSD*.

clusterSD, miRNA2D и miCompare са три отделни програми, събрани заедно в общият софтуерен пакет **miRDEG** за идентификация на растителни миРНК чрез анализиране на резултати от масово-паралелно секвениране като се интегрират два типа данни – малки РНК и дргрaдомни секвенции.

LoopRNA е самостоятелен Java модул, предназначен да моделира вторичната структура на РНК секвенции в паметта чрез теориите на обектно-ориентираното програмиране и математическите графи, нужен за изследването на 2D РНК структурата на предполагаемите миРНК прекурсори и представлява част от програмата *miRNA2D.*

4.1. Clustering Module

Clustering Module е първата разработена Java програма за изследването и анализ на данни от секвенирани малки РНК и клъстерирането им при картирането на секвенциите с цел предвиждане на локуси за потенциални нови миРНК. Тя представлява първоначална експериментална стъпка в прилагане на методите на ООП към биоинформатични изследвания.

Clustering Module се фокусира върху намирането на „двойки“ от „клъстери“ от малки РНК рийдове, които съвпадат с дистрибуцията от остатъци от миРНК биогенеза, като инструмент за идентификация на нови миРНК. Тук „клъстер“ се дефинира като последователност от малки РНК рийдове, отговарящи на дадени характеристики, а „двойка“ се дефинира като просто два такива клъстера, отговарящи на дадени характеристики, които се задават от потребителя.

Такива клъстери (и по-конкретно двойки от клъстери) се формират най-често когато всички isomiR на даден миРНК локус се картират към референтния геном. Това е една от ключовите характеристики на миРНК, която може да се използва, за да се предвиждат миРНК гени/локуси, като се изследва модела на картиране на малките РНК, получени от библиотеката от МПС (Фигура 1). Целта на Clustering Module е да “прихване” точно този профил на картирани секвенции, с които се характеризират миРНК гените и по този начин да могат да се предвидят *de novo* миРНК локуси в растителния геном.

Програмата няма графичен интерфейс и се изпълнява директно от командния ред. Clustering Module изисква точно 5 настройки на потребителя, които контролират изпълнението на самата програма. Техните стойности се задават като допълнителни конзолни параметри при изпълнението на програмата. Те се задават директно като стойности и техният ред е фиксиран. Всички конзолни параметри са задължителни и нямат стойности по подразбиране.

Кратко описание на тези конзолни параметри е показано на Таблица 1.

Поредност на параметъра	Описание
1	Име на файла с входните данни
2	Максимален размер на клъстер
3	Минимално разстояние между клъстерите
4	Минимално разстояние между двойка и фланкиращите я клъстери
5	Максимален размер на двойка

Таблица 1. Списък от конзолните параметри на програмата Clustering Module

4.1.1. Зареждане на входните данни

При стартиране, програмата Clustering Module изисква точно един файл с входни данни, подаден като първи параметър. Това е файлът с координати на картираните малки РНК рийдове, които ще се подреждат в клъстери и после в двойки клъстери. От файла се очаква да е в GFF формат, като неговото съдържание трябва да е сортирано първо по начални координати на рийда, после по крайни координати.

Програмата Clustering Module не изисква изрично задаване на файл за изписване на изходните данни. Неговото име и местонахождение се определя автоматично.

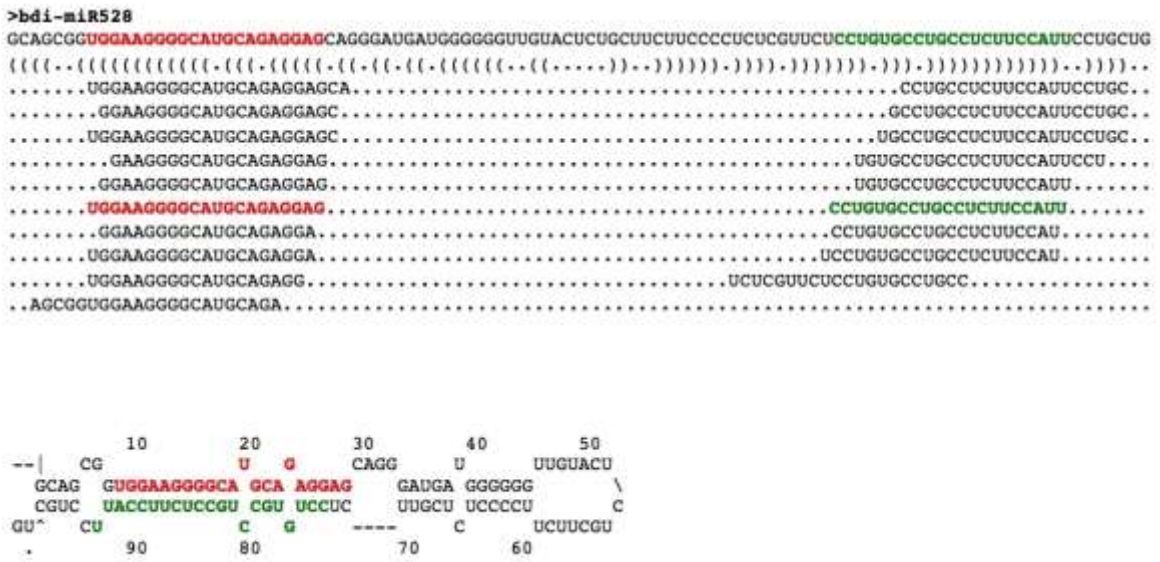
4.1.2. Създаване на списък с предварителни клъстери

След зареждане на входните данни като списък в паметта, в него започват да се търсят клъстери. Тук клъстер се дефинира като последователност от малки РНК рийдове, чийто общ размер не надхвърля дадена максимална стойност, без да се взима предвид тяхното застъпване (overlap) или разстоянието между тях. Максималният размер на клъстер се подава като втори конзолен параметър при стартиране на програмата.

Техническото търсене се извършва като един рийд се приема за предполагаемо начало на клъстер и се сравнява със следващите рийдове в списъка с входните данни. Когато се намери рийд, който прави клъстера прекалено голям, то този рийд се пропуска и търсенето спира до там. След това се търси от началото на клъстера назад към началото на списъка със същото условие за край. Така полученият клъстер се записва в списък с клъстери. Следващият рийд след края на клъстера служи за предполагаемо начало на следващия клъстер.

4.1.3. Филтриране на предварителните клъстери

Понеже единствения критерий за предварителен клъстер е неговият максимален размер, методът за откриване на клъстерите създава много припокриващи се клъстери. Те не биха могли да образуват допустими двойки и трябва да бъдат премахнати. Тази функция изпълнява първата стъпка за филтриране – филтриране по overlap. В тази стъпка, всички припокриващи се клъстери се премахват. Премахват се и



Фигура 1. Схематична фигура на миРНК bdi-mir528 и нейните изомиР секвенции, картирани върху прекурсора.

неприпокриващи се клъстери, които са по-близо един от друг от минималното разстояние между клъстерите. Това разстояние се задава като трети конзолен параметър при стартиране на програмата.

Технически, това наподобява търсенето на предварителни клъстери. Алгоритмът избира даден клъстер и започва да го сравнява с всички следващи, докато се намери клъстер, който е на достатъчно голямо разстояние от предишния. Това формира група от клъстери, които са прекалено нагъсто и не отговарят на миРНК биогенезата. Всички тези клъстери се премахват от списъка.

След този филтър, неприпокриващите се клъстери се проверяват за посока на секвенцията, + или - спрямо 3‘ края. От клъстера се очаква всички рийдове в него да са в една и съща посока. Ако клъстерът е съставен от повече от 2 рийда, то се позволява най-много един да е в обратна посока. Клъстери, които имат повече от 1 рийд и в двете посоки се премахват от списъка. Клъстери в + посока се маркират с 1 ако няма рийдове в отрицателна посока и 2 ако има точно един. Клъстери в - посока се маркират с -1 ако няма рийдове в положителна посока и -2 ако има.

4.1.4. Създаване на двойки от клъстери

След като програмата се е подсигурила, че в списъка с клъстерите има само истински клъстери, тя започва да търси подходящи двойки между тях, конкретно между всеки два последователни клъстера. От двойките се изисква да не надвишават даден максимален размер, да са на поне дадено минимално разстояние от фланкиращите ги клъстери и клъстерите в тях да са еднопосочни. Минималното разстояние до фланкиращите клъстери се задава с четвъртия конзолен параметър, а максималния размер на двойката – с петия.

По конкретно, алгоритмът сравнява всеки клъстер с неговия следващ, като образува временна двойка. Първо се проверява разстоянието между началните координати на първия клъстер и крайните координати на втория. Ако то е прекалено голямо, то тази двойка се отхвърля. След това се проверява разстоянието между началните координати на първия клъстер в двойката и крайните координати на предшестващия го клъстер в списъка, ако такъв съществува. Ако разстоянието е прекалено малко, то двойката се отхвърля. Същата проверка се извършва и между крайните координати на втория клъстер от двойката и последяващия го клъстер в списъка, ако такъв съществува. Ако разстоянието е прекалено малко, двойката се отхвърля.

След това, двойката се проверява за посока. Ако посоките на двата клъстера са различни, двойката се отхвърля. Ако посоките им са еднакви, се извършва допълнителна проверка. Само на един от двата клъстера се позволява да има обратен елемент. Дори и двата клъстера да са еднопосочни, ако и двата имат по един обратен елемент, то двойката се отхвърля.

Ако дадена двойка премине през всички тези проверки, то тя се записва в списъка с двойките.

4.1.5. Изписване на резултатите

След всички филтри, в списъка с двойките остават само такива, които съвпадат с остатъците от биогенезата на миРНК и представляват кандидат миРНК. Те се изписват във файл с изходни данни в GFF формат. Името на изходния файл се генерира като се вземе името на входния файл до първата точка и към него се прикачи „*_output.gff*“.

В първата колона се поставя името на първата секвенция от първия клъстер в двойката, взета от файла с входните данни. В третата колона се поставя думата „*cluster*“. В четвъртата и петата колони се поставят началните координати на първия

клъстер и крайните координати на втория клъстер в двойката, взети от файла с входните данни. В седмата колона се поставя посоката на двойката клъстери, + или -. В деветата колона се поставя думата „*target*“ последвана от броя рийдове в първия клъстер и броя рийдове във втория клъстер от двойката

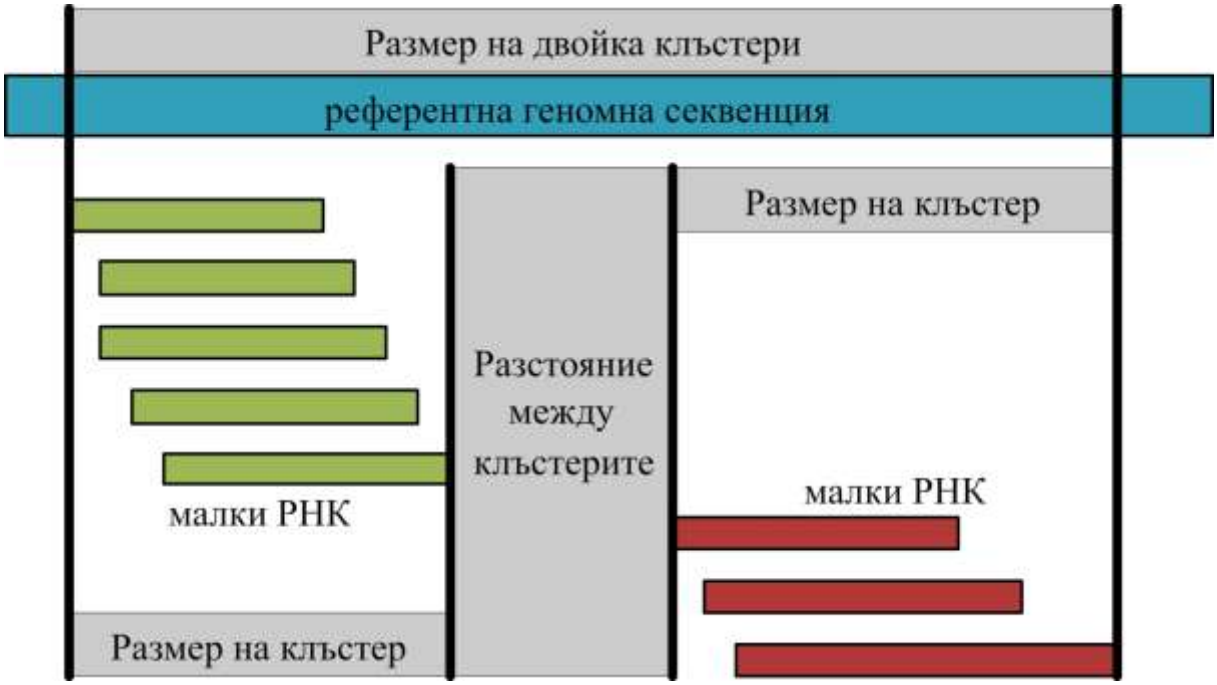
4.1.6. Предвиждане на миРНК в генома на *Brachypodium*

В настоящата дисертация бяха картирани четирите публично-достъпни библиотеки от малки РНК в *Brachypodium* (GSM506620; GSM506621, GSM406303 и GSM406302) към генома на *Brachypodium (v1.0).* Картирането се извърши с програмата Bowtie с опция, която не допуска несъвпадения в секвенцията. Изходният файл беше анализиран с Clustering module. Опциите, които бяха използвани при анализа с програмата, бяха следните:

- максималният размер на клъстера беше 65 нт
- минималното разстояние между два клъстера беше 15 нт
- минималното разстояние между клъстерите в двойка (т.е. разстоянието между кандидат миРНК и миРНК*) беше 15 нт
- максималният размер на цялата двойка беше 350 нт – максималната дължина, известна за *Arabidopsis* (Reinhart et al. 2002; Xie 2005).

Резултатите, генерирани от Clustering module от четирите библиотеки представляват геномни области, фланкирани от двата клъстера (5' и 3'). Те бяха комбинирани в 30 355 резултантни уникални локуса.

Clustering module филтрира локуси, включващи клъстери, които съдържат повече от една малка РНК в обратна посока на геномната секвенция. Такива малки РНК могат да се получат от повтарящи се региони, като например тандемни повторения и транспозони, което може да доведе до грешна идентификация на малки РНК като миРНК (Meyers et al. 2008). След проверка на резултатите за дадения локус за РНК вторична структура и коректно формиране на дуплекс миРНК:миРНК*, чрез използване на софтуерните пакети mirplan и duplex бяха филтрирани 2658 кандидат прекурсорни локуси с малки РНК клъстери, които дадоха 102 предполагаеми миРНК



Фигура 3. Схема на работния подход.

локуси за генома на *Brachypodium*. Резултатите са представени в Таблица 2 и Таблица 3. На Фигура 2 е показана схемата на работния подход.

Към днешна дата за *Brachypodium* са известни множество консервативни миРНК (Unver & Budak 2009; Wei et al. 2009; Zhang et al. 2009). Всички консервативни миРНК, познати до момента, са били идентифицирани по хомология, като се сравняват с малки РНК със секвенции от miRBase. Малко от тях се свързват с геномните локуси, от които са произлезли. Понеже са избрани от библиотека с малки РНК, много от докладваните миРНК секвенции могат да произлязат от множество локуси в генома. Поради това е възможно някои от тях да са фалшиви положителни резултати, а не истински миРНК. Нашият подход тук минимизира фалшивите положителни резултати, защото Clustering module елиминира секвенциите, произхождащи от локуси, които могат да синтезират малки РНК и в двете посоки. Това е отличителен белег на много локуси, генериращи малки РНК.

Открити бяха 56 РНК, които кодират зрели миРНК секвенции и са консервативни в различни растителни видове. Консервативните миРНК бяха класифицирани в 21 потвърдени миРНК семейства. От тях 12 са с повече от един член, а в 7 е открит повече от един вариант на секвенцията (Таблица 2).

Семейство	Координати на прекурсора	Посока	миРНК	Експресия (брой копия) на миРНК			
				GSM506620	GSM506621	GSM406302	GSM40630
156	Bd3:4336022..4336167	-	Bdi-miR156a	46	5276	444	114
	Bd3:39258110..39258314	-	Bdi-miR156b	46	5276	444	114
	Bd5:18202182..18202332	-	Bdi-miR156c	46	5276	444	114
159	Bd2:5602219..5602452	+	Bdi-miR159	342	2	11	7
160	Bd1:4550703..4550877	-	Bdi-miR160a	991	78	79	146
	Bd1:28020411..28020556	-	Bdi-miR160b	991	78	79	146
	Bd3:3414575..3414722	-	Bdi-miR160c	991	78	79	146
	Bd3:12734293..12734442	+	Bdi-miR160d	991	78	79	146
	Bd3:41554285..41554428	+	Bdi-miR160e	2259	33	197	97
164	Bd2:19949322..19949514	-	Bdi-miR164a	309	1972	20	90
	Bd1:14544140..14544348	+	Bdi-miR164b	309	1972	20	90
166	Bd1:6574363..6574508	+	Bdi-miR166a	2619	18956	159	247
	Bd1:30655617..30655860	-	Bdi-miR166b	2619	18956	159	247
	Bd3:33098767..33098938	+	Bdi-miR166c	2619	18956	159	247
	Bd1:71419377..71419541	-	Bdi-miR166d	2619	18956	159	247
	Bd3:51437868..51438018	+	Bdi-miR166e	1			
	Bd4:6090867..6091011	-	Bdi-miR166f	73	917	56	15
167	Bd1:6349025..6349198	+	Bdi-miR167a	6554	3142	182	173
	Bd1:3770002..3770191	+	Bdi-miR167b	6554	3142	182	173
	Bd1:54067075..54067233	+	bdi-miR167c	28334	4607	748	186
	Bd3:3632405..3632608	-	Bdi-miR167d	28334	4607	748	186
168	Bd3:1774700..1774835	-	Bdi-miR168a	89999	454059	3612	4233
169	Bd1:1175425..1175598	+	Bdi-miR169k	257	65	21	16
	Bd1:27159070..27159261	-	Bdi-miR169f	16	516	47	93
	Bd2:7704123..7704303	+	Bdi-miR169a	2143	260	820	2311
	Bd3:16738806..16738956	+	Bdi-miR169b	612	545	1184	669
	Bd4:26242409..26242595	+	Bdi-miR169d	4	24	61	116
	Bd3:43441526..43441689	+	Bdi-miR169e	8	59	106	88
	Bd3:43444486..43444666	+	Bdi-miR169g	8	59	106	88
	Bd4:44509037..44509211	+	Bdi-miR169i	120	15	54	76
	Bd4:44513754..44513936	+	Bdi-miR169j	4	3		
	Bd5:11563834..11563997	-	Bdi-miR169h	49	23	294	598
	Bd5:23763870..23764020	-	Bdi-miR169c	612	545	1184	669
171	Bd1:72765307..72765462	-	Bdi-miR171a	12182	429	382	481
	Bd1:6911133..6911296	+	Bdi-miR171b	12182	429	382	481
172	Bd2:58915755..58916008	-	Bdi-miR172b	68	322	2	159
	Bd3:55737301..55737453	-	Bdi-miR172a	1378	17153	167	437
390	Bd1:2722067..2722275	+	Bdi-miR390	62	68	3	1
393	Bd2:2001005..2001163	-	Bdi-miR393a	46	86	374	242
	Bd5:27613816..27613989	+	Bdi-miR393b	46	86	374	242
394	Bd3:52316372..52316571	-	Bdi-miR394	705	19	69	202
395	Bd1:55440558..55440795	-	Bdi-miR395a	355	14	62	152
	Bd4:16374432..16374612	+	Bdi-miR395b	355	14	62	152
	Bd5:25455997..25456122	+	Bdi-miR395c	355	14	62	152
396	Bd1:46677004..46677155	-	Bdi-miR396d	149	59	49	170
	Bd3:54962836..54963036	+	Bdi-miR396e	1147	20	300	364
	Bd3:54968138..54968290	-	Bdi-miR396c	149	59	49	170
	Bd3:59349783..59349991	-	Bdi-miR396a	27589	865	1908	2001
	Bd5:27112463..27112651	+	Bdi-miR396b	27589	865	1908	2001
397	Bd3:3149689..3149834	-	Bdi-miR397a	1379	66	136	3672
	Bd3:3149689..3149834	-	Bdi-miR397b	22599	3	3346	95556
398	Bd2:35924413..35924562	-	Bdi-miR398a	7		1	26
399	Bd3:7135266..7135425	+	Bdi-miR399a	1		2	
408	Bd2:10450343..10450535	+	Bdi-miR408	13	16	26	528
	Bd1:73059536..73059695	-	Bdi-miR528	211	222	124	2981
827	Bd5:18037481..18037644	-	Bdi-miR827	49	12	355	354

Таблица 2. Предвидени консервативни миРНК на *Brachypodium distachyon* и техните геномни координати.

Интересното е, че сравнителният анализ на рийдовете от четирите библиотеки GSM506620, GSM506621, GSM406303 и GSM406302 показва, че повечето консервативни миРНК имат много по-висока експресия в репродуктивни тъкани, в сравнение с вегетативните тъкани. Наред с откритите 56 консервативни миРНК, бяха намерени още 46 потенциални нови миРНК локуса, като търсенето по хомология чрез BLAST разкри, че никоя от секвенциите, образували правилна вторична структура, нямат прилики с други растителни миРНК. Те бяха маркирани като потенциално нови неконсервативни миРНК за генома на *Brachypodium* (Таблица 3).

За разлика от животните, където половината миРНК гени се намират в интрони, повечето от известните растителни миРНК гени се намират в интергенни области. Много малко са разположени в интроните на протеин-кодиращи гени. Съществуват няколко известни интронни миРНК в *Arabidopsis*, повечето от които не са консервативни в други растителни видове (Brown et al. 2008). Интересно е да се спомене, че от откритите в генома на *Brachipodium*, почти половината от неконсервативните миРНК гени (21) са разположени в интроните на протеин-кодиращи гени със същата посока.

4.2. Софтуерен пакет miRDEG за предвиждане на растителни миРНК, чрез интегриране на данни от малки РНК и деградомни секвении.

За настоящия дисертационен труд беше разработен пакет от софтуерни приложения и модули на име miRDEG, чиято обща цел е откриването на непознати миРНК чрез интегриране на данни от секвениране на малки РНК и деградомно

Координати на прекурсора	Посока	МиРНК	Експресия (брой копия) на миРНК							
			GSM506620		GSM506621		GSM40630		GSM406303	
			5′ сек.	3′ сек.	5′ сек.	3′ сек.	5′ сек.	3′ сек.	5′ сек.	3′ сек.
Bd1:15325058..15325194	+	Bdi-miR1000a	1	14		40		3		1
Bd1:22135899..22136079	–	Bdi-miR1001	47	6	22	4	2	1		4
Bd1:28306160..28306307	–	Bdi-miR1002	3	308	1	63		1		
Bd1:3006392..3006538	–	Bdi-miR1003	1	2		6				
Bd1:30313604..30313744	–	Bdi-miR1004		1				1	3	1
Bd1:44311568..44311956	+	Bdi-miR1005		20	4	9	1	7		3
Bd1:45787966..45788131	–	Bdi-miR1006	7	2		1		3	1	1
Bd1:46051254..46051407	–	Bdi-miR1007	1	1						
Bd1:56853936..56854310	–	Bdi-miR1008	11	17		2			1	1
Bd2:11317397..11317573	–	Bdi-miR1009	1	1						
	–	Bdi-miR1009-1	1	2				1		1
Bd2:14480262..14480480	–	Bdi-miR1010				4	1	1		1
Bd2:22520117..22520363	–	Bdi-miR1011				1			1	1
Bd2:41735710..41735862	+	Bdi-miR1012a	3	7		19	1	1		
Bd2:41890827..41891002	+	Bdi-miR1013	2	1						
Bd2:44925156..44925344	+	Bdi-miR1014	1	1		2				
Bd2:46665653..46665804	–	Bdi-miR1015	1	2		1				
Bd2:47213136..47213398	–	Bdi-miR1016	3	1				4	1	3
Bd2:48358975..48359115	+	Bdi-miR1017	9	9	1	5	5		33	
Bd2:51090546..51090760	–	Bdi-miR1018	5	128		1	1	1	1	1
Bd3:5200959..5201270	–	Bdi-miR1020	1	1	2					
Bd3:6356151..6356409	–	Bdi-miR1021	2	1				2		
Bd3:8463664..8463877	–	Bdi-miR1022	4	9		20	1	11		13
Bd3:9493263..9493513	–	Bdi-miR1023	177	10	355	6	214		45	
Bd3:9879796..9880206	+	Bdi-miR1024	2	1		3		1		
Bd3:14764296..14764443	–	Bdi-miR1025	23	66			2	2		1
Bd3:19465406..19465546	+	Bdi-miR1000b	2	2	22	2		1		
Bd3:21918225..21918387	+	Bdi-miR1041	1	0	0	0	1	1	557	21
Bd3:30460937..30461101	+	Bdi-miR1026	2	3			3	1		1
Bd3:56704432..56704571	+	Bdi-miR1027a							2	1
Bd3:59107074..59107211	–	Bdi-miR1028	2	1	87		19	1	6	
Bd4:2181951..2182125	+	Bdi-miR1029					1	1		1
Bd4:5647864..5647998	–	Bdi-miR1027b							2	1
Bd4:7045106..7045256	+	Bdi-miR1031	1	3		1			2	1
Bd4:10796980..10797184	+	Bdi-miR1032a					2	1	5	
Bd4:11947824..11948009	–	Bdi-miR1033		2265		68	2	25		11
Bd4:14440355..14440694	+	Bdi-miR1034	14	2		3	1			
Bd4:22543862..22544027	–	Bdi-miR1035	8	1	1		1		1	
Bd4:31253293..31253572	+	Bdi-miR1036				14			1	1
Bd4:33186148..33186396	–	Bdi-miR1037	2	2		1				
Bd4:37301939..37302074	+	Bdi-miR1038			2	1			1	
Bd4:37627843..37628050	+	Bdi-miR1039	3	216		494		9	1	5
Bd4:42126781..42127074	–	Bdi-miR1030	115	2	1		2	5	1	5
Bd4:42248170..42248379	–	Bdi-miR1032b	4	4	51		1		4	
Bd5:6094953..6095105	+	Bdi-miR1012b	3	7		19	1	1		
Bd5:24072517..24072691	–	Bdi-miR1040	1	1						

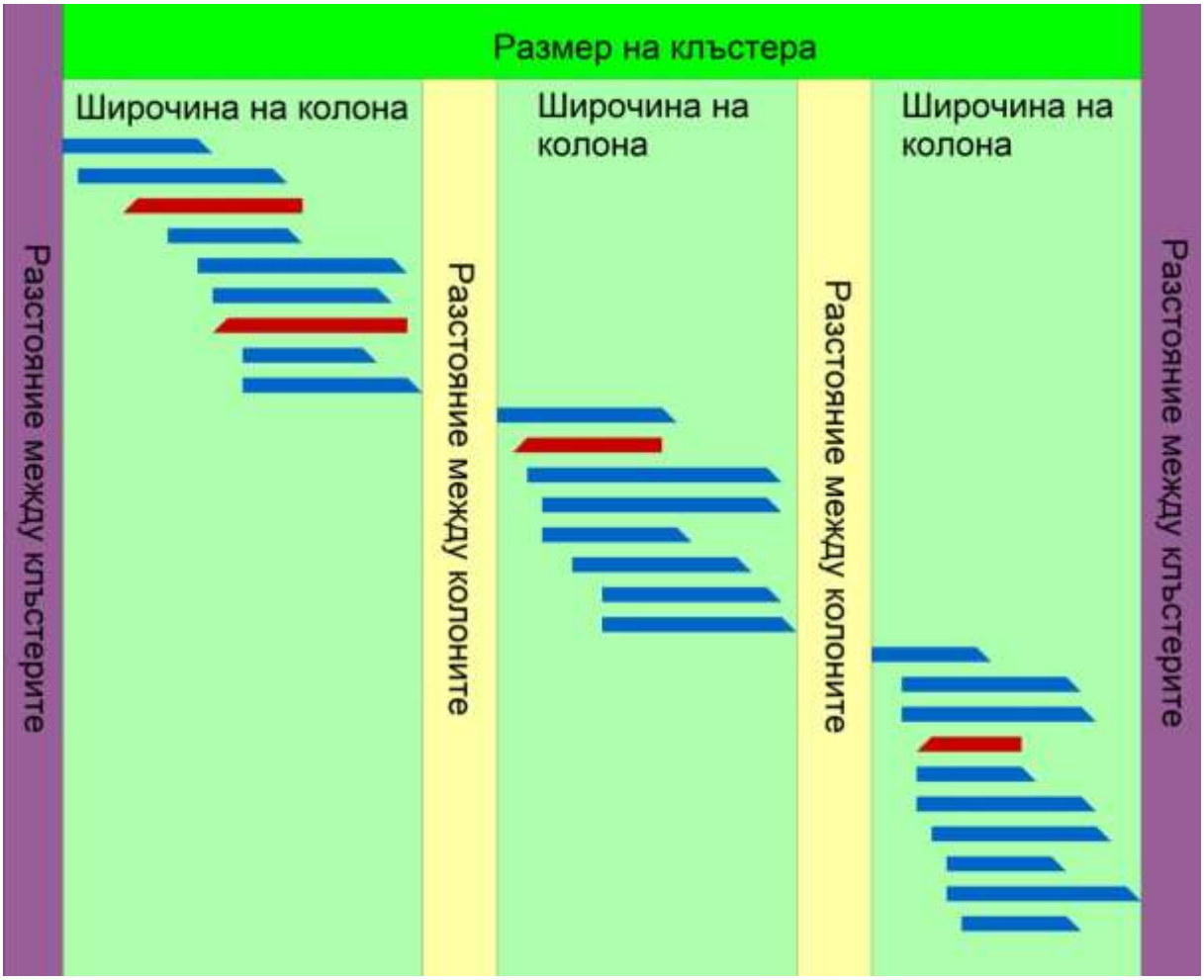
Таблица 3. Предвидени неконсервативни миРНК на *Brachypodium distachyon* и техните геномни координати.

секвениране. miRDEG се състои от програмите clusterSD, miRNA2D и miCompare, а модулът LoopRNA е използван като помощен инструмент. Всички гореспоменати софтуерни приложения са писани на език за програмиране Java. Системните спецификации, кода и подробните алгоритми на гореспоменатите софтуерни приложения ще бъдат разгледани в отделни глави. Тук ще дадем по-обща, теоретична представа за цялостния подход на miRDEG.

4.3. clusterSD

Клъстерирането на малки РНК в софтуерния пакет се извършва от разработената програма clusterSD. Нейната функция е да намира „клъстери“ от картирани малки РНК, които съвпадат с остатъците от процеса на миРНК биогенеза с конкретни стойности, зададени от потребителя. Най-общо, clusterSD цели да намери две групи от малки РНК рийдове с висока експресия, които потенциално биха съвпаднали с остатъците от дуплекса миРНК/миРНК*. За да се избегнат фалшиви позитивни резултати, потенциалните клъстери се подлагат на серия строги филтри.

Анализът се подсилва чрез интегрирането на данни от деградомно секвениране. По такъв начин за първи път могат да се диференцират локуси на дуплицирани миРНК и да се разграничи реално експресираният миРНК локус, тъй като, когато се използва само миРНК, данните се картират еднакво на всички дуплицирани региони.



Фигура 5. Примерно разпределение на клъстер чрез clusterSD

Като входни данни, clusterSD използва база данни от картирани малки РНК и деградомни рийдове (получен от програмата Bowtie, отново с опция за картиране не позволяваща несъвпадения в алайнмънта). В нея се търсят предварителни клъстери – прости последователности от малки РНК рийдове с по-малко от зададеното разстояние между тях. След като се намери така зададен клъстер, той се филтрира по доминантна посока на секвенциите, минимална експресия на рийдовете, максимален размер на клъстера и „очаквани колони“. Тук за „доминантна“ посока се взема посоката на даден процент от всички секвенции. За малките РНК рийдове, посока и експресия се търси само в рамките на клъстера. За деградомните рийдове, експресия и посока се търси в разширен регион около клъстера, с даден размер.

Филтърът за „очаквани колони“ се опитва да раздели малките РНК рийдове в клъстера на дискретни колони, на които се налагат допълнителни изисквания. Целта е да се открият колоните, които потенциално могат да съответстват на остатъците от миРНК и миРНК*. Колоните се определят като последователност от малки РНК рийдове със застъпващи се координати. От клъстера се изисква да има не повече от даден брой колони, както и двете най-големи колони да са еднопосочни и да са разделени на дадено минимално разстояние.

Клъстери, които не преминат дори през един от тези филтри се премахват. По избор на потребителя, останалите клъстери могат да се анотират спрямо зададен от потребителя GFF файл с геномна информация, за да се провери дали те съвпадат с познати геномни структури, като екзони, интрони, UTR и т.н. Крайните резултати се записват във файл с изходни данни.

На Фигура 3 е показано примерно разпределение на клъстер чрез clusterSD.

4.3.1. Техническо въведение

clusterSD е Java програма, част от по-големия софтуерен пакет miRDEG за идентификация на миРНК от резултати от масово-паралелно секвениране и представлява първата голяма стъпка от изследванията в него. Както показва името ѝ, clusterSD се фокусира върху намирането на клъстери от малки РНК рийдове и деградомни рийдове, които съвпадат с дистрибуцията от остатъци от миРНК биогенезата. Конкретните характеристики на търсените клъстери се задават от

Конзолна команда	Описание	Базова стойност
-help	Показва помощна документация	
-map	Името на файла с входни данни	
-annot	Името на файла за сравнение	
-gap	Максималното разстояние между клъстерите	30
-mins	Минималната експресия на малки РНК рийдове	10
-mind	Минималната експресия на деградомни рийдове	10
-srat	Минимален % еднопосочни малки РНК рийдове	95%
-drat	Минимален % еднопосочни деградомни рийдове	95%
-size	Максимален размер на клъстер	350
-dext	Фланкиращ регион за изчисление на деградомна експресия	100
-colwid	Максимална широчина на клъстерна колона	50
-colsep	Минимално разстояние между клъстерни колони	20
-colover	Минимален overlap между рийдовете в една колона	1
-colmax	Максимален брой клъстерни колони	4
-noover	Изключва филтриране по колони	false
-nosize	Изключва филтриране по размер на клъстера	false
-nogene	Изключва интергенната проверка	false
-multi	Указва, че входният файл съдържа множество хромозоми	false
-debug	Включва принтиране на междинни файлове	false
-nodel	Указва, да се не се трият временните директории преди работа	false

Таблица 4. Списък от конзолните параметри на програмата clusterSD с техните стойности по подразбиране и описания.

потребителя или се определят по подразбиране.

Програмата няма графичен интерфейс и се изпълнява директно от командния ред. clusterSD предлага голям брой настройки на потребителя, които контролират изпълнението на самата програма. Техните стойности се задават като допълнителни конзолни параметри при изпълнението на програмата. Кратко описание на тези конзолни параметри е показано на Таблица 4. Параметрите ще бъдат описани по-пълно при подробното описание на clusterSD. Параметрите, задаващи стойности на програмни променливи, имат стойности по подразбиране. Повечето от тях не са задължителни.

Работата на clusterSD се състои от следните стъпки:

4.3.2. Зареждане на входни данни

При изпълнението си, clusterSD изисква точно един файл с входни данни, зададен чрез команда –map. Изисква се този файл да е във формат, получен от Bowtie, като името на всяка една секвенция трябва да съдържа в себе си информация за нейната експресия от един или повече източника. Тези цифри се прибавят към името със знак тире (-) във вид <име>-<експресия>-<експресия>-… Имената на секвенции на малки РНК се очаква да започват с буква **S,** а имената на деградомни секвенции – с всякакъв друг символ.

Името на секвенцията се очаква в първата колона от файла. Във втората колона се очаква посока на секвенцията, отбелязана със знак плюс (+) или минус (-). В третата колона се очаква име на хромозомата, от която е дошла секвенцията. В четвъртата колона се очаква координат за начало на секвенцията. В петата колона се очаква самата секвенция. Останалите колони се пренебрегват. Файлът с входни данни трябва да е подреден първо по хромозома (ако съдържа данни за повече от една) и второ – по начални координати на секвенциите.

Ако входният файл съдържа повече от една хромозома, това задължително трябва да се отбележи като се добави конзолен параметър –multi. Това добавя допълнителна подготвителна стъпка преди да започне филтрирането. В тази стъпка, файлът с входните данни се разпада на серия по-малки файлове, всеки от които държи данните само за една хромозома. Всички следващи стъпки се извършват по веднъж за всеки така създаден временен файл.

При всяко стартиране на програмата, старите временни файлове се изтриват. Ако се работи върху непроменен файл с входни данни който вече веднъж е бил разпаднат при предишно пускане на програмата, то могат да се ползват временните файлове от това предишно пускане като се добави конзолната команда –nodel. Това спестява време.

След всичко това, файла с входните данни (или временна част от него, съдържаща само една хромозома) се зарежда в паметта като индексиран списък от текстови данни.

4.3.3. Създаване на първоначални клъстери

Преди да бъдат наложени каквито и да било филтри, първо трябва да се създаде списък от „клъстери“ който да бъде филтриран, за което е нужна дефиниция на клъстер. Тук програмата дефинира „клъстер“ като последователност от малки РНК рийдове, при които разстоянието между края на един рийд и началото на следващия е по-малко от разстоянието между клъстерите, зададено с конзолен параметър -gap.

Алгоритмът при тази стъпка започва от първият намерен рийд на малка РНК, чийто индекс в списъка с рийдовете се записва като начало на нов клъстер. След това алгоритмът преминава през следващите рийдове. Ако откритият рийд е малка РНК, неговият индекс в списъка се записва като временен край на клъстера. Това

продължава, докато не бъде намерен рийд, чието начало е на разстояние от края на предишния по-голямо от -gap. В тази ситуация, началото и края на клъстера се финализират и той се подава на следващата стъпка.

4.3.4. Филтриране на клъстери по посока

Ако един клъстер, намерен от предишната стъпка, представлява рийдове от истински миРНК/миРНК* дуплекс, то рийдовете в този клъстер трябва да са с една и съща посока на секвенцията. Поради неточност в методите за изследване и методологията за създаване на първоначални клъстери, е възможно в даден клъстер да попаднат и секвенции в „обратна“ посока. Такива клъстери се приемат, стига тяхната експресия да не надвишава дадени параметри.

За да се определи колко рийда има в „обратна“ посока, в клъстера първо трябва да се определи „доминантна“ посока. За тази цел се изчислява четири вида експресия – експресия на малки РНК в + и - посока, експресия на деградомни рийдове в + и - посока. На посоката на всеки отделен рийд се дава тежест, съответстваща на неговата експресия. Например, деградомен рийд с посока - и експресия 7 се брои за -7 деградом. Посоката на малките РНК се изчислява в рамките на намереният клъстер. Посоката на деградомните рийдове се изчислява в рамките на клъстер, разширен в двете посоки с размер зададен от -extend.

След това изчисление се намира доминантната посока на малките РНК и деградомните рийдове поотделно. За целта просто се сравнява експресията на + и - секвенциите, като се взема по-голямото число. Ако доминантната посока на малките РНК не съвпада с доминантната посока на деградомните рийдове, клъстерът се отхвърля.

След това се изчислява процентът на секвенциите в обратна посока спрямо всички секвенции. Това изчисление се извършва за миРНК и деградомните рийдове поотделно. Процентът се изчислява като *обратни_секвенции/всички_секвенции*. Ако обратните секвенции за миРНК или за деградомните рийдове представляват прекалено голям процент, клъстерът се отхвърля. Максималният процент обратни секвенции се задава с –srat за малките РНК и -drat за деградомните остатъци.

Във филтъра за посока, клъстерите също така се филтрират по експресия. Понеже експресията на клъстера трябва да се изчисли за намиране на посоката му така или иначе, по-бързо става тази проверка да се вмъкне там. Ако общата експресия на малките РНК или деградомните рийдове е прекалено малка, клъстерът се отхвърля. Минималната допустима експресия се задава с -mins за малките РНК и -mind за деградомните рийдове.

Клъстери, които преминат този филтър се приемат и записват в списъка с клъстерите. За клъстер се записва информация за неговото начало и край като индекси от списъка с входните данни, и неговата посока като 1 за + или 0 за -.

4.3.5. Филтриране на клъстери по размер

Поради избраната дефиниция за клъстер (последователност от малки РНК рийдове в близост един до друг), голяма част от намерените клъстери са огромни и очевидно не съответстват на биогенезата на миРНК. По тази причина се извършва елементарна и бърза проверка за размера на клъстерите. Всички прекалено големи клъстери се отхвърлят. Максималният размер на клъстер се задава с конзолен параметър -size.

Филтрирането се извършва с много прост алгоритъм. Всеки клъстер от списъка с намерените се изважда и за него се изчислява дължина от началото на първия рийд в

клъстера до края на последния. Ако тази дължина е по-голяма от дадената, клъстерът се премахва от списъка.

4.3.6. Филтриране на клъстери по колони

Ако един клъстер, преминал през предишния филтър, представлява рийдове от истински миРНК/миРНК* дуплекс, то рийдовете в този клъстер трябва да се подреждат поне в две колони, съответно отговарящи на миРНК и миРНК*. От рийдовете във всяка колона се очаква да се разминават с по няколко нуклеотида, а от самите колони да са на конкретно разстояние една от друга. Това изисква сложна проверка, която се прави на няколко подстъпки и като използва допълнителен модул, който моделира клъстерни колони.

Като начало се създава списък от колони от малки РНК в клъстера. Деградомните рийдове се игнорират напълно. Тук „колона“ е дефинирана като последователност от малки РНК рийдове, чиито координати се припокриват с поне даден брой нуклеотиди, дефиниран чрез конзолен параметър –colover. Със списъка от колоните в даден клъстер се правят няколко елементарни първоначални проверки.

Първо се проверява дали броят на колоните надвишава максималния допустим брой, зададен с конзолен параметър –colmax. Понеже е възможно непринадлежащи рийдове да попаднат в клъстера и да формират свои мини-колони, добре е да се позволят поне 4 колони. Ако клъстерът има повече от позволените колони, той се отхвърля. Ако клъстерът има само една колона, тя се проверява за максимален размер, зададен с конзолен параметър -colwid. Ако размерът на колоната е по-голям от това ограничение, клъстерът се премахва от списъка. Ако ли не, се приема. При някои програмни настройки е възможно миРНК и миРНК* да се открият като една колона, за това се позволяват такива клъстери.

След това, от списъка с колони в клъстера се избират двете най-големи – предполагаемите миРНК и миРНК*. Изчисляват се техните доминантни посоки както при филтриране по колони и резултатите се сравняват. Ако двете колони са с различна посока, клъстерът се премахва от списъка. В това правило, обаче, има изключение. Ако по-голямата съдържа в себе си достатъчно голям процент от експресията на целия клъстер, то по-малката колона се игнорира дори да е в обратна посока. Експресията на по-голямата колона се изчислява както при филтриране по колони, като процент от цялата експресия на клъстера. Ако този процент надвишава дадения с конзолен параметър -srat, то по-малката колона се игнорира.

От тук нататък се проверяват размерите на двете колони. Ако по-голямата колона е прекалено голяма, клъстерът се премахва от списъка. Ако по-малката колона е прекалено голяма и не е била игнорирана, то клъстерът се премахва от списъка. Размерът на колоните се изчислява като разстояние от началните координати на първия рийд до крайните координати на последния рийд в колоната. Максималният размер се задава с конзолен параметър -colwid.

Най-накрая се проверява разстоянието между двете избрани колони. Ако то е прекалено голямо, то клъстерът се премахва от списъка. Максималното допустимо разстояние се задава с конзолен параметър -colsep. Клъстерите, преминали всички тези филтри не се променят по никакъв начин.

4.3.7. Проверка за интергенност

Тази стъпка е анотираща, не филтрираща. Тя сравнява координатите на намерените клъстери и търси дали те се припокриват – частично или пълно – с вече известни гени. Като резултат, клъстерът се анотира в изходните данни с допълнителен

идентификатор за съвпадащия ген, ако има такъв. Клъстери, които не съвпаднат с познат ген се анотират като „???“ като знак за това.

За да се извърши тази анотация, в програмата трябва да се зареди файл с геномна анотация. Това се извършва с конзолен параметър -annot. Изисква се даденият файл да е в GFF формат. Съдържанието на файла с генетичната информация трябва да отговаря на съдържанието на файла с входните данни – те трябва да съдържат един и същ брой хромозоми, наименувани по един и същи начин и данните в двата файла трябва да са подредени първо по име на хромозома, после по начални и крайни координати.

Ако файлът с геномната анотация съдържа повече от една хромозома и отговаря на съответен файл с входни данни, то важат същите правила както при зареждане на входни данни. Достатъчно е конзолният параметър -multi да се изпише само веднъж, понеже той се отнася и за двата файла. С него, и двата ще бъдат разпаднати на временни файлове с по една хромозома. Същото се отнася и за конзолният параметър –nodel, който предотвратява изтриването на временните генетични файлове, прескача разделянето на новият файл и просто използва старите временни файлове. Клъстерите от всяка хромозома се анотират поотделно, като се сравняват с генния файл на тази хромозома.

В процеса на анотация, GFF файла се разпада на два списъка – списък от гени в + посока и списък от гени в - посока. Всеки клъстер се сравнява с гени само от списъка със съответната посока. Сравнението се извършва по координати като всеки клъстер се сравнява с гените по ред на техните координати. Ако бъде намерен такъв ген, чийто начални координати да са по-малки от крайните координати на конкретния клъстер и чиито крайни координати да по-големи от началните координати на клъстера, значи е намерен overlap. Тогава клъстерът се маркира с типа ген, с който е съвпаднал (CDS, exon, five_prime_UTR, gene, intro, mRNA, three_prime_UTR). Ако не бъде намерен ген, съвпадащ с клъстера, то той се маркира като „INTERGENIC“.

4.3.8. Изписване на резултатите

След създаване, филтриране и анотиране на клъстерите, последната стъпка от програмата clusterSD е изписването на данните в изходен файл. Името на този изходен файл не се задава от потребителя, а се генерира автоматично, като към името на файла с входни данни се прикача „output.gff3“. Форматът на файла не съвпада съвсем точно с GFF формата, защото при изписването се добавят две нови „контролни“ колони, където се записва общата експресия на малките РНК и на деградомните рийдове.

Ако файла с входните данни е съдържал повече от една хромозома, то този файл е бил разпаднат на по-малки временни файлове с по една хромозома и програмата е била пусната по веднъж за всеки от файловете. Като резултат в такива ситуации се получават множество файлове с изходни данни – по един за всяка хромозома. На края на програмата, всички тези файлове се обединяват в един общ изходен файл както е описано по-горе.

Форматът на изходният файл съвпада с GFF, но има две допълнителни колони накрая. В първата колона е записано името на хромозомата. В четвърта и пета са записани началните и крайните координати на намерените клъстери. В седма колона е записана посоката на клъстерите. В девета колона е записана генната информация на клъстерите, във формат „ID=<тип клъстер>“. Ако интергенната анотация е изключена от потребителя, то ще се появи “ID=???” на нейно място. В новите две колони са записани експресиите на малките РНК и деградомните рийдове, за бърза визуална проверка на резултатите.

С това завършва изпълнението на програмата clusterSD.

4.4. miRNA2D

Предсказването на вторичната РНК структура се поддържа от две външни програми – fastaFromBed и RNAfold, чието изпълнение е автоматизирано в програмата miRNA2D. Тъй като резултатите от клъстериране се записват само като геномни координати, fastaFromBed се използва за извличане на самите секвенции от геномен файл, предоставен от потребителя. След това се използва RNAfold за нагъване на всяка секвенция поотделно. Като резултат се записват получената вторична структура във Vienna формат, както и минималната свободна енергия на нагънатата секвенцията.

Преди да се пусне fastaFromBed, координатите на откритите клъстери могат да бъдат разширени. Ако размерът на клъстер е по-малък от даден минимален размер, то той се разширява с даден брой нуклеотиди и в двете посоки. В противен случай, той се разширява с 20 нуклеотида в двете посоки.

Филтрирането по вторична структура се извършва от програмата miRNA2D. Първоначално, чрез модула LoopRNA се създава дървовиден граф, който представя вторичната РНК структура в паметта. Стеблата, започващи от свободните краища на секвенцията, са корените на този граф, стеблата, които им принадлежат са клоните, а терминалните примки са листата. Това позволява по-бърз и интуитивен достъп до различните части от вторичната структура без да се налага повторен анализ на Vienna формата.

Първо, за всяка дадена секвенция се намира нейната най-голяма терминална примка, която след това се проверява за размер и сдвоеност. Ако примката е по-малка от даден минимален размер (изчислен от първия сдвоен елемент до последния) или нейният процент сдвоени елементи (изчислен като сдвоени елементи / всички елементи) е по-малък от дадена долна граница, целият клъстер се премахва. Целта тук е да се открие подходяща кандидат терминална примка, в която биха могли да комплементират миРНК и миРНК*.

След намирането на кандидат терминална примка, нейната секвенция се проверява за минимална свободна енергия чрез изчисление на Minimal Free Energy Index (MFEI) (Zhang et al. 2006). Този индекс се изчислява като модулът от минималната свободна енергия се раздели на броя гуанин и цитозин в секвенцията. Секвенции чийто MFEI е по-малък от дадена долна граница се премахват. Тази проверка цели да се подсигури условието, че предвидената терминална примка е достатъчно стабилна, за да се счита за истинска. Крайните резултати се записват в изходен файл.

4.4.1. Техническо въведените

miRNA2D е Java програма, част от по-големия софтуерен пакет miRDEG за идентификация на миРНК от резултати от масово-паралелно секвениране, и представлява втората стъпка от изследванията в него. Програмата се фокусира върху изследването на вторичната структура на региона с картираните малки РНК рийдове и деградомни рийдове. Търсят се секвенции, съдържащи тези рийдове, чиято вторична структура съдържа характерната за миРНК фуркетна примка и има достатъчно надеждни признаци на стабилност Конкретните характеристики на търсените вторични структури се задават от потребителя или се определят по подразбиране.

Програмата няма графичен интерфейс и се изпълнява директно от командния ред. miRNA2D предлага голям брой настройки на потребителя, които контролират изпълнението на самата програма. Техните стойности се задават като допълнителни конзолни параметри при изпълнението на програмата. Кратко описание на тези конзолни параметри е показано на Таблица 5. Параметрите ще бъдат описани по-пълно

Конзолна команда	Описание	Базова стойност
-mirna2d	Извиква тази програма	
-help	Показва помощна документация	
-gff	Име на GFF файла с входните данни	
-fasta	Име на FASTA файла със секвенираният геном	
-minlen	Минимален размер на клъстери за extend	100
-ext	Брой нуклеотиди за extend	150
-mfei	Минимална свободна енергия на вторичната структура	0.6
-minsize	Минимален размер на най-голямата фуркетна структура	100
-mincent	Минимален процент на сдвоеност в най-голямата фуркетна структура	0.65
-nodel	Указва да не се трият временните файлове	False
-skip	Указва да се пропуснат fastaFromBed и/или RNAfold	0
-gene	Включва интергенния филтър	false

Таблица 5. Списък от конзолните параметри на програмата miRNA2D с техните стойности по подразбиране и описания.

при подробното описание на miRNA2D. Параметрите, задаващи стойности на програмни променливи имат стойности по подразбиране. Повечето от тях не са задължителни.

Понеже програмата miRNA2D е част от miRDEG, то нейното изпълнение трябва да се повика специално. За целта, като първи конзолен параметър трябва да се постави -mirna2d. Всички останали параметри се поставят след него.

4.4.2. Зареждане на входните данни

При стартирането си, програмата miRNA2D изисква точно един файл с информация за клъстери от секвенции в GFF формат. Въпреки това, на входния файл се позволява да има допълнителни колони, стига всички колони изисквани от GFF формата да са на мястото си. Въпреки че miRNA2D може да работи с всякакви входни данни, тя е предназначена да работи с изходните данни от програмата clusterSD, като втора стъпка от този процес. За разлика от clusterSD, miRNA2D не се интересува дали файла с входните данни съдържа една или повече хромозоми. Файл с входни данни се задава с конзолен параметър -gff.

При стартиране, също е препоръчително на програмата да бъде подаден и файл със секвенциите, от които са произлезли изследваните клъстери. В GFF файла с входните данни са записани само геномните координати на намерените клъстери. За да се изследва тяхната вторична структура, по тези координати трябва да бъдат намерени съответните секвенции. За това е нужен файл с изследвания геном във FASTA формат, като името на всяка хромозома (като име на секвенцията) трябва да съвпада точно с името и във файла с входните данни. Файл със секвенции се задава с конзолен параметър -fasta.

При изпълнението си, miRNA2D използва две външни програми – fastaFromBed и RNAfold, като за тях създава серия временни файлове. Ако програмата се пуска серия пъти с едни и същи входни данни (един и същи –gff файл), то тези външни програми могат да бъдат прескочени за спестяване на време. В такъв случай, miRNA2D трябва да се инструктира да не трие временните файлове с конзолен параметър -nodel. Освен това, тя трябва да се инструктира да прескочи изпълнението на външни програми. Това се извършва с конзолен параметър -nodel, на който трябва да бъде зададено цяло число от следните: 0 не прескача нищо, 1 прескача само fastaFromBed, 2 прескача fastaFromBed и RNAfold, 3 прескача само RNAfold. Не е препоръчително да се използват числа, различни от тези.

Ако временните файлове не са изтрити и всички външни програми са прескочени, то не е задължително да се подава -fasta файл.

4.4.3. Обработка на входните данни

Поради използването на серия външни програми се налага входните данни да бъдат обработени преди да бъдат подадени нататък. При стартиране на програмата, файлът с входните данни се зарежда в паметта като списък, който в последствие се преподрежда по възходящ ред по име, начални координати и крайни координати на клъстера. Всеки ред от списъка се проверява за брой колони. Ако съдържа повече колони от позволените в GFF формата, съдържанието на допълнителните колони се прикача към последната. Всичко това се прави за да се подготви файла с входните данни за fastaFromBed.

Понеже файла с входните данни съдържа клъстери, чиито координати съвпадат точно с началните и крайните координати на потенциални миРНК/миРНК* дуплекси, вторичната структура, изчислена от тези секвенции, може да е неточна. По тази причина, клъстерът се удължава за да се създаде буфер около очакваната терминална примка. Ако клъстерът надвишава минималният допустим размер, той се разширява с 20 нуклеотида в двете посоки. Този минимален размер се задава от потребителя с конзолен параметър -minlen. Клъстери, които са по-малки от този минимален размер се разширяват с повече нуклеотиди в двете посоки. Конкретният размер на това разширение се задава от потребителя с конзолен параметър -ext.

4.4.4. Достъп до вторичната структура на секвенциите на клъстерите

След като входните данни са обработени в изричен GFF формат трябва да се достъпи до техните секвенции. За целта се използва модулът fastaFromBed от софтуерният пакет BedTools, като външна програма. fastaFromBed изисква файл с координати (-gff), файл с генна информация (-fasta) и даден нов файл, в който да се записват резултатите от търсенето. Процесът е напълно автоматизиран и не изисква ръчно извикване на fastaFromBed.

Този файл след това се подава към програмата RNAfold, която изчислява вторичната структура на всички така получени секвенции. RNAfold е конзолна програма, която може да нагъва само по една секвенция на пускане и връща резултата директно в конзолата без да го записва в отделен файл. По тази причина, за да се автоматизира нейното изпълнение, се използва Perl скрипт. Този скрипт пуска RNAfold по веднъж за всяка секвенция, улавя върнатия резултат и го записва в общ файл. Този файл се състои от 6 колони, съдържащи съответно хромозома, в която се намира секвенцията, начални и крайни координати на секвенцията, самата секвенция, вторичната структура на секвенцията във Vienna формат и минималната свободна енергия на секвенцията

Така полученият файл с вторична структура се зарежда в паметта. Филтрите се изпълняват върху него.

4.4.5. Филтър по вторична структура

Според миРНК биогенезата, дуплексът миРНК/миРНК* се намира в стеблото на дадена терминална примка. Всяка вторична структура се проверява за наличието на такава подходяща терминална примка. За улеснение и ускорение на това търсене, тук се използва Java модулът LoopRNA, който взима РНК вторична структура, записана във Vienna формат и я представя по много по интуитивен начин в паметта.

С помощта на LoopRNA се открива най-голямата терминална примка във вторичната структура, по която се филтрира цялата вторична структура. Първо, тя се проверява за размер. Ако примката е прекалено малка, вторичната структура се премахва от списъка. Минималният размер на терминалната примка се задава с

конзолен параметър -minsize. След това, чрез LoopRNA се проверява процентът на сдвоеност на примката. Ако този процент е прекалено малък, то вторичната структура се премахва от списъка.

Ако вторичната структура премине и през двете проверки, нейният запис в списъка се обновява, като в списъка остава само секвенцията и вторичната структура на терминалната примка. И ако процесът изглежда прост от това описание, то е защото по-голямата част от сложните изчисления се извършват от LoopRNA, като за miRNA2D остава само елементарната логика.

4.4.6. Филтър по минимална свободна енергия

Не всички програмно-предвидени вторични структури са реални. В зависимост от това как са извадени секвенциите от собствените си геноми, могат да се получат нестандартни нагъвания. По тази причина, от секвенциите на кандидат миРНК прекурсори се изисква те да са стабилни. В miRNA2D, проверка за стабилност се прави чрез измерване на Minimum Free Energy Index (MFEI). Понеже минималната свободна енергия зависи от размера на секвенцията, самата тя не ни дава достатъчно добра идея за стабилност. MFEI е нормализирана оценка на свободната енергия спрямо дължината на секвенцията. Ако GC content е броят на бази Гуанин и Урацил в дадената вторична структура, то тази оценка се изчислява като:

$$Minimal\ Free\ Energy\ Index = \frac{Minimum\ Free\ Energy}{GC\ content}$$

Програмно, това е доста просто. Първо се изброяват G и C нуклеотидите в секвенцията на най-голямата терминална примка, след което минималната свободна енергия се разделя на резултата. Ако този резултат е прекалено малък, то секвенцията се премахва от списъка.

4.4.7. Изписване на резултатите

Преди да се изпишат резултатите остава една последна стъпка. По време на обработката на файла с входните данни чрез fastaFromBed и RNAfold_се губят някои важни данни. Изчезва записът за посока на секвенцията, както и колоната с допълнителна генетична информация. За да може резултатът от програмата да е възможно най-пълен, данните след филтрирането се обединяват с обработените данни от преди филтрирането в един общ списък.

Последната стъпка в програмата miRNA2D е изписване на резултатите в изходен файл. Името на този файл е фиксирано – *FinalGFF.gff* – а съдържанието му е записано във формат, сходен с GFF. Резултантният файл съдържа всички изисквани колони за GFF формата, като добавя и серия допълнителни такива, конкретно съдържащи секвенцията и вторичната структура на откритата терминална примка, както и минималната и свободна енергия.

4.5. Java модул LoopRNA

LoopRNA е Java модул, предназначен да опрости автоматичния анализ на РНК вторична структура. Чрез методите на обектно-ориентирано програмиране и теория на графите, модулът LoopRNA представя вторичната структура на РНК като нелинейна дървовидна структура от самостоятелни самоопределящи се обекти. По този начин, данните се представят във формат който е удобен за ръчна обработка както и лесен за машинна обработка, която иначе би била доста сложна. Модулът LoopRNA не представлява самостоятелна програма, но интегрирането му във вече съществуващи програми е лесно, благодарение на неговия прост програмен интерфейс.

Главната цел на модула LoopRNA е да улесни писането на програми за обработка и анализ на РНК вторични структури. Въпреки че съществуват редица формати за представянето на такива вторични структури – от по-прости като Vienna до много сложни като RNAML – работата с тях изисква данните да се преобработват много пъти за сравнително елементарни операции. Това губи време и усложнява кода на програмата. LoopRNA обработва данните само веднъж и създава серия обекти, съответстващи на структурните елементи на дадено РНК нагъване. Тези обекти в последствие се подреждат в дървовиден граф с изрична йерархия записана в самите тях. Така няма нужда да се обработва цялата вторична структура само за да се изследва даден неин елемент.

Горепосоченият процес не е особено сложен от програмна гледна точка, но той е построен върху няколко основни идели от обектно-ориентирано програмиране и теория на графите. Като резултат се създава структура от данни, която „знае“ много за себе, което ѝ позволява да извършва сложни действия през много прост програмен интерфейс. В обобщение, целта на LoopRNA е да създаде „умна“ структура от данни, която да улесни работата на програмиста като автоматизира много от повтарящите се изчисления.

LoopRNA е самостоятелен модул, предназначен за вграждане в съществуващи Java програми. Той няма специални системни изисквания и предоставя серия публични методи, достатъчни за работа с него. Единственото изискване на модула е да му бъде предоставена РНК вторична структура във Vienna формат, от която модулът да създаде собствената си вътрешна структура от обекти.

Модулът LoopRNA се състои от три основни компонента: Loop, Structure и пакет с помощни методи. Пакетът с помощни методи се състои от няколко много елементарни класа и не заслужава по-подробно описание.

4.5.1. Loop

Loop е самостоятелен клас, който моделира един елемент от вторичната структура на РНК – стебло или терминална примка. Полупримки и вътрешни примки не са моделирани. Ако съществуват, то те се считат за част от едно стебло. Всеки Loop знае за себе си следната основна информация: собствено уникално име, вторичната структура от която е част, зададена във Vienna формат, начален нуклеотид и краен нуклеотид.

Тук за начален и краен нуклеотид се приемат първият и последният сдвоен нуклеотид. Освен това, Loop знае за себе си към кой друг Loop „принадлежи“ и кои други Loop-ове „принадлежат“ на него.

4.5.2. Structure

Structure е самостоятелен клас, който моделира цялостната вторична структура на дадена РНК, и представлява същността на модула LoopRNA. Structure знае за себе си следната основна информация: вторичната структура на РНК във формат Vienna, анотирано копие на вторичната структура с координати на принадлежащите си сдвоени елементи и списък от всички Loop-ове, подреден по начални координати.

Конструирането на един обект от клас Structure – на една РНК вторична структура – е доста сложен процес, съставен от 4 стъпки. Първо дадената РНК вторична структура във Vienna формат се „анотира“ чрез метод enumerator(). Той създава нов, по-сложен запис, в който всеки символ от структурата носи със себе си и информация с кой друг символ е сдвоен.

След като всички сдвоени елементи са анотирани правилно се преминава към създаване на самите Loop елементи. Това се извършва от метод listBuilder(). Този метод

търси само непрекъснати стебла и в 3‘ посоката и в 5‘ посоката. Въпреки че това дублира почти всички стебла, някои от тях се намират само в една от двете посоки.

На този етап, Loop-овете не са свързани по никакъв начин. За да се свържат се използва метода listOwnership(). Той има две функции – да премахне дублираните елементи и да свържи останалите в посока „предшественик,“ като се подсигури че всеки Loop има само по един предшественик. РНК вторичната структура има дървовиден вид, което налага тази проверка. За да се създаде връзка в посока наследници се използва метода bulgeRemover().

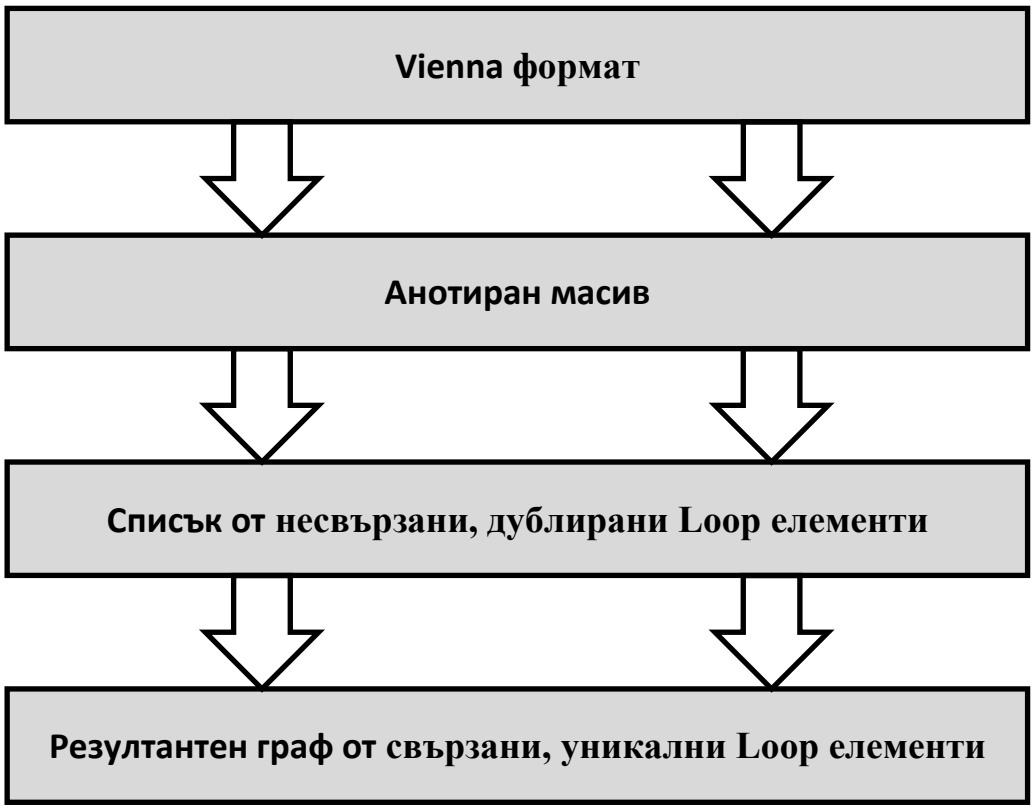
Методът bulgeRemover() се използва не само за свързване на елементите в посока наследници, но и за отстраняване на вътрешни примки и полупримки с цел опростяване на крайният резултат. Информация не се губи, но се намалява броят на Loop елементите, които трябва да се изследват.

С това завършва конструкцията на Structure, което ни предоставя опростена структура от стебла и примки, свързани в дървовиден граф и готови за директно ползване, без да се губи важна информация. Понеже всеки Loop носи в себе си копие на цялата вторична структура, то той има достъп до пълната информация.

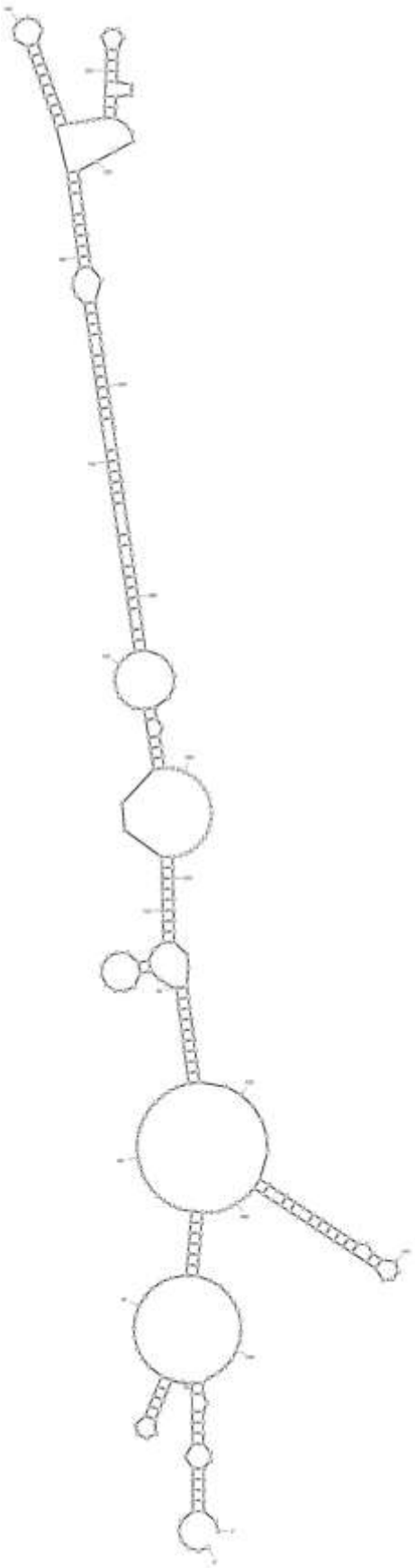
На Фигура 4 е показана опростена визуализация на целия процес по създаване на Structure.

4.5.3. Примерна употреба на LoopRNA

Нека вземем конкретен пример, с който да илюстрираме както техническата част на LoopRNA, така и някои негови потенциални приложения. Тук ще използваме секвенцията на миРНК *mir166a* от *Arabidopsis Thaliana* заедно с фланкиращ регион около нея. Цялата секвенция е с дължина 400 нт, което ни дава достатъчно сложна вторична структура за демонстрация.



Фигура 7. Опростена визуализация на Structure



Фигура 9. Вторична структура на региона на mir166a

Модулът LoopRNA изисква предварително-изчислена вторична структура, за което ще ползваме UNAFold Web Server. На Фигура 5 е показано графично представяне на най-стабилната вторична структура от UNAFold Web Server. Тази картинка е ясна за човешкото око, но не е подходяща за машинна обработка. За тази цел ни трябва Vienna формата на същата структура. В Таблица 6 са показани секвенцията на *mir166a* и Vienna формата на нейната вторична структура.

След като вторичната структура се прекара през модула LoopRNA, като резултат получаваме 9 Loop структури, които са показани в Таблица 7. Номерата не са последователни заради премахването на Loop елементи в процеса на създаване на Structure.

Тук виждаме един Loop, който не принадлежи на нищо (is owned by nothing) – Loop 0. Това е стебло, което започва от свободните краища на РНК секвенцията. Loop 0 притежава (owns) Loop 2 и 3. Това е първото разделение на стеблата във вторичната структура и се вижда като голяма примка на картинката. Loop 2 е малка терминална примка, но Loop 3 съдържа цялата останала секвенция и притежава всички останали Loop-ове (не всички директно). Loop 3 се разделя на малката терминална примка Loop 17 и голямото стебло Loop 4. Loop 4 се разделя на малката терминална примка Loop 5 и Loop 6. Самият Loop 6 се разделя на Loop 42 и 16 – две малки терминални примки.

Този формат улеснява машинното изследване на региона на *mir166a*, а също така улеснява и ръчното изследване до известна степен. Например, ако трябва да намерим всички терминални примки, то е достатъчно да потърсим Loop-ове, които не притежават нищо. В този случай, това са Loop 2, 5, 42, 16, 17. Някои от тези структури съдържат вътрешни примки, но тук вече може да се разгледа самата Vienna структура директно. Всеки Loop има копие от вторичната структура и информация за собствените си координати в нея.

Mir166a има координати 128-148 в региона от примера. От биогенезата на миРНК знаем, че дуплексът миРНК/миРНК* трябва да се намира в дълго стебло или завършващо с терминална примка, или разклоняващо се на две къси терминални примки. Loop 6 е дълго стебло (96-305), което се разклонява на две къси терминални примки (Loop 42 и 16). Дори само това

0	CUCCCUUAAUCUUAGCCGGGUCUCGUGGGGGACGAACAUAGAAAGAGAGAGAUAUAAAGAUAUAUAUUCAGAAACCCUAG
80	AUUCUAUAAUUUCGACUGAAAAGAAAAAGGGGCUUUCUCUUUUGAGGGGACUGUUGUCUGGCUCGAGGACUCUGGCUCGC
160	UCUAUUCAUGUUGGAUCUCUUUCGAUCUAACAAUCGAAUUGAACCUUCAGAUUUCAGAUUUGAUUAGGGUUUUAGCGUCU
240	UCGGACCAGGCUUCAUUCCCCCCAAUUGUUGCUCCCUGUUUACUCCAUAUUUCUUCCUUCUUUUCAAAUUAGGGUUUCAG
320	AUCCAGUGAAUGAACCCUUGUUAAAGGUUUGAUCUCUUACCUUACUUUCUCAGAUUCAUUUACAUGCGUUAUUGAUUGUU

0	......(((((.((((((.(((((....)))))...........(((((((...................((((((((((
80	..((.........)).(((((((((..((((((........(((.(((((.((..((((((..(((((((.((((...((
160	((((.((((((((((((......)))))))))...(((((((....))))..))).....))).)))))).)))).))))
240	)))..))))))..)).))))).))).....)).))))....................)))))))))..))))))))))..
320	....((.((.((((((.((....)))))))).)).)).......)))))))..............)).))))..)))))..

Таблица 6. Секвенция и вторична структура на mir166a региона

елементарно наблюдение на потенциален миРНК сайт, който в този случай е верен – mir166a действително попада в стеблото на Loop 6.

4.5.4. Заключение за LoopRNA

Идеята на модула LoopRNA е да опише дълга, сложна РНК вторична структура по прост начин чрез малък брой основни елементи без да загуби никаква информация или точност. Дори когато се наложи по-сложна обработка на самата секвенция, структурите на LoopRNA все пак опростяват процеса като смаляват региона, който трябва да се изследва. Това прави програмите по-бързи, по-лесни за писане и по-лесни за обновление.

Важно е да се подчертае, че LoopRNA не е формат за съхранение на данни. По-добри такива вече съществуват, като например Vienna и RNAML. LoopRNA също така не е инструмент за нагъване на РНК. LoopRNA е начин на представяне на данни за РНК вторична структура в паметта чрез използване на теория на графите и обектно-ориентирано програмиране.

Целта на LoopRNA е да помогне на дадена програма да „разбере“ идеите на РНК вторична структура и да ги разпознае по подобен начин както човек би могъл, гледайки картинката от Фигура 5. Вместо да разглеждат РНК вторична структура като серия от незначещи, самостоятелни символи, програми използващи LoopRNA я разглеждат като йерархична съвкупност от значещи логични елементи. Така сложни изчислителни проблеми се свеждат до прости логически въпроси.

4.5.5. Уникални аспекти на LoopRNA

LoopRNA се различава от вече съществуващи биологични приложения по обектно-ориентирания си дизайн и основната си идея.

В биоинформатиката, обектно-ориентираното програмиране до момента се използва основно за писане на интерфейси – прозорци, бутони, графики, текстови полета и т.н. – където няма много други удобни варианти. Изчислителната част на повечето биоинформатични програми използва традиционно линейно програмиране - алгоритми, съставени от последователни команди. LoopRNA има по-различен подход.

```
Loop 0 start 6 end 397, owns 2 3, is owned by nothing
Loop 2 start 19 end 32, owns nothing, is owned by 0
Loop 3 start 44 end 370, owns 4 17, is owned by 0
Loop 4 start 70 end 317, owns 5 6, is owned by 3
Loop 5 start 82 end 94, owns nothing, is owned by 4
Loop 6 start 96 end 305, owns 42 16, is owned by 4
Loop 42 start 168 end 191, owns nothing, is owned by 6
Loop 16 start 195 end 214, owns nothing, is owned by 6
Loop 17 start 324 end 356, owns nothing, is owned by 3
```

Таблица 7. Изходни данни от LoopRNA за региона на mir166a

Модулът създава по един отделен независим логически обект за всяка отделна РНК структура – стебло или примка. Това прави програмата "по-умна" и позволява да се извършват много сложни изчисления с много прост програмен код.

Основната идея на LoopRNA не е да нагъва РНК (тя изисква вече нагъната РНК във Vienna формат) и не е предназначена да представя данни на потребителя. Идеята на модула е да представя дадените ѝ данни на програма, в която е вградена. LoopRNA формулира данните по начин подобен на този, по който хората я възприемат при преглед на графично представяне на нагъната РНК. Vienna просто записва нагънатата секвенция като комбинация от скоби и точки, докато RNAML добавя допълнителна химична информация към секвенцията. И в двата случая, обаче, не е лесно да се установи кое стебло къде започва, до къде свършва и как се разклонява. Ако човек погледне картинка на нагъната РНК, то вижда стебла и примки. Компютърът вижда само съвкупност от нуклеотиди.

LoopRNA позволява на една програма да "мисли" като човек, поне до известна степен. Вместо да обработва цялата секвенция, програмата може да запита LoopRNA за всички фуркетни структури, или всички стебла, или колко пъти се "разклонява" една секвенция и всичкото това без да трябва да обработва секвенцията отново и отново. Дори да трябва да се обработва секвенцията директно, LoopRNA намалява обработвания регион. Вместо да се търси във всичките 400 нуклеотида, например, може да се търси в два региона от по 15 нулкеотида. Така „компютърът“ може да „погледне“ една РНК вторична структура и да отговори на „разговорни“ въпроси като например „Коя е най-голямата фуркетна структура?“ или „Колко фуркетни структури има?" и дори „Има ли стебло по-голямо от 100 нуклеотида, което се разклонява не повече от два пъти, като всяко разклонение е фуркетна структура, не по-голяма от 20 нулкеотида?“

Накратко, идеята на LoopRNA е да позволи на биологичен софтуер лесно да намира отговори на неясни разговорни въпроси без да изисква дълго, сложно и многократно обработване на самата секвенция.

4.6. miCompare

Анотацията на резултатите от miRDEG се извършва от програмата miCompare и не е задължителна. Програмата сравнява резултатите от предишната стъпка с даден файл, съдържащ известни миРНК и анотира съвпадащите резултати по координати или по хомология с едно несъответствие на секвенцията с други зрели растителни миРНК. Целта на тази стъпка е да се осъществи качествен контрол над резултатите като провери колко и кои вече-известни миРНК (от дадена база данни, като например miRBase) са открити чрез miRDEG.

4.6.1. Техническо въведение

miCompare е Java програма, част от по-големия софтуерен пакет miRDEG за идентификация на миРНК от резултати от масово-паралелно секвениране и представлява третата (незадължителна) стъпка от изследванията в него. Програмата се фокусира върху качествената проверка на резултатите от miRNA2D, като сравнява идентифицираните кандидат миРНК с познати, доказани миРНК от съществуващи бази данни като mirBase.

Понеже до момента не е открита една точна, обща дефиниция на миРНК, то стойностите на филтрите в програмите clusterSD и miRNA2D са компромисни. Те се избират така, че да се намерят възможно най-много от вече познатите миРНК, като общият брой на всички кандидат ми-РНК е достатъчно малък, че те да бъдат проверени експериментално. miCompare ни дава представа дали избраните от потребителя стойности за филтрите са прекалено стриктни или прекалено отпуснати.

miCompare няма графичен интерфейс и се изпълнява директно от командния ред. Тя предлага няколко настройки на потребителя, които контролират изпълнението на самата програма. Техните стойности се задават като допълнителни конзолни параметри при изпълнението на програмата. Кратко описание на тези конзолни параметри е показано на Таблица 8. Параметрите, задаващи стойности на програмни променливи имат стойности по подразбиране. Някои от тях не са задължителни.

Понеже програмата miCompare е част от miRDEG, то нейното изпълнение трябва да се повика специално. За целта, като първи конзолен параметър трябва да се постави –compare. Всички останали параметри се поставят след него.

4.6.2. Зареждане на входните данни

При стартиране, програмата miCompare изисква поне два файла. Първият файл е този с изходните данни от miRNA2D в GFF формат, които представляват входни данни към miCompare. Това е файлът с кандидат миРНК, които ще се сверяват с вече известните такива. Той се задава с конзолен параметър -result и се зарежда в списък с входни данни.

Вторият задължителен файл е извадка от база данни, съдържаща координатите на вече известни миРНК. От този файл също се изисква да е в GFF формат, като имената на хромозомите на миРНК в него трябва да съвпадат точно с тези от файла с изходните данни. Този файл се задава с конзолен параметър -coord и се записва в списък с координати.

Освен задължителните файлове, miCompare може да приеме и един незадължителен файл, съдържащ секвенциите на вече познати миРНК. Форматът на този файл е нестандартен. От него се очаква да се състои от две колони, като първата колона съдържа името на дадена миРНК, а втората колона съдържа нейната секвенция.

Файлът със секвенции за сравнение се задава с конзолен параметър -seq и се записва в списък със секвенции.

4.6.3. Сравнение по координати

След като данните са заредени в паметта, кандидат миРНК от списъка с входните данни се сравнява със списъка с координатите. Всяка кандидат миРНК се сравнява с всички познати миРНК, докато не се намери съвпадение или не се изчерпа списъка с координатите. Първо се правят две бързи сравнения, за да се отхвърлят очевидните несъответствия. Ако хромозомите на кандидат миРНК и на познатата миРНК не съвпадат се преминава към следващата. Ако посоките на секвенциите им не съвпадат, се преминава към следващата.

Конзолна команда	Описание	Базова стойност
-result	Име на файла с кандидат миРНК	
-coord	Име на файла с координати на известни миРНК	
-seq	Име на файла със секвенции на известни миРНК	
-overlap	Минимален overlap между резултатите и познатите миРНК координати	1
-mis	Максимален брой несъответствия с познатите миРНК секвенции	1

Таблица 8. Списък от конзолните параметри на програмата miCompare с техните стойности по подразбиране и описания.

Когато бъде намерена вече позната миРНК със същата посока и от същата хромозома, както кандидат миРНК, тогава се прави сравнение по координати. Търси се дали тези секвенции се засичат с поне даден брой нклеотиди, определен с конзолен параметър –overlap. Конкретно, проверява се дали началото на кандидат миРНК се намира преди края на познатата на разстояние поне -overlap и дали началото на познатата се намира преди края на кандидата на разстояние поне -overlap.

Ако бъде намерено такова съвпадение, то се записва и в двата списъка, накрая на съвпадащите редове. На реда в списъка с входните данни се записва името на познатата миРНК, съвпаднала с него. На реда в списъка с координатите се записва думата „*FOUND*“. Ако не бъде намерено съвпадение, този факт се записва само в списъка с входните данни, като накрая на несъвпадащият ред се поставя буквата „*N*“.

4.6.4. Сравнение по секвенция

Ако е бил предоставен списък със секвенции на вече -познати миРНК при стартиране на програмата, то кандидат миРНК от списъка с входните данни се сравняват с познатите миРНК от списъка със секвенциите. Всяка кандидат миРНК се сравнява с всички познати миРНК, докато не се намери съвпадение или не се изчерпа списъка със секвенциите. Първо, кандидат миРНК се сравнява със секвенцията на познатата миРНК както е записана във файла. Ако това сравнение е неуспешно, то познатата миРНК се сравнява с обърнатата секвенция на познатата. И в двата случая, на алайнмента се позволяват най-много дадено количество грешки, зададено с конзолен параметър –mis.

Ако бъде намерено такова съвпадение, то се записва и в двата списъка, накрая на съвпадащите редове. На реда в списъка с входните данни се записва името на познатата миРНК, съвпаднала с него, последвана от буквите „*AN*“. На реда в списъка със секвенциите се записва думата „*FOUND*“. Ако не бъде намерено съвпадение, този факт се записва само в списъка с входните данни, като накрая на несъвпадащият ред се поставя буквата „*N*“.

4.6.5. Изписване на резултатите

След като са извършени всички стъпки за сравнение, списъците от данни се изписват в изходни файлове. Първо се изписва списъкът с входни данни. Името на неговия изходен файл съответства на името на входния му файл. То се образува като от входния файл се премахне удължението и към края му се прикрепи „*found.gff*“. Този файл се създава в директория *./output*.

След това се изписва списъкът с координатите. Името на неговия изходен файл се определя по същия начин и той също се изписва в директория *./output*. Първоначалното копие на файла не се променя – създава се ново. Ако при стартиране на програмата е бил подаден файл със секвенции, то след това се изписва и списъкът със секвенциите. Името на неговия изходен файл се определя по същия начин и той също се изписва в директория *./output*. Първоначалното копие на файла не се променя – създава се ново.

И при файла с координатите и при файла със секвенциите е възможно да се появят по няколко колони с думата „*FOUND*“ ако няколко кандидат миРНК съвпаднат с една и съща вече-позната миРНК.

С това завършва изпълнението на програмата miCompare.

4.7. Предвиждане на миРНК в генома на *Solanum lycopersicum*

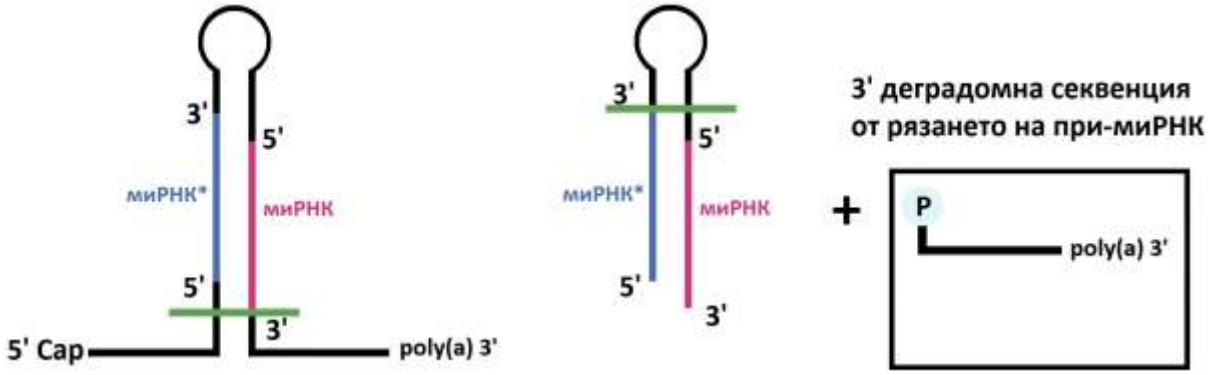
В много растителни видове, зрели миРНК с една и съща секвенция могат да произлизат от различни локуси, групирани или разпръснати върху целия геном. С напредването на технологиите за секвениране на цели геноми, секвенциите на предполагаеми миРНК гени могат да се предсказват *in silico*, по присъствието на зряла миРНК-кодираща геномна секвенция. Търси се потенциална фуркетна структура (пре-миРНК), където секвенцията на предполагаемата миРНК преобладава и където са известни голямо количество малки РНК (миРНК*), комплементиращи със секвенцията пре-миРНК фуркетата (Meyers et al. 2008).

За съжаление, идентификацията на ниско експресирани миРНК е трудна поради ниското количество на съответните и миРНК* секвенции. Понеже дадена зряла миРНК може да произлезе от много различни геномни локуси *in silico*, трудно се определя кои миРНК гени наистина се експресорат в дадена тъкан.

При растенията, разрязването на иРНК е главният механизъм за пост-транскрипционна регулация чрез миРНК. миРНК таргети могат да се идентифицират чрез секвениране на 20 нт секвенциите на 5' краищата на кДНК фрагменти, получени от разрязването на таргетната иРНК чрез миРНК. Голямо количество уникални 20 нт тагове в 3' посока от предполагаемото миРНК срязване се счита за доказателство за значителната активност на предполагаемата миРНК, в конкретната тъкан и растение. Този метод (Parallel Analysis of RNA ends (PARE) или „деградомно секвениране“) вече е приложен в много растителни видове за предвиждане на миРНК таргетните молекули (Addo-Quaye et al. 2008; German et al. 2008).

Секвенциите в деградомните библиотеки не са само продукти от миРНК-насочено срязване. Те са продукти на други процеси на деградация на иРНК със свободни 5' краища. при-миРНК транскриптите се синтезират от РНК полимераза II, и са поли-аденилирани (Xie 2005). Техните гени могат да бъдат значително по-дълги от прекурсорните пре-миРНК и могат да съдържат интрони (Szarzynska et al. 2009). Така биогенезата на пре-миРНК също произвежда деградомни секвенции, които могат да се използват за проследяване на експресираният миРНК локус (Addo-Quaye et al. 2009; Meng et al. 2010), (Фигура 6).

В настоящата дисертация беше предложена следната хипотеза: Експресираните зрели миРНК секвенции, (особено без намерени съответни миРНК* секвенции) може и да не са достатъчни, за да се идентифицира недвусмислено кодиращата пре-миРНК. Ако обаче те се комбинират с изомиР секвенции и с уникални картирани деградомни секвенции, тази идентификация става възможна. Освен това, този подход може да



Фигура 11. Схема на миРНК биогенеза. DCL1-медииранoто рязане на пре-миРНК и при-миРНК е показано в зелено.

идентифицира и нови непознати при-миРНК гени, както и да разграничи реално експресираните миРНК локуси. За тази цел беше разработен софтуерният пакет miRDEG за идентификация на миРНК, който да използва на мапнати малки РНК заедно с деградомни данни, за да идентифицира кандидат миРНК (Фигура 7).

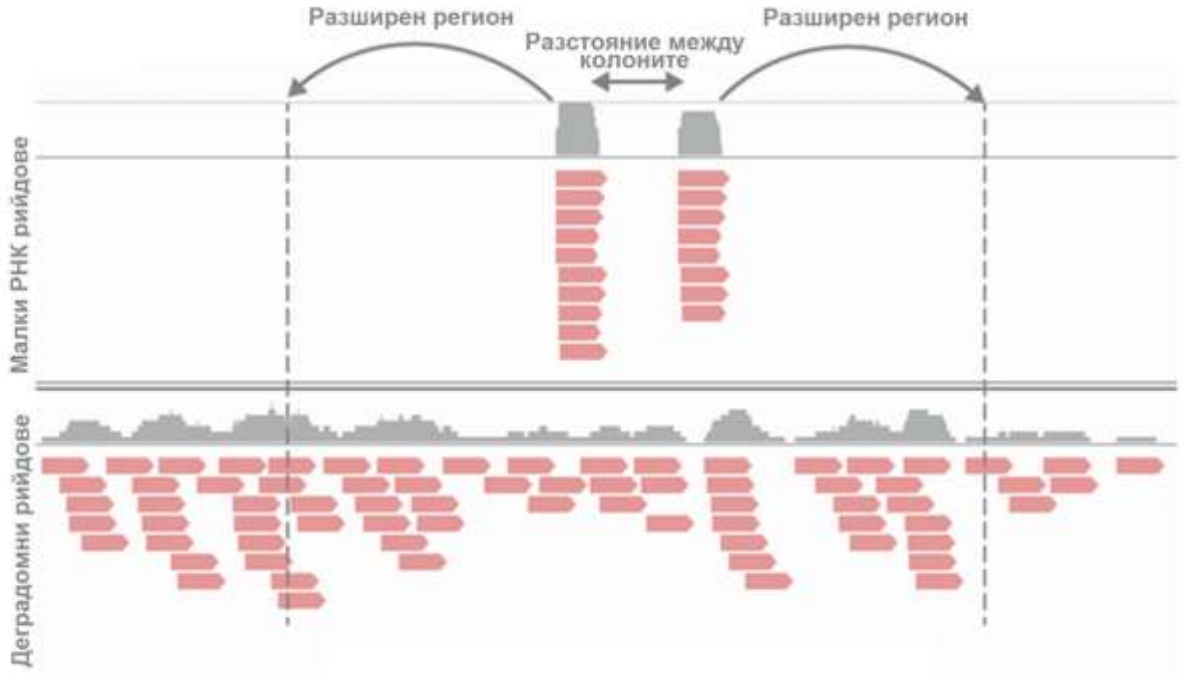
В miRDEG, clusterSD първо сканира мапнатите рийдове и открива клъстери от малки РНК с припокриващи се рийдове. Идентифицират се координати на клъстери от малки РНК или двойки клъстери (в случай че миРНК и миРНК* присъстват в клъстера). Малките РНК рийдове в клъстерите трябва да имат една и съща посока на секвенцията и минимална експресия, които се задават от потребителя. За клъстерите от малки РНК, преминали тези филтри, clusterSD търси деградомни рийдове в разширен геномен регион около дадения клъстер. Намерените деградомни рийдове трябва да имат същата посока като малките РНК и минимална експресия, които се задават от потребителя. Клъстерите, преминали през clusterSD се изписват в изходен файл и се предават на miRNA2D. От своя страна miRNA2D извлича региони на клъстерите (с разширение, зададено от потребителя) от генома (Фиг. 7). Така получените секвенции се тестват за образуване на стабилна фуркетна структура и за достатъчна MFEI оценка. Локусите, преминали през тези филтри, се изписват в изходен файл и се предават на miCompare.

miCompare сравнява клъстерите с геномни структури, като например вече известни пре-миРНК. Използвайки търсене по хомоложност, miCompare може да анотира нови локуси, кодиращи вече известни зрели миРНК в растителният геном.

Чрез софтуерния пакет miRDEG, включващ модулите ClusterSD и LoopRNA бяха анализирани публично наличните библиотеки за малки РНК (GSM452714, GSM452715, GSM452716, GSM452717, GSM452718, GSM452719, GSM452720, GSM452721) и от деградомни данни (GSM1047560, to GSM1047562, GSM1047561, GSM10475630) при Solanum lycopersicum със следните параметри:

ClusterSD:

- Максималното разстояние между клъстерите: 60 нт



Фигура 13. Схематична фигура на алгоритъма на софтуерният пакет miRDEG базиран на използването на данни от малки РНК и деградомни данни от МПС

Геномни координати	Посока	Малки РНК	Дег РНК	миРНК	Номер в mirBase
SL2.40ch01:88965114-88965497	+	25	19	sly-MIR5304	MI0018481
SL2.40ch01:70879614-70879989	+	537	12	sly-MIR403	MI0029100
SL2.40ch01:74999558-74999982	-	131	147	sly-MIR6024	MI0020242
SL2.40ch03:58819214-58819537	+	50	729	sly-MIR482a	MI0018482
SL2.40ch03:49898337-49898791	+	397	80	sly-MIR319c	MI0029099
SL2.40ch04:31078638-31079048	-	57	10	sly-MIR5303	MI0018480
SL2.40ch04:63484351-63484753	+	158	11	sly-MIR394	MI0029102
SL2.40ch04:55142028-55142488	+	1178	611	sly-MIR482e	MI0018478
SL2.40ch05:64582538-64582923	-	38	50	sly-MIR160a	MI0008357
SL2.40ch06:39241528-39241928	-	35431	14	sly-MIR156a	MI0009971
SL2.40ch06:42918586-42918991	+	131	55	sly-MIR172a	MI0009976
SL2.40ch06:33877843-33878215	-	56	31	sly-MIR482c	MI0020251
SL2.40ch06:33869995-33870539	-	461	2656	sly-MIR482d	MI0029113
SL2.40ch07:58089013-58089404	+	72	2513	sly-MIR171e	MI0029125
SL2.40ch07:2628613-2629038	-	1632	483	sly-MIR396b	MI0029124
SL2.40ch08:49154054-49154637	-	225	10	sly-MIR156e	MI0029110
SL2.40ch08:62860590-62860916	+	61354	1022	sly-MIR166b	MI0008359
SL2.40ch08:53776187-53776648	+	1117	14	sly-MIR1919c	MI0008356
SL2.40ch12:63568987-63569310	+	125	140	sly-MIR168b	MI0024353
SL2.40ch12:1977598-1978019	-	17026	368	sly-MIR9471a	MI0029107

Таблица 9. Предвидени миРНК в генома на *Solanum lycopersicum* присъстващи в miRBase

- Минимален процент еднопосочни малки РНК рийдове: 90%
- Минимален процент еднопосочни деградомни рийдове: 90%
- Минималната експресия (брой копия) на малки РНК рийдове: 30
- Минималната експресия(брой копия) на деградомни рийдове: 10
- Максимален размер на клъстер: 300 нт
- Фланкиращ регион за изчисление на деградомна експресия: 100 нт
- Максимална широчина на клъстерна колона: 60 нт
- Минимално разстояние между клъстерни колони: 15 нт
- Минимално препокриване между рийдовете в една колона: 1 нт
- Максимален брой клъстерни колони: 4

miRNA2D:

- Минимален размер на разширения регион: 500 нт
- Минимален MFEI: 0.6
- Минимален размер на най-голямата фуркетна структура: 70 нт
- Минимален процент сдвоеност в най-голямата фуркетна структура: 50%

Чрез посочените параметри бяха потвърдени нови 25 миРНК (Таблица 9). Открити бяха 20 нови локуса на известни миРНК (Таблица 10) и 7 напълно нови миРНК гена (Таблица 11).

Трябва да се подчертае, че посоченият нов метод не се стреми да предвиди или открие всички миРНК в даден геном. До момента, много софтуерни продукти са се опитвали да постигнат точно това, поради което голям процент от подадените и аноритани миРНК в базите данни се оказват фалшиви позитиви и съответно не реални миРНК гени. miRDEG се стреми да използва не само данни от малки РНК (на които разчитат останалите програми), а допълнително интегрира и данни от деградомно секвениране. По този начин, софтуерният пакет увеличава “доказателствата” за експресия на даден миРНК локус.

Геномни координати	Посока	Малки РНК	Деградомни РНК	миРНК
SL2.40ch00:12537365-12537776	+	4684	2818	sly-miR396a-5p
SL2.40ch01:79166892-79167311	+	63803	179	sly-miR166c-3p
SL2.40ch02:16516453-16516835	-	26	516	sly-miR160a
SL2.40ch03:46023406-46023734	-	62651	11	sly-miR166c-3p
SL2.40ch03:58491392-58491775	+	1763	78	sly-miR162
SL2.40ch03:61719880-61720209	+	35967	18	sly-miR156
SL2.40ch03:61785942-61786392	+	22591	48066	sly-miR159
SL2.40ch04:4093633-4093963	-	48	76	sly-miR172b
SL2.40ch05:64339521-64339926	+	130	571	sly-miR172b
SL2.40ch06:33130203-33130543	+	62821	85	sly-miR166c-3p
SL2.40ch06:39462967-39463365	+	1763	211	sly-miR162
SL2.40ch07:323840-324411	+	9733	36	sly-miR156d-5p
SL2.40ch07:58136935-58137318	+	63	682	sly-miR171c
SL2.40ch08:49143133-49143565	-	349	19	sly-miR156d-5p
SL2.40ch08:49143382-49143780	-	9662	10	sly-miR156d-5p
SL2.40ch08:49143624-49144048	-	9687	96	sly-miR156d-5p
SL2.40ch08:61949380-61949931	-	574	23	sly-miR319b
SL2.40ch09:63580236-63580567	-	33	107	sly-miR5303
SL2.40ch09:63883259-63883813	+	1043	224	sly-miR167a
SL2.40ch10:3321737-3322079	+	61394	136	sly-miR166
SL2.40ch11:2634467-2634813	+	121	68	sly-miR172b
SL2.40ch12:2098644-2099114	-	88	44	sly-miR9471b
SL2.40ch12:2564779-2565108	-	21	35	sly-miR9471b
SL2.40ch12:8971285-8971747	-	78	11	sly-miR9471b
SL2.40ch12:9245971-9246297	-	374	10	sly-miR9471b

Таблица 10. Предвидени нови локуси за познати миРНК в генома на *Solanum lycopersicum*

До момента, за предвиждането на миРНК се е разчитало единствено на малки РНК секвенции, но не трябва да забравяме, че зрелите миРНК са идентични, когато става въпрос за експресия от различни дуплицирани локуси. По тази причина не може да се идентифицира реалният миРНК локус от дадената проба, ако в дадени условия се експресира само един от дубликатите чрез картиране на зрелите миРНК. Още повече, могат да възникнат и грешно картирани зрели миРНК. Тези проблеми могат да се решат чрез интегрирането и на втори тип данни – деградомни, които допълнително ще служат като доказателство за експресията на даденият локус.

Тъй като само зрялата миРНК и понякога миРНК* са идентични в прекурсора на дуплицираните миРНК, то деградомните секвенции могат да служат като допълнителен белег за индивидуалността на идентифицираният миРНК локус. Освен това деградомните данните допълнително ще обогатят картината на миРНК биогенезата и т.нар „шаблон“ при картиране на малки РНК към генома. Това ще направи откриването на миРНК гени по-лесно, като същевременно ще се намалят фалшивите позитиви в резултатите на разработеният софтуер.

Геномни координати	Посока	Малки РНК	Деградомни РНК	Зрели миРНК
SL2.40ch02:40705715-40706139	-	122	24	
SL2.40ch01:82801169-82801545	-	135	24	
SL2.40ch06:32251666-32252036	+	46	719	
SL2.40ch09:4663828-4664243	-	631	12	
SL2.40ch04:2407356-2407791	-	63	104	
SL2.40ch06:34713514-34713983	+	341	2727	
SL2.40ch12:39442704-39443131	+	184	1025	

Таблица 11. Намерени нови миРНК за генома на *Solanum lycopersicum*

5. **Изводи**

1. Разработеният Clustering Module, базиран на *de novo* методи за клъстеризация на малки РНК от МПС данни и на идентифициране на профила, характерен за миРНК локусите, позволява успешно да се предвиждат миРНК гени в растителни геноми.
2. Прилагането на Clustering Module за анализ на малки РНК от масово–паралелно секвениране данни от *Brachypodium distachyon* за първи път бяха открити 102 миРНК гена – 56 консервативни миРНК, принадлежащи към 21 семейства и 46 неконсервативни миРНК.
3. Чрез интегриране на два типа данни от масово-паралелно секвениране - малки РНК секвенции и деградомни секвенции, беше разработен модул "*ClusterSD*" за клъстеризация и предвиждане на миРНК гени и за идентифициране на реално експресирани миРНК локуси.
4. Чрез използването на методите на обектно ориентирано програмиране беше създаден модулът LoopRNA, който анотира РНК вторичната структура и я представя като дървовиден граф. Работата с този граф е значително по-лесна и интуитивна в сравнение със съществуващите формати.
5. Чрез прилагане на софтуерния пакет miRDEG, включващ модулите *ClusterSD* и *LoopRNA* в генома на *Solanum lycopersicum,* бяха потвърдени 25 миРНК, открити 22 нови локуса на известни миРНК и 7 локуса на нови миРНК гена.

6. **Литература**

Addo-Quaye, C. et al., 2008. Endogenous siRNA and miRNA Targets Identified by Sequencing of the Arabidopsis Degradome. *Current Biology*, 18(10), pp.758–762.

Addo-Quaye, C. et al., 2009. Sliced microRNA targets and precise loop-first processing of MIR319 hairpins revealed by analysis of the Physcomitrella patens degradome. *RNA*, 15(12), pp.2112–2121. Available at: http://rnajournal.cshlp.org/cgi/doi/10.1261/rna.1774909.

Brown, J.W.S., Marshall, D.F. & Echeverria, M., 2008. Intronic noncoding RNAs and splicing. *Trends in Plant Science*, 13(7), pp.335–342.

German, M.A. et al., 2008. Global identification of microRNA–target RNA pairs by parallel analysis of RNA ends. *Nature Biotechnology*, 26(8), pp.941–946. Available at: http://www.nature.com/doifinder/10.1038/nbt1417.

Meng, Y. et al., 2010. High-throughput degradome sequencing can be used to gain insights into microRNA precursor metabolism. *Journal of Experimental Botany*, 61(14), pp.3833–3837.

Meyers, B.C. et al., 2008. Criteria for Annotation of Plant MicroRNAs. *THE PLANT CELL ONLINE*, 20(12), pp.3186–3190. Available at: http://www.plantcell.org/cgi/doi/10.1105/tpc.108.064311.

Reinhart, B.J. et al., 2002. MicroRNAs in plants. *Genes and Development*, 16(13), pp.1616–1626.

Szarzynska, B. et al., 2009. Gene structures and processing of Arabidopsis thaliana HYL1-dependent pri-miRNAs. *Nucleic Acids Research*, 37(9), pp.3083–3093.

Unver, T. & Budak, H., 2009. Conserved micrornas and their targets in model grass species brachypodium distachyon. *Planta*, 230(4), pp.659–669.

Wei, B. et al., 2009. Novel microRNAs uncovered by deep sequencing of small RNA transcriptomes in bread wheat (Triticum aestivum L.) and Brachypodium distachyon (L.) Beauv. *Functional and Integrative Genomics*, 9(4), pp.499–511.

Xie, Z., 2005. Expression of Arabidopsis MIRNA Genes. *PLANT PHYSIOLOGY*, 138(4), pp.2145–2154. Available at: http://www.plantphysiol.org/cgi/doi/10.1104/pp.105.062943.

Zhang, B.H. et al., 2006. Evidence that miRNAs are different from other RNAs. *Cellular and Molecular Life Sciences*, 63(2), pp.246–254.

Zhang, J. et al., 2009. Deep sequencing of Brachypodium small RNAs at the global genome level identifies microRNAs involved in cold stress response. *BMC Genomics*, 10(1), p.449. Available at: http://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-10-449.

7. **Съкращения**

AFL	Academic Free License, академичен свободен лиценз
AGO	Аргонавт (Argonaute) протеини
ASCII	American Standard Code for Information Interchange, Американски стандартен код за обмен на информация
DCL	Dcer-Like ген
EPL	Eclipse Public License, Eclipse публичен лиценз
GPL	GNU General Purpose License, Общ публичен лиценз на ГНУ
IDE	Integrated Development Environment, среда за софтуерна разработка
JVM	Java virtual machine, виртуална машина на Java
MFE	Minimum Free Energy, минимална свободна енергия
MFEI	Minimal Free Energy Index, индекс на минималната свободна енергия
MITE	Miniature Inverted-repeat Transposable Element, малки обратни повтори
NGS/МПС	Next Generation Sequencing, масово паралелно секвениране
qRT-PCR	quantitative Real-Rime Polymerase Chain Reaction, количествена Полимеразна Верижна Реакция (кПВР)
RACE	Rapid Amplification of cDNA Ends, бърза амплификация на кДНК краища
RISC	RNA-Induced Silencing Complex, РНК-индуциран сайлънсинг комплекс
sRNA-seq	small RNA sequencing, технология за секвениране на малки РНК
TE	Transposable Elements, транспозонни елементи
TE-MIR	растителни миРНК, произхождащи от Transposable Element
UTR	Untranslated Region, нетранслируем регион
иРНК	информационна РНК
миРНК	микро РНК
мтДНК	митохондриална РНК
ООП	Обектно-Ориентирано Програмиране
при-миРНК	първична микро РНК
рРНК	рибозомна РНК
тРНК	транспортна РНК

8. **Научни приноси, свързани с дисертационния труд**

- Разработен е за първи път многопрофилен биоинформатичен софтуер, интегриращ методите на обектно-ориентираното програмиране и теорията на математичните графи за анотиране на нови потенциални миРНА гени при разтения.
- За пръв път е използвана комбинация от данни на малки РНК и деградомни секвенции от масово–паралелно секвениране за идентификацияна миРНК гени.
- Чрез разработените софтуерни продукти за първи път бяха идентифицирани 102 нови миРНК гена при *Brachypodium distachyon* и 29 нови миРНК при *Solanum lycopersicum*.

9. **Публикации, свързани с дисертационния труд**

LoopRNA – Java module for RNA structure annotation. Minkov G., Toneva V., Ivanova Z., Baev V. 2017. Comptes rendus de l'Acade'mie bulgare des Sciences (in press).
(IF 0,25)

Implementation of a de novo genome-wide computational approach for updating Brachypodium miRNAs. Baev, V., Milev, I., Naydenov, M., Apostolova, M., Minkov, G., Minkov, I., Yahubyan G. Genomics. 97, 282–293, 2011.
(IF 2,8)