ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αναζήτηση σε επιστημονικές βάσεις δεδομένων με βάση την ιστορική εξέλιξη των δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΗΛΙΑ ΚΑΝΕΛΛΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2012

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αναζήτηση σε επιστημονικές βάσεις δεδομένων με βάση την ιστορική εξέλιξη των δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΗΛΙΑ ΚΑΝΕΛΛΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9η Ιουλίου 2012.

.....	.....	.....
Τιμολέων Σελλής, Καθηγητής Ε.Μ.Π.	Ιωάννης Βασιλείου, Καθηγητής Ε.Μ.Π.	Θοδωρής Δαλαμάγκας, Ερευνητής Β΄ ΙΠΣΥ/Ε.Κ. “Αθηνά”

Αθήνα, Ιούλιος 2012

.....

ΗΛΙΑΣ ΚΑΝΕΛΛΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Σκοπός της εργασίας ήταν η ανάπτυξη μιας εφαρμογής προβολής των αλλαγών που αφορούν τα δεδομένα που καταγράφονται για βιομόρια micro RNA. Η εφαρμογή ενσωματώνεται ως επέκταση σε λογισμικό που διατίθεται από την ιστοσελίδα DIANA, η οποία αναπτύχθηκε σε συνεργασία με το ερευνητικό κέντρο «Αλέξανδρος Φλέμινγκ». Τα μόρια micro RNA αποτελούν ένα σημαντικό τομέα έρευνας της βιολογίας, καθώς σχετίζονται με την έκφραση γονιδίων και συνεπώς ευθύνονται για την εκδήλωση ασθενειών. Καθώς εμπλουτίζονται τα δεδομένα που προκύπτουν από την έρευνα γύρω από τα μόρια αυτά, εκδίδονται νέες εκδόσεις των βάσεων δεδομένων που τα καταγράφουν. Από έκδοση σε έκδοση των βάσεων αυτών, ενδέχεται να παρουσιάζονται αλλαγές στα δεδομένα αυτά. Το αποτέλεσμα είναι η παρουσίαση δυσκολιών στην έρευνα, καθώς ο ερευνητής δεν μπορεί να έχει τη συνολική εικόνα για τη γνώση που υπάρχει γύρω από ένα συγκεκριμένο μόριο micro RNA. Στα πλαίσια της διπλωματικής (α) κατασκευάστηκε μια βάση δεδομένων που κωδικοποιεί όλη την εξέλιξη των δεδομένων για τα micro RNAs που καταγράφονται από την βάση mirbase (www.mirbase.org) και (β) υλοποιήθηκε μια διαδικτυακή εφαρμογή που προβάλλει τους γράφους εξέλιξης των δεδομένων για micro RNAs από έκδοση σε έκδοση, κάνοντας χρήση της παραπάνω βάσης.

Λέξεις κλειδιά: βιολογία, micro RNA, mirbase, διαδικτυακή εφαρμογή, βάσεις δεδομένων, εξέλιξη δεδομένων, Yii widget.

Abstract

The aim of this thesis is to implement a web application for presenting changes recorded in multiple versions of microRNA databanks. The application has been integrated into the DIANA web application toolkit, which is a set of tools for analyzing microRNA data, and has been developed in cooperation with the research center "Alexander Fleming". MicroRNA molecules constitute an important branch of biological research, since their function is correlated with gene expression, and thus, microRNAs are related to diseases and treatments. As new data emerges from research concerning these biological molecules, new versions of databases recording them are issued. It is possible that recorded data differs between various versions of these databases. As a result, there can be difficulties in research, since a researcher cannot have a complete overview of all recorded knowledge concerning a specific microRNA. The aim of this thesis is (i) to implement a database containing information about the evolution of microRNA data recorded in the mirbase archives (www.mirbase.org) and (ii) to implement a web application that presents evolution graphs for recorded microRNA data.

Keywords: biology, micro RNA, mirbase, web application, databases, data evolution, Yii widget.

Περιεχόμενα

1 Εισαγωγή 11

1.1 Το πρόβλημα της αναζήτησης πληροφοριών για micro RNAs 11

1.1.1 Το κεντρικό δόγμα της μοριακής βιολογίας 11

1.1.2 Τι είναι τα micro RNAs; 13

1.1.3 Το πρόβλημα της εξέλιξης δεδομένων στην έρευνα βιομορίων 15

1.2 Αντικείμενο της διπλωματικής 16

1.2.1 Συνεισφορά 16

1.3 Οργάνωση κειμένου 17

2 Θεωρητικό υπόβαθρο και σχετικές εργασίες 19

2.1 Η βάση βιολογικών δεδομένων mirbase 19

2.2 Αρχεία δεδομένων της βάσης mirbase 20

2.2.1 Αρχεία .dat 22

2.2.2 Αρχεία .fa 33

2.2.3 Αρχεία .diff 35

2.2.4 Αρχεία .dead 35

2.3 Διαφοροποιήσεις παρεχόμενων αρχείων της mirbase βάσει της έκδοσης. 36

2.4 Μοντελοποίηση αλλαγών στην καταγραφή δεδομένων των μορίων micro RNA .. 38

2.5 Το σύστημα DIANA microT v4.0 41

2.6 Σύντομη παρουσίαση της βάσης ensEMBL 42

3 Σχεδίαση συστήματος I: Χτίζοντας το backbone της εφαρμογής 45

3.1 Μοντελοποίηση της εξέλιξης δεδομένων με σχεσιακή βάση mySQL 45

3.1.1 Ο πίνακας hairpinhistory 46

3.1.2 Ο πίνακας maturehistory 54

3.1.3 Υλοποίηση script για το χτίσιμο της βάσης 59

3.1.4 Σύγκριση αποτελεσμάτων με τα αρχεία της mirbase 62

3.1.5 Υλοποίηση script για την αναβάθμιση της βάσης σε νέες εκδόσεις 63

3.1.6 Σενάρια χρήσης διαχειριστή της βάσης δεδομένων 65

3.1.7 Περιπτώσεις εσφαλμένων αποτελεσμάτων λόγω ασυνέπειας των καταγεγραμμένων δεδομένων της mirBase 75
4 Σχεδίαση (II) και ανάπτυξη εφαρμογής παρουσίασης εξελικτικών γράφων 79
4.1 Ανάλυση απαιτήσεων συστήματος 79
4.2 Τεχνολογίες, πλατφόρμες και προγραμματιστικά εργαλεία 82
4.2.1. Apache http server 83
4.2.2. mySQL 83
4.2.3. PHP 84
4.2.4. jQuery και τεχνολογία AJAX 88
4.3 Μοντελοποίηση και υλοποίηση εφαρμογής εξελικτικών γράφων 90
4.3.1 Περιγραφή κλάσεων 90
4.3.2 Λεπτομέρειες υλοποίησης 94
4.4 Παρουσίαση εφαρμογής 103
5 Επίλογος 111
5.1 Σύνοψη 111
5.2 Μελλοντικές εργασίες 112
Επέκταση εφαρμογής γράφων εξέλιξης σε δεδομένα γονιδίων της βάσης ensEMBL .. 112
Ενσωμάτωση νεότερων εκδόσεων της mirbase στις εφαρμογές DIANA 113
Πρόσθετες λειτουργίες και χρήση σε άλλες εφαρμογές DIANA 113
6 Βιβλιογραφία 115

1

Εισαγωγή

1.1 Το πρόβλημα της αναζήτησης πληροφοριών για micro RNAs

1.1.1 Το κεντρικό δόγμα της μοριακής βιολογίας

Μακρομόρια στο εσωτερικό των κυττάρων είναι υπεύθυνα για τις διάφορες λειτουργίες του: αποτελούν στοιχεία της δομής του, λειτουργούν ως καταλύτες για χημικές αντιδράσεις (ένζυμα), ανιχνεύουν αλλαγές που συμβαίνουν στο περιβάλλον. Τα μόρια αυτά συνήθως είναι πρωτεΐνες, ή άλλα είδη βιοπολυμερών στο κύτταρο.

Το γενετικό υλικό (DNA - δεοξυριβονουκλεϊκό οξύ) δεν είναι υπεύθυνο για τις παραπάνω λειτουργίες. Αποτελεί όμως ένα είδος αποθήκης δεδομένων, καθώς περιέχει τη γενετική πληροφορία των οργανισμών. Η πληροφορία αυτή πρέπει να αποκωδικοποιηθεί για να παραχθούν άλλα μόρια, όπως είναι οι πρωτεΐνες.

Το DNA βρίσκεται κυρίως στον πυρήνα των κυττάρων και περιέχει τέσσερις χημικές βάσεις: την αδενίνη (A), τη γουανίνη (G), τη θυμίνη (T) και την κυτοσίνη (C). Οι βάσεις αυτές σχηματίζουν ζευγάρια. Η αδενίνη ταιριάζει με τη θυμίνη και η γουανίνη

με τη κυτοσίνη. Τα ζεύγη αυτά ονομάζονται ζεύγη βάσεων. Κάθε βάση ενώνεται με μια ομάδα φωσφορικού οξέως και με ένα σάκχαρο. Ο σχηματισμός αυτός ονομάζεται νουκλεοτίδιο. Τα νουκλεοτίδια οργανώνονται σε δύο κλώνους, οι οποίοι σχηματίζουν μια έλικα. Η έλικα αυτή ονομάζεται διπλή έλικα.

Για να περιέχει το κάθε κύτταρο ενός οργανισμού την ίδια πληροφορία, η οποία βρίσκεται στο DNA, απαιτείται η δημιουργία των αντίγραφών του. Αυτό πραγματοποιείται σε μια διαδικασία που ονομάζεται αντιγραφή. Κάθε κλώνος της διπλής έλικας του DNA μπορεί να λειτουργήσει ως πρότυπο για την αντιγραφή της ακολουθίας βάσεων.

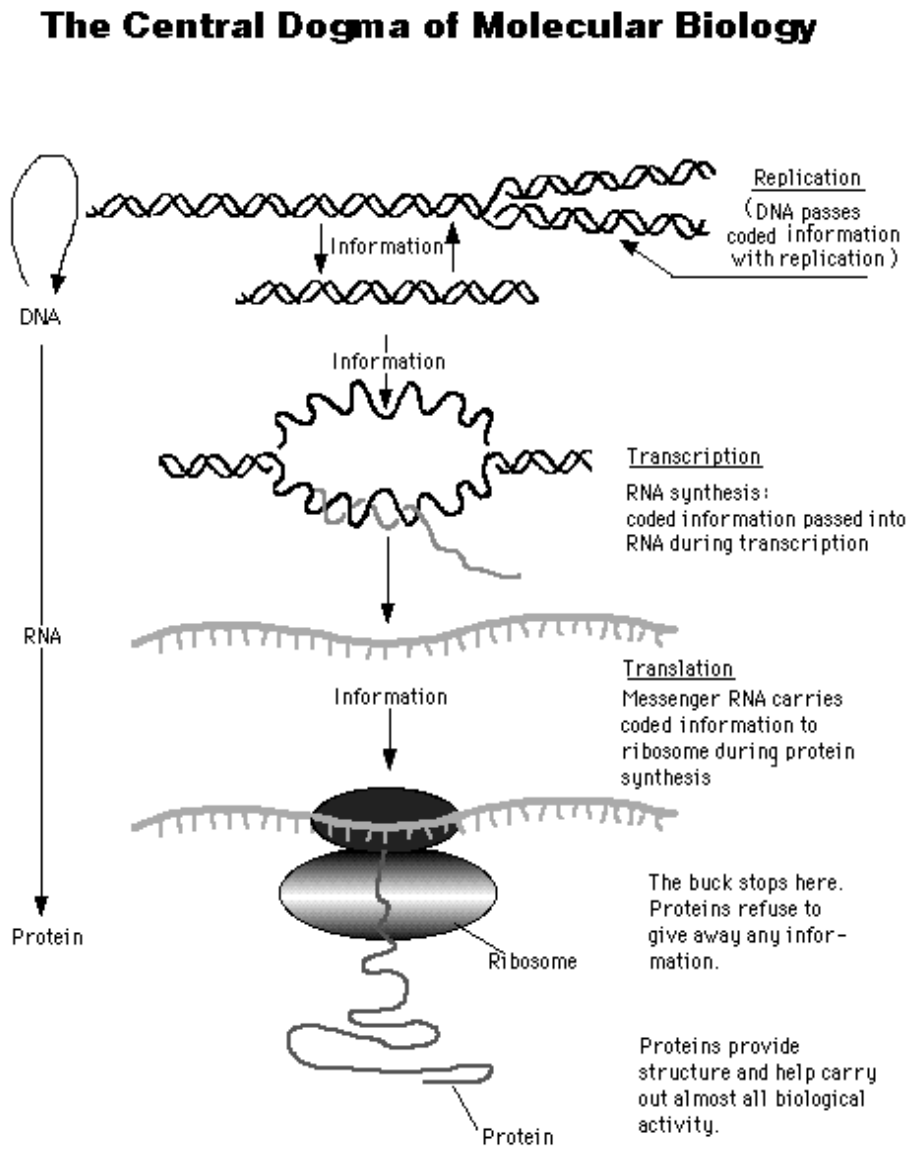
Η πληροφορία που περιέχει το DNA καθεαυτή μεταφέρεται με μια διαδικασία που ονομάζεται μεταγραφή. Στη διαδικασία αυτή δημιουργείται ένα δεύτερο είδος νουκλεϊκού οξέως, το ριβονουκλεϊκό οξύ (RNA). Το RNA διαφέρει από το DNA στο ότι το σάκχαρο που περιέχει είναι η ριβόζη και όχι η δεοξυριβόζη. Επιπλέον αντί της βάσης θυμίνη περιέχει τη βάση ουρακίλη. Η βάση αυτή είναι συμπληρωματική με την αδενίνη κατά τρόπο αντίστοιχο όπως είναι η θυμίνη στο DNA. Το RNA που παράγεται από το DNA λειτουργεί ως αγγελιαφόρος. Για το λόγο αυτό ονομάζεται αγγελιαφόρο RNA (messenger RNA ή mRNA).

Η πληροφορία του DNA που μεταφέρεται με τη μεταγραφή στο RNA αποκωδικοποιείται με μια διαδικασία που ονομάζεται μετάφραση. Στη διαδικασία αυτή παράγονται οι πρωτεΐνες. Η διαδικασία αυτή λαμβάνει χώρα σε σωματίδια του κυττάρου που ονομάζονται ριβοσώματα. Τα ριβοσώματα «διαβάζουν» την πληροφορία του mRNA και πραγματοποιούν την πρωτεϊνοσύνθεση.

Οι πρωτεΐνες αποτελούνται από μόρια που ονομάζονται αμινοξέα. Σε αντίθεση με τη συμπληρωματικότητα των DNA και RNA όπου κάθε βάση αντιστοιχεί σε μια άλλη βάση, δεν έχουμε αντιστοίχιση ένα προς ένα μεταξύ των βάσεων του mRNA και των αμινοξέων. Αντίθετα, ένα αμινοξύ κωδικοποιείται από μια τριάδα βάσεων (η τριπλέτα αυτή ονομάζεται κωδικόνιο).

Οι πρωτεΐνες **δεν** κωδικοποιούν άλλες πρωτεΐνες, ή μόρια RNA, ή DNA. Η ροή της γενετικής πληροφορίας από το DNA στο RNA και από το RNA σε πρωτεΐνες, δηλαδή από το γενετικό υλικό στο σχηματισμό πρωτεϊνών, είναι γνωστή ως το κεντρικό δόγμα της μοριακής βιολογίας. Διατυπωμένο διαφορετικά, με τα λόγια του Francis Crick: «Από τη στιγμή που η πληροφορία θα περάσει στις πρωτεΐνες, δεν εξέρχεται ξανά από αυτές – ρέει μονόδρομα».

Σχηματικά το κεντρικό δόγμα της μοριακής βιολογίας δίνεται στην εικόνα 1.1.



1.1 Ροή γενετικής πληροφορίας

1.1.2 Τι είναι τα micro RNAs;

Έχει αποδειχθεί ότι δεν αποτελεί όλο το DNA πληροφορία που κωδικοποιεί πρωτεΐνες. Περίπου 2% μόνο από τα δισεκατομμύρια ζεύγη βάσεων του γενετικού υλικού αποτελούν περιοχές που μεταγράφονται σε mRNA το οποίο μεταφράζεται σε πρωτεΐνες.

Τα micro RNAs είναι μικρά μόρια RNA με μήκος ακολουθίας γύρω στις 22 βάσεις τα οποία δεν κωδικοποιούν πρωτεΐνες. Πρόκειται για μόρια RNA, τα οποία ελέγχουν την

έκφραση γονιδίων, μέσω της παρεμπόδισης της διαδικασίας μετάφρασης των μορίων mRNA σε πρωτεΐνες. Η έρευνα οδηγεί στο συμπέρασμα ότι υπάρχει άμεση συσχέτιση μεταξύ της λειτουργίας των micro RNAs - ως ρυθμιστικών παραγόντων της έκφρασης πρωτεϊνών - και της εμφάνισης ασθενειών. Συσχέτιση εμφανίζεται για παράδειγμα μεταξύ της λειτουργίας των micro RNA και διαφόρων ειδών καρκίνου, ενώ φαίνεται να ευθύνονται και για ορισμένες γενετικές ασθένειες.

Τα micro RNA ακολουθούν μια σύνθετη διαδικασία μεταγραφής, έως ότου προκύψει το τελικό προϊόν. Αρχικά τα micro RNAs προκύπτουν ως τμήμα ενός μεγαλύτερου προϊόντος μεταγραφής. Το αρχικό αυτό προϊόν μπορεί να έχει μήκος της τάξης των χιλίων βάσεων. Στη συνέχεια περνάει από μια σειρά βημάτων ωρίμανσης, μέχρι να προκύψει το τελικό micro RNA με μήκος περί τα 22 νουκλεοτίδια.

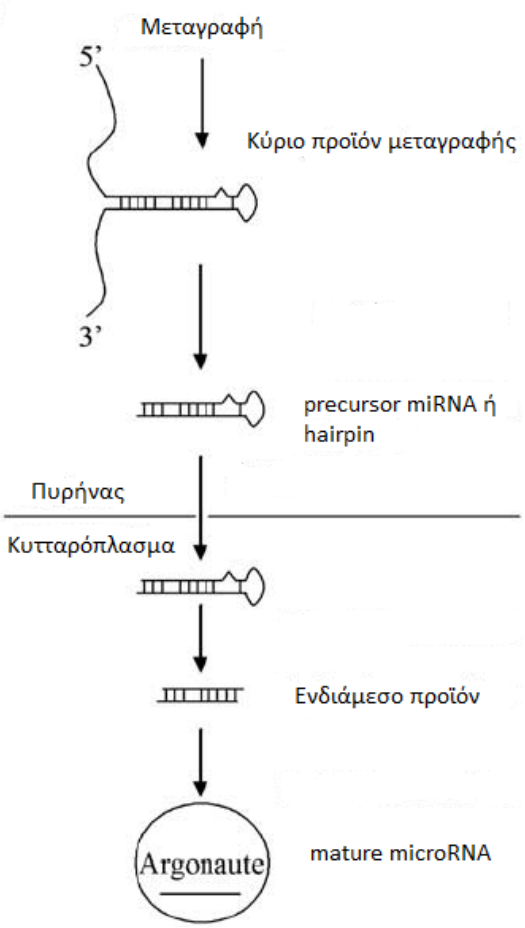
1.1.2.1 Hairpins

Αρχικά παράγεται στον πυρήνα του κυττάρου ένα micro RNA τύπου hairpin. Αυτό έχει μήκος περί τα 60 νουκλεοτίδια. Παίρνει την ονομασία του από το σχήμα του, όπως φαίνεται στην εικόνα 1.2. Το μόριο αυτό ονομάζεται και precursor miRNA, ή pre-miRNA. Τα μόρια αυτά αφού παραχθούν στον πυρήνα του κυττάρου εξάγονται στο κυτταρόπλασμα.

1.1.2.2 Mature micro RNAs

Στο κυτταρόπλασμα παράγεται από το hairpin ένα ενδιάμεσο προϊόν, το οποίο αποτελείται από δύο κλώνους ακολουθιών νουκλεοτιδίων. Τελικά απομονώνονται οι δύο αυτοί κλώνοι και δημιουργείται το τελικό ώριμο προϊόν, το οποίο ονομάζουμε mature micro RNA. Συνήθως εντοπίζεται ως ώριμο τελικό προϊόν στο κύτταρο ο ένας από τους δύο αυτούς κλώνους.

Τα mature micro RNA τελικά μεταφέρονται στις πρωτεΐνες καταστολείς (Argonaute) και χρησιμεύουν ως καθοδηγητές της καταστολής της δράσης των μορίων RNA.



1.2 Μηχανισμός ωρίμανσης micro RNA

1.1.3 Το πρόβλημα της εξέλιξης δεδομένων στην έρευνα βιομορίων

Στην ερευνητική διαδικασία της μοριακής βιολογίας έχουν αναπτυχθεί διάφορες βάσεις δεδομένων στις οποίες καταγράφονται τα ευρήματα των βιολόγων σε συγκεντρωμένη και οργανωμένη μορφή. Τέτοιες δομές δε θα μπορούσαν να λείπουν και για τα micro RNAs, διότι παρά τη σχετικά πρόσφατη ανακάλυψή τους, αποτελούν έναν σημαντικό ερευνητικό τομέα.

Στη μελέτη ενδιαφέρουν τόσο τα μόρια τύπου hairpin, όσο και τα ώριμα mature micro RNA. Στις βάσεις δεδομένων καταγράφονται πληροφορίες που αφορούν τα μόρια αυτά. Από τις πληροφορίες αυτές οι σημαντικότερες αφορούν το όνομα ενός βιομορίου και την ακολουθία νουκλεοτιδίων από την οποία αποτελείται. Καθώς προχωράει η έρευνα και προκύπτουν νέα δεδομένα, μπορεί να αλλάζουν, ή να αναθεωρούνται τα δεδομένα που καταγράφονται. Εμφανίζονται επιπλέον καινούριες εγγραφές στη βάση, ενώ παλιότερες μπορεί να διαγράφονται.

Οι ερευνητές βιολόγοι κάνοντας χρήση των διαδικτυακών εφαρμογών, ή βάσεων δεδομένων που αφορούν τα micro RNAs στην έρευνά τους, ενδέχεται να μη βρίσκουν τα δεδομένα που τους ενδιαφέρουν επειδή για παράδειγμα άλλαξε η καταγεγραμμένη ακολουθία νουκλεοτιδίων, ή το όνομα ενός τέτοιου βιομορίου. Επιπλέον αναζητώντας δημοσιεύσεις για ένα micro RNA μπορεί να χάνουν πληροφορίες που το αφορούν, επειδή σε διαφορετικές δημοσιεύσεις το ίδιο μόριο εμφανίζεται με άλλο όνομα. Προκύπτει έτσι η ανάγκη για την καταγραφή της εξέλιξης των δεδομένων που πραγματοποιείται καθώς οι εφαρμογές και οι βάσεις δεδομένων ανανεώνονται και εμπλουτίζονται με νέα στοιχεία.

1.2 Αντικείμενο της διπλωματικής

Η ιστοσελίδα DIANA (http://diana.cslab.ece.ntua.gr/) του ερευνητικού κέντρου Αλέξανδρος Φλέμινγκ διαθέτει μια σειρά εφαρμογών, οι οποίες χρησιμεύουν στην έρευνα γύρω από τα βιομόρια τύπου micro RNA. Επειδή οι εφαρμογές διαθέτουν δεδομένα για micro RNAs όπως καταγράφονται κάποια συγκεκριμένη στιγμή, ενδέχεται να δημιουργούνται προβλήματα στην έρευνα όταν τα δεδομένα μεταβάλλονται. Η παρούσα διπλωματική εργασία έχει ως αντικείμενο την δημιουργία μιας δομής (βάσης δεδομένων) που έρχεται να αντιμετωπίσει το πρόβλημα αυτό. Στοχεύει επιπλέον στην δημιουργία εφαρμογής που με χρήση της παραπάνω δομής θα μπορεί να παρουσιάζει συνοπτικά στον ερευνητή την εξέλιξη και τις μεταβολές των καταγεγραμμένων δεδομένων των micro RNA που τον ενδιαφέρουν.

1.2.1 Συνεισφορά

Η εφαρμογή που αναπτύχθηκε στα πλαίσια της εργασίας και οι απαραίτητες δομές στις οποίες βασίζεται συνεισφέρουν στις δυνατότητες της ιστοσελίδας DIANA με τους ακόλουθους τρόπους:

- Έχει δημιουργηθεί μια βάση δεδομένων που καταγράφει όλες τις μεταβολές των δεδομένων για micro RNAs που αποθηκεύονται στην βάση mirbase.
- Πληροφορία στην παραπάνω βάση που μπορεί να χρησιμοποιηθεί σε εφαρμογές πέρα από αυτήν της προβολής γράφων εξέλιξης δεδομένων.
- Διατίθενται scripts για την αναβάθμιση της βάσης αυτής, όταν κυκλοφορούν νέες εκδόσεις των βάσεων που καταγράφουν micro RNAs.

- Δημιουργία εφαρμογής που προβάλλει γράφους εξέλιξης δεδομένων στα πλαίσια της πλατφόρμας Yii, η οποία χρησιμοποιείται από τις εφαρμογές της ιστοσελίδας. Η εφαρμογή αυτή είναι επαναχρησιμοποιήσιμη ως τμήμα άλλων λογισμικών που διαθέτει η ιστοσελίδα.

1.3 Οργάνωση κειμένου

Η οργάνωση του κειμένου της διπλωματικής εργασίας έχει γίνει με βάση τη σειρά μελέτης και υλοποίησης των συστημάτων που απαιτούνταν για τη συγγραφή της τελικής εφαρμογής. Συνοπτικά τα κεφάλαια που ακολουθούν αναφέρονται στα εξής αντικείμενα:

Στο κεφάλαιο 2 περιγράφεται η βάση δεδομένων mirbase που αποθηκεύει δεδομένα για μόρια micro RNA και από την οποία αντλήσαμε την απαιτούμενη πληροφορία. Περιγράφονται οι τύποι αρχείων της και οι πληροφορίες που μπορούν να αντληθούν από αυτά και η μοντελοποίηση των αλλαγών που μπορούν να παρατηρηθούν για μια εγγραφή, ενώ γίνεται συνοπτική αναφορά στην εφαρμογή DIANA microT v4.0 και στην βάση δεδομένων ensEMBL.

Στο κεφάλαιο 3 περιγράφονται τα βήματα σχεδίασης και υλοποίησης της βάσης δεδομένων που θα αποθηκεύει το ιστορικό των αλλαγών για όλες τις εγγραφές. Περιγράφονται επίσης τα προγράμματα που αναβαθμίζουν τη βάση και ελέγχουν διαφορές σε σχέση με τα στοιχεία που δίνει η mirbase. Τέλος δίνονται ορισμένα σενάρια χρήσης ενός διαχειριστή της βάσης, τα οποία παρουσιάζουν την πληροφορία που μπορεί να αντληθεί από αυτήν.

Στο κεφάλαιο 4 περιγράφονται οι λειτουργικές απαιτήσεις της καθεαυτό εφαρμογής προβολής γράφων εξέλιξης δεδομένων. Δίνονται οι τεχνολογίες και οι πλατφόρμες που χρησιμοποιήθηκαν για την ανάπτυξή της και περιγράφονται επιγραμματικά οι βασικοί αλγόριθμοι που υλοποιούν τις λειτουργίες της.

Τέλος, στο κεφάλαιο 5 γίνεται αναφορά σε ζητήματα που μένουν ακόμη ανοιχτά. Αυτά αφορούν πεδία ανάπτυξης μελλοντικών εφαρμογών, η επέκτασης αυτών που ήδη υπάρχουν.

2

Θεωρητικό υπόβαθρο και σχετικές εργασίες

2.1 Η βάση βιολογικών δεδομένων mirbase

Η βάση δεδομένων mirbase (http://www.mirbase.org/) αποτελεί αρχείο για δεδομένα που αφορούν αλληλουχίες δημοσιευμένων micro RNAs και το σχολιασμό τους. Η ιστοσελίδα δίνει τη δυνατότητα αναζήτησης και προβολής των πληροφοριών που έχουν καταγραφεί και δημοσιευτεί τόσο για hairpins, όσο και για mature micro RNAs. Παράλληλα μπορεί κανείς να κατεβάσει το αρχειακό της υλικό ανά έκδοση. Σύμφωνα με τους Kozomara και Griffiths-Jones η βάση δεδομένων έχει τους εξής στόχους:

- Να ορίσει ένα σχήμα ονοματολογίας για micro RNAs.
- Να αποτελέσει τράπεζα δεδομένων για δημοσιευμένες αλληλουχίες micro RNAs, καθώς και να διευκολύνει τη διαδικτυακή αναζήτηση και το κατέβασμα όλων των δεδομένων τους.
- Να παρέχει πληροφορίες για της αλληλουχίες των micro RNAs, οι οποίες μπορούν να διαβάζονται από ανθρώπους και υπολογιστές.

- Να παρέχει πρόσβαση στα κύρια αποδεικτικά στοιχεία που αφορούν τα καταγεγραμμένα micro RNAs.
- Να παρέχει συνδέσμους και να συγκεντρώνει πληροφορίες που αφορούν στόχους των microRNA.

Η ιστοσελίδα δημιουργήθηκε το 2002 και συντηρείται από το τμήμα Faculty of Life Sciences του πανεπιστημίου του Manchester. Μέχρι σήμερα έχουν κυκλοφορήσει 33 εκδόσεις της βάσης, ξεκινώντας από την έκδοση 1.0 μέχρι την τρέχουσα έκδοση 18.0. Η τρέχουσα έκδοση περιέχει 18226 εγγραφές που αφορούν hairpins και 21643 εγγραφές που αφορούν mature micro RNAs. Τα micro RNAs αυτά αφορούν 168 είδη ζωντανών οργανισμών. Όλα τα δεδομένα είναι διαθέσιμα σε μορφή αρχείων κειμένου στο σύνδεσμο ftp://mirbase.org/pub/mirbase/.

2.2 Αρχεία δεδομένων της βάσης mirbase

Τα αρχεία εκδόσεων της mirbase που μπορεί κανείς να κατεβάσει αποτελούνται από διάφορους τύπους αρχείων. Τα είδη των αρχείων δεν είναι απαραιτήτως ίδια σε κάθε έκδοση. Οι τύποι αρχείων που υπάρχουν συνολικά είναι οι ακόλουθοι: αρχεία .dat, .str, mature.fa, precursor.fa, hairpin.fa, αρχεία .diff, .dead, mifam.txt ή mifam.list, dme_marks_targets.txt, miRNA.xls, maturestar.fa, organisms.txt. Για τους τύπους αυτούς ισχύουν τα παρακάτω:

1. Ο τύπος αρχείων .dat είναι ο μόνος που εμφανίζεται σε όλες τις εκδόσεις ως τώρα. Τα αρχεία αυτά περιέχουν όλη την πληροφορία για τα hairpins της κάθε έκδοσης. Η πληροφορία αυτή είναι κωδικοποιημένη σε μια μορφή γνωστή ως EMBL format.

2. Τα αρχεία .str εμφανίζονται σε όλες τις εκδόσεις της mirbase εκτός από την 1.0 και την 3.0. Τα αρχεία αυτά περιέχουν σε μορφή κειμένου εγγραφές που αποτελούνται από το όνομα ενός hairpin και τη σχετική ακολουθία βάσεων. Περιλαμβάνεται επίσης πληροφορία για τα mature micro RNAs που παράγονται από αυτό το hairpin.

3. Το αρχείο mature.fa υπάρχει σε όλες τις εκδόσεις από την 1.1 και έπειτα. Περιέχει πληροφορίες για όλες τις ακολουθίες mature miRNA μιας έκδοσης. Η πληροφορία αυτή κωδικοποιείται σε μια μορφή που είναι γνωστή ως FASTA format.

4. Τα αρχεία precursor.fa υπάρχουν από την έκδοση 1.1 ως την έκδοση 3.0 και περιέχουν πληροφορία για την προβλεπόμενη ακολουθία των hairpins σε FASTA format. Από την έκδοση 3.1 και έπειτα έχουμε τα αρχεία hairpin.fa στη θέση των αρχείων precursor.fa τα οποία περιέχουν σε FASTA format τις ακολουθίες για όλα τα hairpins.

5. Τα αρχεία .diff εμφανίζονται από την έκδοση 4.0 και έπειτα και περιέχουν πληροφορίες για τις αλλαγές που υπέστησαν οι διάφορες εγγραφές των hairpins, ή mature σε σχέση με την προηγούμενη έκδοση.

6. Τα αρχεία .dead εμφανίζονται από την έκδοση 5.0 και έπειτα και αποτελούν μια λίστα των εγγραφών που αφορούν hairpins, τα οποία διαγράφηκαν από τη βάση δεδομένων, ή προωθήθηκαν σε άλλη εγγραφή. Παράλληλα περιέχουν πληροφορία για το λόγο της διαγραφής, ή προώθησης.

7. Τα αρχεία mifam.txt/mifam.list εμφανίζονται από την έκδοση 5.0 και έπειτα και περιέχουν πληροφορία για την κατηγοριοποίηση ακολουθιών hairpin των miRNA σε οικογένειες.

8. Τα αρχεία dme_marks_targets.txt εμφανίζονται μόνο στις εκδόσεις 5.1 και 6.0. Δεν αναφέρεται εξήγηση για το περιεχόμενό τους στα αρχεία τεκμηρίωσης των εκδόσεων. Φαίνεται να αφορούν microRNAs του οργανισμού "Drosophila melanogaster" και τα γονίδια- στόχους αυτών.

9. Τα αρχεία miRNA.xls είναι αρχεία exel που περιέχουν όλη την πληροφορία που αφορά το όνομα ενός hairpin, τα ονόματα και τις ακολουθίες που αντιστοιχούν στα mature miRNAs που παράγονται από αυτά, καθώς και μια ένδειξη για τυχόν αλλαγή που έγινε σε σχέση με προηγούμενες εκδόσεις. Τα αρχεία αυτά υπάρχουν σε όλες τις εκδόσεις από την 7.0 και έπειτα, με εξαίρεση την έκδοση 7.1.

10. Τα αρχεία maturestar.fa περιέχουν πληροφορία σε FASTA format που αφορά mature miRNAs, των οποίων το όνομα περιέχει τον χαρακτήρα *. Ο χαρακτήρας αυτός για κάποιο διάστημα χρησιμοποιούνταν για να δείξει ότι ένα παραγόμενο mature miRNA από ένα hairpin, αποτελούσε το δευτερεύον προϊόν του hairpin αυτού. Συνεπώς σήμαινε ότι το mature βρίσκεται σε μικρότερη συγκέντρωση σε ένα κύτταρο σε σχέση με το κύριο προϊόν του hairpin και θεωρούνταν συνήθως μη λειτουργικό. Η αντίληψη αυτή για το «δευτερεύον» προϊόν έχει πλέον αλλάξει

και συνεπώς και ο τρόπος ονοματολογίας των mature που παράγονται από τα hairpins. Τα αρχεία αυτά συνεπώς έχουν εγκαταλειφτεί πλέον. Υπάρχουν από την έκδοση 10.0 ως την έκδοση 14.

11. Τέλος, τα αρχεία organisms.txt εμφανίζονται από την έκδοση 12.0 και έπειτα και περιέχουν σε κείμενο πληροφορίες για τους οργανισμούς των οποίων τα miRNAs υπάρχουν στην τρέχουσα έκδοση της βάσης.

Για τους σκοπούς της ανάλυσης μας θα εξετάσουμε στη συνέχεια τους τύπους αρχείων .dat, .fa, .diff και .dead.

2.2.1 Αρχεία .dat

2.2.1.1 Το format δεδομένων EMBL

Για την καταγραφή της πληροφορίας που αφορά ακολουθίες νουκλεοτιδίων (γενετικό υλικό) σε αρχεία κειμένου μπορούν να χρησιμοποιηθούν διάφοροι τρόποι οργάνωσης της. Ένας τέτοιος τρόπος είναι η μορφή EMBL (EMBL format). Το EMBL format χρησιμοποιείται από το European Molecular Biology Laboratory του European Bioinformatics Institute στην τράπεζα δεδομένων γονιδίων EMBL- Bank. Αυτή αποτελεί ως βάση δεδομένων τον κύριο πόρο για πληροφορίες ακολουθιών νουκλεοτιδίων στην Ευρώπη.

Η πληροφορία που συσχετίζεται με κάθε καταγεγραμμένο γονίδιο, καθώς και η ίδια η αλληλουχία νουκλεοτιδίων του γονιδίου αποτελούν μια ξεχωριστή εγγραφή σε κάθε αρχείο κειμένου που γράφεται σε EMBL format. Τέτοιου τύπου πληροφορία μπορεί να είναι το όνομα ενός γονιδίου, ένας κωδικός που αντιστοιχεί στο συγκεκριμένο γονίδιο, μια σύντομη περιγραφή του γονιδίου, η ακολουθία καθεαυτή κ.α. Η πληροφορία για κάθε εγγραφή δομείται με τέτοιο τρόπο ώστε να είναι εύκολη η ανάγνωση τόσο από ανθρώπους, όσο και από προγράμματα υπολογιστή. Πληροφορίες όπως οι εξηγήσεις, οι περιγραφές, οι κατηγοριοποιήσεις και τα σχόλια καταγράφονται σε φυσική γλώσσα. Τα σύμβολα και η διάταξη που χρησιμοποιούνται για τις ακολουθίες βάσεων έχουν επιλεγεί έτσι ώστε να είναι ευανάγνωστα. Όπου είναι δυνατό, γίνεται χρήση συμβόλων με τα οποία είναι εξοικειωμένοι οι μοριακοί βιολόγοι. Ταυτόχρονα κάθε εγγραφή είναι δομημένη έτσι ώστε ένα πρόγραμμα να μπορεί εύκολα να διαβάζει, να αναγνωρίζει και να μεταχειρίζεται τους διάφορους τύπους δεδομένων που αποθηκεύονται.

Κάθε εγγραφή αποτελείται από ένα σύνολο γραμμών κειμένου. Για την καταγραφή των διαφόρων τύπων δεδομένων που αποτελούν μια εγγραφή χρησιμοποιούνται διαφορετικών ειδών γραμμές κειμένου. Ο διαχωρισμός των γραμμών αυτών γίνεται με βάση έναν κωδικό στην αρχή τους. Γενικά αποφεύγεται η χρήση κάποιας σταθερής διάταξης όσο αφορά την τοποθέτηση δεδομένων σε στήλες. Υπάρχουν δύο εξαιρέσεις σε αυτόν τον κανόνα: Οι γραμμές που αναφέρονται στην ακολουθία βάσεων και οι γραμμές FT (feature table). Όλες οι άλλες γραμμές μπορούν να έχουν οποιαδήποτε διάταξη.

Κάθε γραμμή ξεκινά με ένα κωδικό δύο χαρακτήρων, ο οποίος υποδηλώνει το είδος της πληροφορίας που εμπεριέχεται σε αυτήν. Τα είδη γραμμών που χρησιμοποιούνται αυτή τη στιγμή στο EMBL format, μαζί με τους αντίστοιχους κωδικούς τους παρουσιάζονται παρακάτω.

ID - αναγνωριστικό (όνομα)	(Βρίσκεται στην αρχή κάθε εγγραφής. 1 ανά εγγραφή.)
AC - αλφαριθμητικό κωδικού (accession)	(>=1 ανά εγγραφή)
PR - αναγνωριστικό έργου	(0 ή 1 ανά εγγραφή)
DT - ημερομηνία	(2 ανά εγγραφή)
DE - περιγραφή	(>=1 ανά εγγραφή)
KW - λέξη-κλειδί	(>=1 ανά εγγραφή)
OS - είδος οργανισμού	(>=1 ανά εγγραφή)
OC - κατηγορία οργανισμού	(>=1 ανά εγγραφή)
OG - οργανίδιο	(0 ή 1 ανά εγγραφή)
RN - αριθμός παραπομπής	(>=1 ανά εγγραφή)
RC - σχόλιο παραπομπής	(>=0 ανά εγγραφή)
RP - θέση παραπομπής	(>=1 ανά εγγραφή)
RX - παραπομπή σε cross-reference	(>=0 ανά εγγραφή)
RG - ομάδα παραπομπής	(>=0 ανά εγγραφή)
RA - συγγραφείς παραπομπής	(>=0 ανά εγγραφή)
RT - τίτλος παραπομπής	(>=1 ανά εγγραφή)
RL - τοποθεσία παραπομπής	(>=1 ανά εγγραφή)
DR - cross-reference βάσεων δεδομένων	(>=0 ανά εγγραφή)
CC - σχόλια ή σημειώσεις	(>=0 ανά εγγραφή)
AH - assembly header	(0 ή 1 ανά εγγραφή)
AS - assembly information	(0 ή >=1 ανά εγγραφή)
FH - επικεφαλίδα του πίνακα χαρακτηριστικών	(2 ανά εγγραφή)

FT - δεδομένα του πίνακα χαρακτηριστικών	(>=2 ανά εγγραφή)
XX - διαχωριστική γραμμή	(πολλά ανά εγγραφή)
SQ - επικεφαλίδα ακολουθίας	(1 ανά εγγραφή)
CO - contig/construct line	(0 ή >=1 ανά εγγραφή)
bb - (κενά) δεδομένα ακολουθίας	(>=1 ανά εγγραφή)
// - γραμμή τερματισμού	(Στο τέλος κάθε εγγραφής. 1 ανά εγγραφή)

Σημειώνεται ότι δεν περιέχουν όλες οι εγγραφές κάθε τύπο γραμμής. Επίσης κάποιοι τύποι γραμμών εμφανίζονται περισσότερες φορές σε μια εγγραφή. Κάθε εγγραφή ξεκινάει με μια γραμμή ID και τελειώνει με μια γραμμή τερματισμού (//). Τα διάφορα είδη γραμμών εμφανίζονται σε κάθε εγγραφή με τη σειρά με την οποία δίνονται παραπάνω σε αρχεία που έχουν το EMBL format. Εξαίρεση αποτελούν οι γραμμές XX, οι οποίες εμφανίζονται οπουδήποτε μεταξύ των γραμμών ID και SQ και χρησιμεύουν ως διαχωριστικό διαφορετικών τύπων γραμμών.

2.2.1.2 Πληροφορία EMBL αρχείων .dat

Τα αρχεία .dat της mirbase περιέχουν πληροφορία για καταγεγραμμένα micro RNAs σε μια μορφή που ορίζεται ως **σχεδόν** EMBL. Συνεπώς, στις εγγραφές της mirbase δεν εμφανίζονται όλοι οι τύποι γραμμών που περιγράφθηκαν στην προηγούμενη παράγραφο. Επιπλέον κάποιοι τύποι γραμμών παρουσιάζονται με διαφορετική σειρά.

Οι εγγραφές της mirbase από την έκδοση 1.2 μέχρι και την τρέχουσα χρησιμοποιούν τις γραμμές:

ID, AC, DE, RN, RX, RA, RT, RL, RC, DR, CC, FH, FT, XX, SQ, bb, //

Σημειώνεται ότι η διαφορά σε σχέση με προηγούμενες εκδόσεις είναι ότι στις εκδόσεις 1.0 και 1.1 δεν εμφανίζονται γραμμές CC.

Στη συνέχεια παρουσιάζεται η μορφή και τα δεδομένα των γραμμών που εμφανίζονται στα αρχεία .dat της mirbase.

Η γραμμή ID

Η γραμμή ID είναι πάντοτε η πρώτη γραμμή μιας εγγραφής. Η μορφή μιας γραμμής ID με βάση το format της mirbase είναι:

ID <1> <2>; <3>; <4>; <5> BP.

Όπου στα παραπάνω οι αριθμοί αντιστοιχούν στην ακόλουθη πληροφορία:

- <1>: Το όνομα της εγγραφής
- <2>: Τοπολογία
- <3>: Είδος μορίου
- <4>: Ταξινομική βαθμίδα.
- <5>: Μήκος ακολουθίας.

Στο σημείο <4> μπορεί να βρίσκεται μια από τις τιμές που βρίσκονται στο αρχείο ftp://mirbase.org/pub/mirbase/CURRENT/organisms.txt.

Η γραμμή AC

Η γραμμή AC δίνει ένα κωδικό αλφαριθμητικό που υπάρχει για κάθε εγγραφή και ονομάζεται accession number. Ο κωδικός αυτός είναι μοναδικός για κάθε εγγραφή. Ο αριθμός accession είναι το μόνο πραγματικό αναγνωριστικό για μια εγγραφή. Τα ονόματα των miRNA (ID) μπορεί να αλλάξουν σε σχέση με αυτά που είναι δημοσιευμένα, καθώς ξεκαθαρίζουν οι σχέσεις μεταξύ ακολουθιών. Το πλεονέκτημα του συστήματος που χρησιμοποιεί accession number είναι ότι τέτοιες αλλαγές μπορούν να παρακολουθηθούν στη βάση δεδομένων, επιτρέποντας την εξέλιξη ονομάτων, παρέχοντας στον χρήστη πλήρη πρόσβαση στα δεδομένα και την ιστορία. Από την άλλη τα accessions δεν περιέχουν πολλή βιολογική πληροφορία. Για το λόγο αυτό οι δημοσιεύσεις που αφορούν κάποιο micro RNA πρέπει να χρησιμοποιούν κάθε φορά το ID του. Το τελευταίο εμπεριέχει πληροφορία διάφορων τύπων, όπως π.χ. ο οργανισμός στον οποίο εντοπίζεται.

Παράδειγμα μιας γραμμής AC είναι το παρακάτω:

```
AC  MI0000001;
```

Η γραμμή DE

Η γραμμή DE αποτελεί μια συνοπτική περιγραφή της ακολουθίας μιας εγγραφής. Οι πληροφορίες της γραμμής μπορεί να αναφέρονται σε προσδιορισμούς των γονιδίων που κωδικοποιεί η ακολουθία, την περιοχή του γονιδιώματος από την οποία παράγεται η ακολουθία ή κάποια άλλη πληροφορία που βοηθά στην αναγνώριση της ακολουθίας. Ένα παράδειγμα μιας γραμμής DE από ένα αρχείο .dat που αντιστοιχεί σε εγγραφή με ID cel-let-7 δίνεται παρακάτω:

```
DE   Caenorhabditis elegans let-7 stem-loop
```

Η περιγραφή δίνεται σε φυσική γλώσσα (αγγλικά) και έχει ελεύθερη διάταξη. Μπορεί να απαιτούνται περισσότερες από μία γραμμές DE. Συνήθως η πρώτη γραμμή DE περιέχει μια σύντομη περιγραφή που μπορεί να χρησιμοποιηθεί και μόνη για λόγους καταχώρησης.

Οι γραμμές αναφοράς (reference lines RN, RC, RX, RA, RT, RL)

Οι γραμμές αυτές αποτελούν τις βιβλιογραφικές παραπομπές που υπάρχουν στη βάση. Οι παραπομπές δίνουν πρόσβαση στις δημοσιεύσεις από τις οποίες έχουν ληφθεί τα δεδομένα. Οι γραμμές αναφοράς για μια δεδομένη παραπομπή εμφανίζονται σε ομάδα και έχουν πάντα τη σειρά RN, RX, RA, RT, RL, RC. Σε κάθε τέτοια ομάδα η γραμμή RN εμφανίζεται μια φορά, οι γραμμές RC και RX εμφανίζονται μηδέν, ή περισσότερες φορές, και οι γραμμές RA, RT, RL εμφανίζονται μία ή περισσότερες φορές. Αν έχουμε περισσότερες από μια αναφορές για μια εγγραφή θα υπάρχει μια τέτοια ομάδα γραμμών για κάθε αναφορά. Παράδειγμα αναφορών:

```
RN   [1]
RX   PUBMED; 12554860.
RA   Dostie J, Mourelatos Z, Yang M, Sharma A, Dreyfuss G;
RT   "Numerous microRNPs in neuronal cells containing novel microRNAs";
RL   RNA 9:180-186(2003).
RC   Erratum RNA 9:631-632(2003)
```

Ακολούθως περιγράφονται οι επιμέρους γραμμές.

Η γραμμή RN

Η γραμμή RN (reference number) αποδίδει ένα μοναδικό αριθμό για κάθε αναφορά σε παραπομπή μιας εγγραφής. Ο αριθμός αυτός χρησιμοποιείται για τον προσδιορισμό μιας παραπομπής στο τμήμα σχολίων (CC) της εγγραφής, ή στον τμήμα FT (feature table). Η διάταξη μιας γραμμής RN είναι:

```
RN   [n]
```

Στο παραπάνω το n είναι ένας αριθμός. Ο αριθμός παραπομπής βρίσκεται πάντοτε σε αγκύλες. Σημειώνουμε ότι το σύνολο των αριθμών παραπομπής σε μια εγγραφή δεν αποτελεί απαραίτητα μια συνεχόμενη ακολουθία από το 1 ως το n, όπου η εγγραφή θα πρέπει να περιέχει n αναφορές. Καθώς προσθαφαιρούνται παραπομπές σε μια εγγραφή, μπορεί να δημιουργηθούν κενά στην ακολουθία των αριθμών. Το σημαντικό σημείο είναι ότι από τη στιγμή που θα καταχωρηθεί ένας αριθμός για μια παραπομπή σε μια εγγραφή, αυτός ποτέ δεν αλλάζει.

Η γραμμή RX

Η γραμμή RX (reference cross-reference) είναι ένας προαιρετικός τύπος γραμμής που περιέχει μια παραπομπή προς κάποια εξωτερική δημοσίευση. Για παράδειγμα αν υπάρχει μια παραπομπή στην βάση δεδομένων PUBMED θα υπάρχει μια γραμμή RX που θα παραπέμπει προς το σχετικό αναγνωριστικό στην βάση PUBMED. Η διάταξη μιας τέτοιας γραμμής είναι:

```
RX resource identifier; identifier.
```

Το resource_identifier είναι μια συντομευμένη ονομασία της συλλογής δεδομένων στην οποία γίνεται η παραπομπή. Το τρέχον σύνολο εξωτερικών πηγών για αρχεία EMBL είναι:

Resource ID	Full name
PUBMED	PUBMED bibliographic database (NLM)
DOI	Digital Object Identifier (International DOI Foundation)
AGRICOLA	US National Agriculture Library (NAL) of the US Departmentof Agriculture (USDA)

Η τιμή του identifier είναι ένας δείκτης προς την καταχώρηση στην εξωτερική πηγή προς την οποία γίνεται η παραπομπή. Το δεδομένο που χρησιμοποιείται ως identifier εξαρτάται από την πηγή στην οποία γίνεται αναφορά. Για παράδειγμα:

```
RX   PUBMED; 12747828.
```

Η γραμμή RA

Η γραμμή RA (reference author) καταγράφει τους συγγραφείς της δημοσίευσης (ή οποιασδήποτε άλλης δουλειάς) που αναφέρεται. Όλοι οι συγγραφείς συμπεριλαμβάνονται και καταγράφονται με τη σειρά που δίνονται στη δημοσίευση. Τα ονόματα καταγράφονται με το επίθετο να προηγείται ακολουθούμενο από ένα

κενό και τα αρχικά του ονόματος. Ενίοτε τα αρχικά μπορεί να είναι άγνωστα, περίπτωση στην οποία καταγράφεται μόνο το επίθετο. Τα ονόματα των συγγραφέων διαχωρίζονται με κόμμα και τελειώνουν με το χαρακτήρα semicolon. Δεν γίνεται διαχωρισμός των ονομάτων κατά γραμμές. Ένα παράδειγμα γραμμής RA είναι:

```
RA   Ambros V, Lee RC, Lavanway A, Williams PT, Jewell D;
```

Περιλαμβάνονται πάντοτε τόσες γραμμές RA όσες χρειάζονται για κάθε αναφορά.

Η γραμμή RT

Η γραμμή RT (Reference Title) δίνει τον τίτλο της δημοσίευσης, (ή άλλης δουλειάς) όσο ακριβέστερα είναι δυνατό, δεδομένων των περιορισμών των συνόλων χαρακτήρων που χρησιμοποιούν οι υπολογιστές. Σημειώνεται ότι η μορφή του τίτλου που χρησιμοποιείται είναι αυτή που θα φαινόταν σε μια παραπομπή, παρά η επικεφαλίδα που αναφέρεται στον τίτλο της δημοσίευσης. Για παράδειγμα δεν διατηρούνται τα κεφαλαία στους τίτλους. Ο τίτλος βρίσκεται μέσα σε εισαγωγικά και μπορεί να συνεχιστεί για όσες γραμμές είναι απαραίτητο. Οι γραμμές του τίτλου τερματίζουν με το χαρακτήρα semicolon. Ως παράδειγμα τέτοιας γραμμής αναφέρεται το:

```
RT   "MicroRNAs and other tiny endogenous RNAs in C. elegans";
RL   Curr Biol 13:807-818(2003).
```

Τα ελληνικά γράμματα σε τίτλους διατυπώνονται ολογράφως. Για παράδειγμα ένας τίτλος μπορεί να περιέχει το "kappa-immunoglobulin" παρ' ότι στη δημοσίευση μπορεί να χρησιμοποιείται το γράμμα "K". Παρόμοιες απλοποιήσεις έχουν γίνει και σε άλλες περιπτώσεις (π.χ. υπογραμμίσεις). Σημειώνεται ότι μια γραμμή RT μιας παραπομπής που δεν έχει τίτλο (όπως μια καταχώρηση στη βάση δεδομένων) περιέχει μόνο το χαρακτήρα semicolon.

Η γραμμή RL

Η γραμμή RL (Reference Location) περιέχει την τυπική πληροφορία για την παραπομπή που γίνεται. Γενικά περιέχει το περιοδικό, τον αριθμό τόμου, τις σελίδες και το έτος για κάθε δημοσίευση. Τα ονόματα των περιοδικών συντομεύονται βάσει κανόνων ISO. Η διάταξη μιας γραμμής RL είναι:

```
RL   journal vol:pp-pp(year).
```

Περιστασιακά, κάποια περιοδικά δεν αριθμούν συνεχόμενα τις σελίδες ενός τόμου, αλλά ξεκινούν από την αρχή την αρίθμηση για κάθε έκδοση. Σε αυτή την περίπτωση ο αριθμός της έκδοσης πρέπει να συμπεριλαμβάνεται και η διάταξη γίνεται:

```
RL   journal vol(no):pp-pp(year).
```

Αν μια δημοσίευση δεν έχει γίνει ακόμα, η γραμμή εμφανίζεται με όση πληροφορία είναι διαθέσιμη. Όποια πληροφορία λείπει σημειώνεται με μηδενικά. Για παράδειγμα η γραμμή:

```
RL   Nucleic Acids Res. 0:0-0(2004)
```

κάνει αναφορά σε μια δημοσίευση που θα εκδοθεί στο περιοδικό Nucleic Acids Research κάποια στιγμή το 2004 και δεν υπάρχει πληροφορία για τον τόμο, ή τις σελίδες. Τέτοιες αναφορές ενημερώνονται ώστε να συμπεριλαμβάνουν και την πληροφορία που λείπει, όταν αυτή γίνει διαθέσιμη. Μια άλλη παραλλαγή της γραμμής RL χρησιμοποιείται για δημοσιεύσεις παρμένες από βιβλία, ή άλλες παρόμοιες δημοσιεύσεις, που καταγράφονται ως ακολούθως:

```
RA   Birnstiel M., Portmann R., Busslinger M., Schaffner W.,
RA   Probst E., Kressmeann A.;
RT   "Functional organization of the histone genes in the
RT   sea urchin Psammechinus:  A progress report";
RL   (in) Engberg J., Klenow H., Leick V. (Eds.);
RL   SPECIFIC EUKARYOTIC GENES:117-132;
RL   Munksgaard, Copenhagen (1979).
```

Εκεί που θα βρισκόταν η πληροφορία για την θέση στο περιοδικό, εμφανίζεται η βιβλιογραφική παραπομπή. Η πρώτη γραμμή RL σε αυτή τη περίπτωση περιέχει τον προσδιορισμό "(in)", που δείχνει ότι πρόκειται για αναφορά σε βιβλίο. Τα ακόλουθα παραδείγματα δείχνουν γραμμές RL που χρησιμοποιούνται για την περίπτωση που έχουν υποβληθεί δεδομένα.

```
RL   Submitted (19-NOV-1990) to the EMBL/GenBank/DDBJ databases.
RL   M.A. Hughes, UNIVERSITY OF NEWCASTLE UPON TYNE, MEDICAL SCHOOL, NEW
RL   CASTLE UPON TYNE, NE2  4HH, UK
```

Η διεύθυνση του ατόμου που συνέλεξε τα δεδομένα συμπεριλαμβάνεται σε καινούριες καταχωρήσεις, αλλά μερικές παλιότερες καταχωρήσεις δεν περιέχουν

αυτή τη πληροφορία. Οι γραμμές RL έχουν άλλη μορφή για αναφορές σε διατριβές. Για παράδειγμα:

```
RL   Thesis (1999), Department of Genetics,
RL   University of Cambridge, Cambridge, U.K.
```

Για μη δημοσιευμένες αναφορές, οι γραμμές RL έχουν την ακόλουθη μορφή:

```
RL   Unpublished.
```

Για αναφορές σε διπλώματα ευρεσιτεχνίας έχουν τη μορφή:

```
RL   Patent number EP0238993-A/3, 30-SEP-1987.
RL   BAYER AG.
```

Οι λέξεις "Patent number" ακολουθούνται από τον αριθμό αίτησης του διπλώματος ευρεσιτεχνίας, τον τύπο του διπλώματος ευρεσιτεχνίας (διαχωρισμένο με παύλες), το σειριακό αριθμό της ακολουθίας στο δίπλωμα ευρεσιτεχνίας (διαχωρισμένο με κάθετο) και την ημερομηνία αίτησης του διπλώματος ευρεσιτεχνίας. Οι γραμμές RL που ακολουθούν καταγράφουν τους αιτούντες για το δίπλωμα ευρεσιτεχνίας, που είναι συνήθως ονόματα εταιριών (όπως στο παραπάνω παράδειγμα).

Τέλος, για δημοσιεύσεις σε περιοδικά όπου δεν υπάρχει διαθέσιμος αριθμός ISSN, η γραμμή RL περιέχει την ένδειξη "(misc)" όπως στο ακόλουθο παράδειγμα:

```
RL   (misc) Proc. Vth Int. Symp. Biol. Terr. Isopods 2:365-380(2003).
```

Η γραμμή RC

Η γραμμή RC (Reference Comment) είναι μια προαιρετική γραμμή που εμφανίζεται αν υπάρχει σχόλιο στην παραπομπή. Το σχόλιο είναι γραμμένο στα Αγγλικά και χρησιμοποιούνται όσες γραμμές RC είναι απαραίτητες. Ένα παράδειγμα είναι το ακόλουθο:

```
RC   Erratum RNA 9:631-632(2003).
```

Η γραμμή DR

Η γραμμή DR (Database Cross-reference) αναφέρει εξωτερικές βάσεις δεδομένων (cross-reference) που περιέχουν πληροφορία που σχετίζεται με την εγγραφή στην οποία εμφανίζεται. Για παράδειγμα, αν υπάρχει μια ακολουθία της εγγραφής και σε

μια άλλη βάση, την IMGT/LIGM θα υπάρχει μια γραμμή DR που θα δείχνει προς τη σχετική καταχώρηση στην IMGT/LIGM. Η διάταξη μιας γραμμής DR είναι:

```
DR   database_identifier; primary identifier; secondary_identifier.
```

Το πρώτο στοιχείο, το αναγνωριστικό της βάσης, είναι ένα συντομευμένο όνομα της συλλογής δεδομένων στην οποία γίνεται αναφορά. Το δεύτερο στοιχείο, το κύριο αναγνωριστικό, είναι ένας δείκτης στην καταχώριση της εξωτερικής βάσης στην οποία γίνεται αναφορά. Το τρίτο στοιχείο στη γραμμή DR είναι το δευτερεύον αναγνωριστικό, αν υπάρχει, της αναφερόμενης βάσης δεδομένων. Ακολουθεί ένα παράδειγμα:

```
DR   EMBL; AJ550395; HSA550395.
```

Η γραμμή CC

Γραμμές CC είναι σχόλια σχετικά με την εγγραφή, σε μορφή κειμένου. Μπορούν να χρησιμοποιηθούν για να αποδώσουν οποιαδήποτε πληροφορία θεωρείται χρήσιμη και δεν μπορεί να εκφραστεί σε άλλους τύπους γραμμών.

Η γραμμή FH

Οι γραμμές FH (Feature Header) υπάρχουν μόνο για να διευκολύνουν την ανάγνωση μιας εγγραφής, όταν αυτή τυπώνεται σε χαρτί, ή προβάλλεται σε οθόνη υπολογιστή. Χρησιμοποιούνται ως επικεφαλίδες για το feature table που ακολουθεί και περιέχει χρήσιμη πληροφορία. Οι ίδιες οι γραμμές δεν περιέχουν δεδομένα και μπορούν να αγνοηθούν από προγράμματα υπολογιστή. Η μορφή αυτών των γραμμών είναι πάντοτε η ίδια:

```
FH   Key             Location/Qualifiers
FH
```

Η πρώτη γραμμή ορίζει τους τίτλους των στηλών ενός πίνακα χαρακτηριστικών και η δεύτερη γραμμή χρησιμεύει ως διαχωριστικό. Αν μια εγγραφή δεν περιέχει πίνακα χαρακτηριστικών (δηλαδή δεν υπάρχουν γραμμές FT, βλέπε παρακάτω), η γραμμές FH δεν εμφανίζονται.

Η γραμμή FT

Οι γραμμές FT (Feature Table) παρέχουν ένα μηχανισμό σχολιασμού των δεδομένων της ακολουθίας. Περιοχές, ή τοποθεσίες ενδιαφέροντος στην ακολουθία καταγράφονται στον πίνακα. Σε αρχεία EMBL, γενικά τα χαρακτηριστικά του πίνακα παριστάνουν σήματα, ή άλλα χαρακτηριστικά που αναφέρονται στις παραπομπές. Σε μερικές περιπτώσεις συμπεριλαμβάνονται αμφισημίες, ή χαρακτηριστικά που σημειώνονται κατά την προετοιμασία των δεδομένων. Ο πίνακας χαρακτηριστικών μπορεί να επεκταθεί, ή να αλλάξει, καθώς μαθαίνουμε περισσότερα για μια ακολουθία.

Στην περίπτωση των αρχείων της mirbase ο πίνακας χαρακτηριστικών περιέχει πληροφορίες για τα παραγόμενα mature micro RNAs ενός δεδομένου hairpin. Για παράδειγμα μπορούμε να έχουμε την ακόλουθη καταχώρηση πίνακα δεδομένων:

```
FH   Key             Location/Qualifiers
FH
FT   miRNA           17..38
FT                   /accession="MIMAT0000001"
FT                   /product="cel-let-7"
FT                   /evidence=experimental
FT                   /experiment="cloned [1-3,5], Northern [1], PCR [4], Solexa
FT                   [6], CLIPseq [7]"
FT   miRNA           56..80
FT                   /accession="MIMAT0015091"
FT                   /product="cel-let-7*"
FT                   /evidence=experimental
FT                   /experiment="CLIPseq [7]"
```

Η γραμμή SQ

Η γραμμή SQ (SeQuence header) σηματοδοτεί την αρχή των δεδομένων της ακολουθίας και δίνει μια περίληψη του περιεχομένου της. Ένα παράδειγμα είναι το:

```
SQ   Sequence 1859 BP; 609 A; 314 C; 355 G; 581 T; 0 other;
```

Όπως φαίνεται, η γραμμή αναφέρει το μήκος της ακολουθίας σε ζεύγη βάσεων. Αυτό ακολουθείται από τη σύσταση της ακολουθίας σε βάσεις. Βάσεις διάφορες των A, C, G και T ομαδοποιούνται ως "other". (Σημειώνεται ότι το "BP", δηλαδή base pair- ζεύγη βάσεων- χρησιμοποιείται επίσης για μονόκλωνες ακολουθίες RNA, πράγμα το οποίο δεν είναι αυστηρώς ακριβές, αλλά έχει χρησιμοποιηθεί αυτός ο τρόπος για λόγους συνοχής της διάταξης των γραμμών SQ). Η πληροφορία αυτή μπορεί να

χρησιμοποιηθεί ως έλεγχος ακρίβειας για στατιστικούς σκοπούς. Η λέξη "Sequence" εμφανίζεται μόνο ως ένδειξη για καλύτερη αναγνωσιμότητα.

Η γραμμή XX

Η γραμμή XX (spacer) δεν περιέχει δεδομένα, ή σχόλια. Χρησιμοποιείται για να κάνει μια καταχώρηση πιο ευανάγνωστη σε μια σελίδα, ή σε οθόνη, διαχωρίζοντας τα διάφορα είδη πληροφορίας σε κατάλληλες ομάδες. Το XX χρησιμοποιείται αντί για κενές γραμμές για να αποφευχθεί σύγχυση με τις γραμμές των δεδομένων της ακολουθίας βάσεων. Οι γραμμές XX μπορούν πάντοτε να αγνοηθούν από προγράμματα υπολογιστή.

Η γραμμή //

Η γραμμή // (γραμμή τερματισμού) επίσης δεν περιέχει δεδομένα, ή σχόλια. Χρησιμοποιείται για να δείξει το τέλος μιας εγγραφής.

Εγγραφές αρχείων .dat

Τα αρχεία .dat αποτελούνται από εγγραφές που αντιστοιχούν σε βιομόρια hairpins. Δεδομένου ότι τα mature micro RNAs παράγονται από hairpins, θα πρέπει να καταγράφεται πληροφορία και για τα παραγόμενα αυτά βιομόρια. Η πληροφορία αυτή δεν καταγράφεται με ξεχωριστές εγγραφές στα αρχεία .dat, αλλά ενσωματώνεται στις γραμμές FT, όπως αναφέρθηκε παραπάνω.

2.2.2 Αρχεία .fa

Τα αρχεία με την κατάληξη .fa περιέχουν πληροφορία για τα hairpins, ή τα mature micro RNAs σε μορφή FASTA. Περιγράφεται στη συνέχεια αυτή η μορφή αποθήκευσης δεδομένων σε κείμενο.

Το FASTA format

FASTA format είναι μια μορφή καταγραφής βασισμένη σε κείμενο για απεικόνιση είτε ακολουθιών νουκλεοτιδίων, ή ακολουθιών πεπτιδίων. Εδώ τα ζεύγη βάσεων, ή τα αμινοξέα αναπαρίστανται με τη χρήστη μονών γραμμάτων. Μια ακολουθία σε FASTA format ξεκινάει με μια περιγραφή μιας γραμμής, και ακολουθείται από

γραμμές δεδομένων ακολουθίας. Η γραμμή περιγραφής διαχωρίζεται από τα δεδομένα ακολουθίας με ένα σύμβολο ">" στην πρώτη στήλη. Συνίσταται όλες οι γραμμές κειμένου να είναι μικρότερες των 80 χαρακτήρων σε μήκος. Ένα παράδειγμα ακολουθίας σε fasta FORMAT είναι:

```
>gi|186681228|ref|YP_001864424.1|  phycoerythrobilin:ferredoxin oxidoreductase
MNSERSDVTLYQPFLDYAIAYMRSRLDLEPYPIPTGFESNSAVVGKGKNQEEVVTTSYAFQTAKLRQIRA
AHVQGGNSLQVLNFVIFPHLNYDLPFFGADLVTLPGGHLIALDMQPLFRDDSAYQAKYTEPILPIFHAHQ
QHLSWGGDFPEEAQPFFSPAFLWTRPQETAVVETQVFAAFKDYLKAYLDFVEQAEAVTDSQNLVAIKQAQ
LRYLRYRAEKDPARGMFKRFYGAEWTEEYIHGFLFDLERKLTVVK
```

Οι ακολουθίες πρέπει να αναπαρίστανται με χρήση των standard IUB/IUPAC κωδικών για αμινοξέα και νουκλεϊκά οξέα, με τις ακόλουθες εξαιρέσεις:

- Επιτρέπονται τα μικρά γράμματα, αλλά αντιστοιχίζονται στα κεφαλαία
- Μια μονή παύλα μπορεί να χρησιμοποιηθεί για να αναπαραστήσει ένα κενό απροσδιόριστου μεγέθους
- Σε ακολουθίες αμινοξέων το U και * είναι δεκτά γράμματα.
- Οποιαδήποτε αριθμητικά ψηφία στην ακολουθία αναζήτησης πρέπει είτε να αφαιρεθούν, ή να αντικατασταθούν με κατάλληλους κωδικούς γραμμάτων (π.χ. N για άγνωστο υπόλοιπο νουκλεϊκών οξέων, ή X για άγνωστο υπόλοιπο αμινοξέων).

Οι κωδικοί των νουκλεϊκών οξέων είναι:

A --> adenosine	M --> A C (amino)
C --> cytidine	S --> G C (strong)
G --> guanine	W --> A T (weak)
T --> thymidine	B --> G T C
U --> uridine	D --> G A T
R --> G A (purine)	H --> A C T
Y --> T C (pyrimidine)	V --> G C A
K --> G T (keto)	N --> A G C T (any)
	- κενό απροσδιόριστου μήκους

Οι επιτρεπτοί κωδικοί αμινοξέων είναι:

A alanine	P proline
B aspartate or asparagine	Q glutamine
C cystine	R arginine
D aspartate	S serine
E glutamate	T threonine
F phenylalanine	U selenocysteine
G glycine	V valine
H histidine	W tryptophan
I isoleucine	Y tyrosine
K lysine	Z glutamate or glutamine

L leucine	X any
M methionine	* translation stop
N asparagine	- κενό απροσδιόριστου μεγέθους

Ένα παράδειγμα μιας τέτοιας εγγραφής από αρχείο της mirbase είναι το ακόλουθο:

```
>cel-let-7 Caenorhabditis elegans let-7 stem-loop
UACACUGUGGAUCCGGUGAGGUAGUAGGUUGUAUAGUUUGGAAUAUUACCACCGGUGAAC
UAUGCAAUUUUCUACCUUACCGGAGACAGAACUCUUCGA
```

2.2.3 Αρχεία .diff

Τα αρχεία .diff καταγράφουν για κάθε εγγραφή που έχει υποστεί αλλαγή σε σχέση με προηγούμενη έκδοσή της το είδος αυτής της αλλαγής. Η πληροφορία που διατηρείται στα αρχεία .diff αφορά μόνο το είδος της αλλαγής και όχι την αλλαγή καθεαυτή. Για να βρούμε της συγκεκριμένες αλλαγές πρέπει να ανατρέξουμε στα αρχεία .dat των δύο εκδόσεων και να εντοπίσουμε την διαφορά.

Παράδειγμα: Η οντότητα με accession (ID) MI0000685 εμφανίζει στο αρχείο .diff της έκδοσης 5.0 την εγγραφή

```
MI0000685  mmu-mir-10a  SEQUENCE  NAME.
```

Ανατρέχοντας στα αρχεία .dat των εκδόσεων 4.0 και 5.0 μπορούμε να βρούμε ότι η οντότητα με accession MI0000685 στην έκδοση 4.0 έχει όνομα mmu-mir-10a-1, ενώ στην έκδοση 5.0 έχει όνομα mmu-mir-10a. Με αντίστοιχο τρόπο μπορεί να βρεθεί ότι στην έκδοση 4.0 έχουμε μια αλληλουχία βάσεων με στοιχεία

Sequence 110 BP; 29 A; 20 C; 23 G; 0 T; 38 other;

Ενώ στην έκδοση 5.0 η αλληλουχία βάσεων του miRNA έχει στοιχεία:

Sequence 110 BP; 29 A; 21 C; 23 G; 0 T; 37 other;

Προκύπτει από τα παραπάνω ότι η αλλαγή καθεαυτή πρέπει να αναζητηθεί στα αρχεία .dat, ενώ το είδος της αλλαγής βρίσκεται στα αρχεία .diff. Τα είδη αλλαγών που καταγράφονται σε αρχεία .diff στις εκδόσεις της mirbase από την 4.0 μέχρι την τρέχουσα είναι NAME, SEQUENCE, DELETE, NEW και MATURE.

2.2.4 Αρχεία .dead

Τα αρχεία .dead καταγράφουν πληροφορία που αφορά hairpins που έχουν διαγραφεί από την βάση, ή έχουν αντικατασταθεί από κάποιο καινούριο accession. Δεν καταγράφουν πληροφορία για διαγραμμένα matures. Περιέχουν πληροφορίες για την

αιτία διαγραφής, ή προώθησης μιας εγγραφής. Αρχεία .dead υπάρχουν από την έκδοση 5.0 και έπειτα. Η δομή των εγγραφών ενός αρχείου .dead φαίνεται παρακάτω:

```
AC   MI0000260
ID   hsa-mir-124b
FW   MI0000716
CC   miR-124b is not found in either mouse or human genome assemblies so is
CC   removed.
//
```

Η παραπάνω εγγραφή πληροφορεί ότι η οντότητα με accession MI0000260 είχε όνομα hsa-mir-124b και προωθείται στην οντότητα MI0000716. Η γραμμή FW δηλαδή έχει ως δεδομένο το accession στο οποίο θα πρέπει στο εξής να αναφέρεται κανείς στη θέση του accession της γραμμής AC. Οι γραμμές CC παρέχουν πληροφορία για τον λόγω που αυτό συνέβη. Όπως και στην περίπτωση των αρχείων EMBL, η γραμμή που ξεκινάει με τους χαρακτήρες // χρησιμεύει ως διαχωριστικό εγγραφών. Σε περίπτωση που τα πεδία AC και FW περιέχουν την ίδια τιμή, πληροφορούμαστε ότι έχει συντελεστεί μια πλήρης διαγραφή της συγκεκριμένης οντότητας στην οποία αντιστοιχεί το accession της γραμμής AC.

2.3 Διαφοροποιήσεις παρεχόμενων αρχείων της mirbase βάσει της έκδοσης.

Αρχεία .dat

Όπως αναφέρθηκε τα αρχεία .dat καταγράφουν πληροφορία εγγραφών που αντιστοιχούν σε hairpins. Με εξαίρεση τις πρώτες δύο εκδόσεις, όπου απουσιάζουν οι γραμμές σχολίων CC, η πληροφορία που καταγράφεται στα αρχεία .dat αποτελείται από όλους τους τύπους γραμμών που περιγράφθηκαν στην προηγούμενη παράγραφο. Επομένως υπάρχει ομαδοποίηση πληροφορίας με τον ίδιο τρόπο, πράγμα που διευκολύνει και την εξαγωγή της.

Τα πράγματα διαφέρουν όσο αφορά την πληροφορία που δίνεται για mature micro RNAs. Όπως αναφέρθηκε, η πληροφορία αυτή δίνεται στο feature table (γραμμές FT) μιας εγγραφής, το οποίο όμως έχει σχετικά ελεύθερη διάταξη. Τα αρχεία .dat παρουσιάζουν τις ακόλουθες διαφοροποιήσεις με βάση τις εκδόσεις όσο αφορά το feature table των εγγραφών:

Οι εκδόσεις 1.0 ως 4.0 δίνουν στο feature table πληροφορία για βιομόρια που χαρακτηρίζονται ως misc_RNA. Επιπλέον στις ίδιες εκδόσεις παρέχεται μόνο η πληροφορία για το όνομα του παραγόμενου βιομορίου. Δεν έχουμε δηλαδή στοιχεία όπως το accession, που είναι απαραίτητο, διότι χαρακτηρίζει ένα καταγεγραμμένο βιομόριο με μοναδικό τρόπο.

Στις εκδόσεις 5.0 και 5.1 η μορφή ενός FT έχει την ακόλουθη μορφή:

```
/product="hsa-miR-189"                    | Το όνομα του παραγόμενου βιομορίου
/evidence=not_experimental                | Πειραματική, ή όχι εύρεση του βιομορίου
/similarity="MI0000231"                   | Το accession ενός hairpin, που παράγει ένα
                                          |ομόλογο mature και εντοπίζεται σε άλλον
                                          |οργανισμό, από αυτόν που αφορά η εγγραφή.
/experiment="cloned [1-3], Northern [1]"  | Βιβλιογραφική αναφορά στο πείραμα.
```

Πλέον διαθέτουμε πληροφορία για miRNAs, όμως εξακολουθεί να λείπει ο αναγνωριστικός κωδικός accession.

Το πρόβλημα αυτό παύει να υπάρχει από την έκδοση 6.0 και έπειτα. Από την έκδοση αυτή και πέρα καταγράφεται ένα accession για κάθε παραγόμενο mature micro RNA. Από την έκδοση 9.2 και έπειτα υπάρχει επιπλέον η γραμμή

/mod_base= "τιμή".

σε εγγραφές του feature table. Η γραμμή αυτή συνοδεύεται από μια γραμμή όπως η ακόλουθη:

```
modified_base   48
```

Οι γραμμές αυτές δίνουν την πληροφορία ότι η 48^{η} βάση στην ακολουθία του hairpin της εγγραφής πρέπει να τροποποιηθεί και στη θέση της να τοποθετηθεί η βάση που αναγράφεται στην τιμή του πεδίου mod_base. Οι τιμές που μπορεί να πάρει το πεδίο αυτό είναι περιορισμένες σε ένα συγκεκριμένο σύνολο.

Στα παραπάνω, οι γραμμές evidence, similarity, experiment, modified base, mod_base δεν είναι υποχρεωτικές. Τελικά ένα τυπικό feature table από την έκδοση 6.0 και έπειτα έχει τη μορφή:

```
FH   Key          Location/Qualifiers
FH
FT   miRNA         17..38
FT                 /accession="MIMAT0000001"
FT                 /product="cel-let-7"
FT                 /evidence=experimental
```

```
FT                   /experiment="cloned [1-3,5], Northern [1], PCR [4], Solexa
FT                   [6], CLIPseq [7]"
FT   miRNA           56..80
FT                   /accession="MIMAT0015091"
FT                   /product="cel-let-7*"
FT                   /evidence=experimental
FT                   /experiment="CLIPseq [7]"
```

Αρχεία .diff

Τα αρχεία .diff καταγράφονται για πρώτη φορά από την έκδοση 4.0 και έπειτα. Από την έκδοση 4.0 και μέχρι την έκδοση 7.0, η πληροφορία που καταγράφεται στα αρχεία .diff αφορά άμεσα μόνο τις αλλαγές που υφίστανται τα καταγεγραμμένα hairpins. Οι αλλαγές αυτές είναι NEW, NAME, SEQUENCE και DELETE, ενώ σε περιπτώσεις όπου έχουμε κάποιου τύπου αλλαγή στα παραγόμενα mature micro RNAs εμφανίζεται η ένδειξη MATURE. Το είδος της αλλαγής που παρατηρείται στο mature δεν είναι κατά συνέπεια γνωστό.

Από την έκδοση 7.1 και έπειτα τα αρχεία .diff αποτελούνται από δύο διακριτά τμήματα. Το πρώτο αφορά όλες τις αλλαγές που παρατηρούνται για τα hairpins μιας έκδοσης, ενώ το δεύτερο καταγράφει αποκλειστικά τις αλλαγές που υφίστανται τα matures.

2.4 Μοντελοποίηση αλλαγών στην καταγραφή δεδομένων των μορίων micro RNA

Η βάση mirbase διαθέτει σε μορφή αρχείων κειμένου το σύνολο των εγγραφών της που καταγράφονται σε κάθε έκδοση. Το πρόβλημα προς αντιμετώπιση είναι η καταγραφή της εξέλιξης των δεδομένων αυτών από έκδοση σε έκδοση. Για το λόγο αυτό πρέπει αρχικά να προσδιορίσουμε τα διαφορετικά είδη αλλαγών που μπορεί να υφίσταται μια εγγραφή της βάσης. Στη συνέχεια θα πρέπει να αντιστοιχίσουμε σε κάθε τέτοιο είδος αλλαγής από ένα όνομα. Τα δύο αυτά βήματα αποτελούν προϋπόθεση προτού προχωρήσουμε στην καταγραφή των αλλαγών κάθε εγγραφής σε μια δομή, όπως πχ μια βάση δεδομένων.

Κάθε εγγραφή στα αρχεία .dat αποτελεί ένα σύνολο πληροφοριών για ένα hairpin. Το μοναδικό αναγνωριστικό της κάθε εγγραφής είναι ο κωδικός accession. Από τις

υπόλοιπες πληροφορίες που συνοδεύουν μια εγγραφή και δύνανται να μεταβληθούν από έκδοση σε έκδοση ενδιαφέρουν το όνομα, δηλαδή το ID και η ακολουθία νουκλεοτιδίων, δηλαδή το SQ. Επίσης γνωρίζουμε ότι η βάση δεδομένων επεκτείνεται από έκδοση σε έκδοση με την ανακάλυψη νέων hairpins και επομένως ενδιαφέρει να καταγράψουμε την πρώτη εμφάνιση ενός hairpin, ή - ισοδύναμα όσο αφορά τα δεδομένα - την πρώτη εμφάνιση ενός accession. Τέλος, όπως είδαμε υπάρχει η περίπτωση των αρχείων .dead που δίνει πληροφορίες για hairpins που διαγράφηκαν, ή που αντικαταστάθηκαν από άλλα. Εδώ ενδιαφέρει σε ποια έκδοση έγινε η διαγραφή ενός hairpin, δηλαδή του accession που το χαρακτηρίζει, ή αντίστοιχα σε ποια έκδοση έγινε μια προώθηση. Καταλήγουμε έτσι σε έξι δυνατά είδη αλλαγών που αφορούν τα hairpins και τα οποία ονοματίζουμε ως εξής:

- NEW – Υποδηλώνει ότι ένα accession εμφανίζεται για πρώτη φορά στη βάση.
- NAME – Υποδηλώνει ότι το ID ενός hairpin, δηλαδή το ID που συνοδεύει την εγγραφή με δεδομένο accession, μεταβλήθηκε σε σχέση με την προηγούμενη έκδοση.
- SEQUENCE – Υποδηλώνει ότι η ακολουθία νουκλεοτιδίων ενός hairpin, δηλαδή ενός δεδομένου accession μεταβλήθηκε σε σχέση με την προηγούμενη έκδοση. Η αλλαγή μπορεί να αφορά το μήκος της αλληλουχίας, ή την αντικατάσταση κάποιων βάσεων της αλληλουχίας από άλλες, ή και τα δύο.
- DELETE – Υποδηλώνει ότι σταματάει η καταγραφή δεδομένων ενός accession, καθώς αυτό διεγράφη από τη βάση.
- FORWARD –Υποδηλώνει ότι σταματάει η καταγραφή δεδομένων ενός accession, καθώς αυτό διεγράφη από τη βάση, όμως στη θέση του μπορούμε να θεωρήσουμε κάποιο άλλο accession.

Σημειώνουμε ότι το είδος αλλαγής FORWARD στα αρχεία .diff της mirbase καταγράφεται ως είδος DELETE. Σημειώνεται επίσης ότι δύναται μια εγγραφή να υφίσταται ταυτόχρονα την αλλαγή NAME και SEQUENCE, περίπτωση την οποία θα ονομάζουμε NAME SEQUENCE.

Η μοντελοποίηση είναι πιο σύνθετη όταν αναφερόμαστε στα mature micro RNAs. Όπως αναφέρθηκε στο κεφάλαιο 1, κάθε hairpin μπορεί να παράγει ένα, ή περισσότερα matures. Επιπλέον όμοιες ακολουθίες νουκλεοτιδίων που αντιστοιχούν σε δεδομένο mature, δηλαδή σε δεδομένο accession μπορεί να παράγονται από διαφορετικά hairpins. Το αποτέλεσμα είναι ότι για το ίδιο accession που αντιστοιχεί

σε ένα mature μπορούμε να έχουμε πολλές εγγραφές, ανάλογα με το hairpin από το οποίο παράγεται. Έτσι μπορούμε να έχουμε κάποια έκδοση στην οποία για πρώτη φορά εμφανίζεται ένα mature accession συνολικά. Υπάρχει όμως και η περίπτωση στην οποία ένα mature accession καταγράφεται γενικά στη βάση, αλλά εμφανίζεται για πρώτη φορά παραγόμενο από κάποιο άλλο hairpin. Κάτι αντίστοιχο ισχύει και στις περιπτώσεις της διαγραφής. Μπορεί ένα mature accession να διαγραφεί συνολικά και να μην εμφανίζεται πλέον στη βάση. Μπορεί όμως να σταματήσει να καταγράφεται μόνο όσο αφορά ένα σύνολο από hairpins που το παρήγαγαν, ενώ για ένα άλλο σύνολο από hairpins εξακολουθεί να καταγράφεται η παραγωγή του δεδομένου mature. Υπενθυμίζεται ότι δεν υπάρχει η περίπτωση προώθησης για matures, καθώς δεν υπάρχουν αρχεία όπως τα αρχεία .dead που να αφορούν το λόγο της διαγραφής ενός mature. Τέλος εξακολουθούν να υπάρχουν οι περιπτώσεις όπου μεταβάλλεται το ID, ή το SQ, όπως και στα hairpins. Καταλήγουμε έτσι στα ακόλουθα είδη αλλαγών:

- NEW – Υποδηλώνει ότι ένα mature, δηλαδή το accession του, εμφανίζεται για πρώτη φορά στη βάση.
- ADD PARENT HAIRPIN – Υποδηλώνει ότι το mature accession υπάρχει ήδη στη βάση, όμως προσθέτουμε ένα καινούριο "γονιό" hairpin, το οποίο παράγει το mature για πρώτη φορά.
- NAME – Υποδηλώνει ότι το ID ενός mature μεταβλήθηκε σε σχέση με την προηγούμενη έκδοση.
- SEQUENCE – Υποδηλώνει ότι η ακολουθία νουκλεοτιδίων ενός mature μεταβλήθηκε σε σχέση με την προηγούμενη έκδοση. Η αλλαγή μπορεί να αφορά το μήκος της αλληλουχίας, ή την αντικατάσταση κάποιων βάσεων της αλληλουχίας από άλλες, ή και τα δύο.
- DELETE – Υποδηλώνει ότι ένα mature accession αφαιρείται πλήρως από τη βάση.
- REMOVE PARENT HAIRPIN – Υποδηλώνει ότι το mature accession εξακολουθεί να υπάρχει στη βάση, παραγόμενο πλέον όμως μόνο από άλλα hairpins.

Σημειώνεται ότι οι αλλαγές ADD PARENT HAIRPIN, NAME και SEQUENCE, είναι δυνατόν να εμφανίζονται ταυτόχρονα όσο αφορά ένα mature.

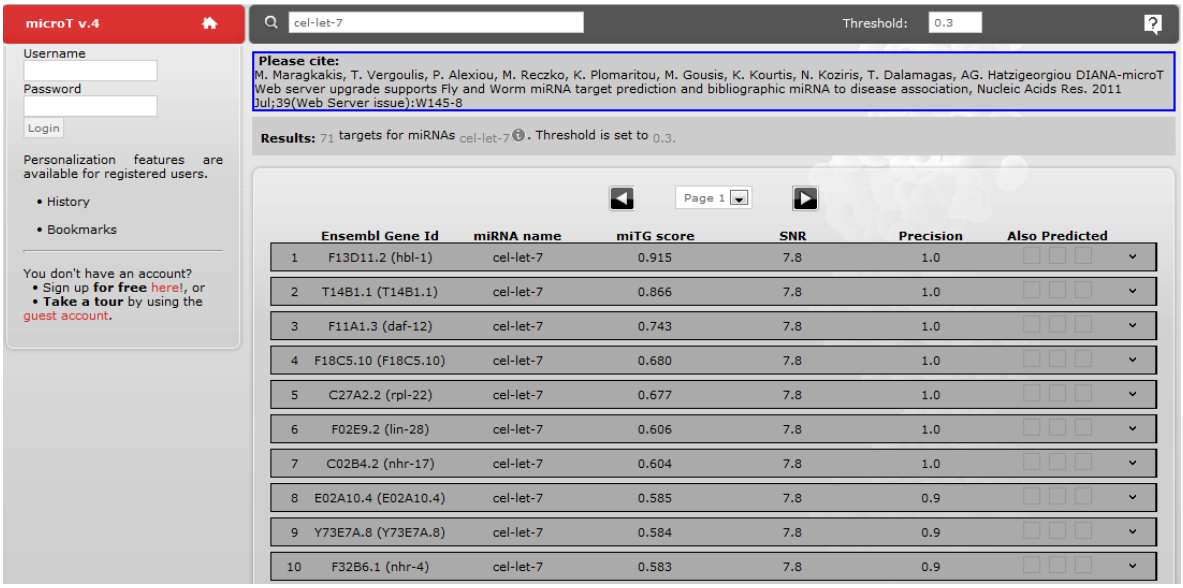
Στα παραπάνω, τα είδη αλλαγής που επιλέξαμε να κωδικοποιήσουμε διαφέρουν από τα αντίστοιχα που καταγράφονται στα αρχεία .diff της mirbase. Αρχικά αποκλείστηκε εντελώς η αλλαγή τύπου MATURE για hairpins, διότι απλά υποδηλώνει κάποια αλλαγή στα παραγόμενα matures, χωρίς να δίνει ουσιαστική πληροφορία για το ποια δεδομένα άλλαξαν. Επιπλέον προσθέσαμε τις αλλαγές ADD PARENT HAIRPIN και REMOVE PARENT HAIRPIN ώστε να παρακολουθήσουμε στη συνέχεια κάθε mature ξεχωριστά με βάση το hairpin από το οποίο παράγεται. Στα αρχεία .diff της mirbase δίνονται μόνο η περιπτώσεις μιας πρώτης συνολικά εμφάνισης (NEW), καθώς και της συνολικής διαγραφής (DELETE).

Έχοντας πλέον κωδικοποιήσει τα είδη των αλλαγών που μπορούν να υφίστανται οι εγγραφές των αρχείων .dat της mirbase θα μπορέσουμε στη συνέχεια συγκρίνοντας τα αρχεία .dat να εξάγουμε πληροφορία για την εξέλιξη της κάθε εγγραφής.

2.5 Το σύστημα DIANA microT v4.0

Η ιστοσελίδα DIANA lab βρίσκεται στην διεύθυνση http://diana.cslab.ece.ntua.gr/. Το DIANA lab εστιάζει στην ανάπτυξη αλγορίθμων, βάσεων δεδομένων και εργαλείων για την καταγραφή και ερμηνεία γονιδιακών δεδομένων στα πλαίσια μιας συστηματικής ανάλυσης. Η έμφαση αυτή τη στιγμή δίνεται στην ανάλυση των micro RNAs και γονιδίων που κωδικοποιούν πρωτεΐνες.

Στη ιστοσελίδα διατίθενται μια σειρά από εφαρμογές που παρέχουν δυνατότητες σε ερευνητές βιολόγους. Μια από αυτές τις εφαρμογές είναι η microT v4.0. Πρόκειται για ένα πρόγραμμα που ασχολείται με την πρόβλεψη στόχων των micro RNAs και βασίζεται σε τεχνητά νευρωνικά δίκτυα. Μπορεί να χρησιμοποιηθεί για να αναζητηθούν γονίδια στόχοι ακολουθιών micro RNA. Η διεύθυνση στην οποία μπορεί κανείς να βρει την εφαρμογή είναι η ακόλουθη: http://diana.cslab.ece.ntua.gr/DianaTools/index.php?r=microtv4/index



Εφαρμογή DIANA microT v4.0 1

Ο χρήστης μπορεί να αναζητήσει γονίδια στόχους δίνοντας ως παράμετρο αναζήτησης το όνομα ενός micro RNA, είτε δίνοντας το accession ενός mature micro RNA. Επίσης μπορεί να δοθεί το όνομα ενός γονιδίου και να βρεθούν τα micro RNAs που το έχουν ως στόχο. Τέλος μπορούν να δοθούν περισσότερες από μια παράμετροι που είναι είτε γονίδια, είτε micro RNAs.

Το πρόβλημα της εξέλιξης των καταγεγραμμένων δεδομένων των micro RNAs εμφανίζεται και στη χρήση της εφαρμογής DIANA microT v4.0. Καθώς απαιτείται η εισαγωγή ενός ονόματος microRNA για αναζήτηση στόχων, υπάρχει περίπτωση να μη μπορούν να βρεθούν αποτελέσματα. Αυτό μπορεί να είναι αποτέλεσμα της αλλαγής στο όνομα ενός micro RNA, σε σχέση με παλιότερες, ή νεότερες εκδόσεις της βάσης.

2.6 Σύντομη παρουσίαση της βάσης ensEMBL

Το σχέδιο της ensEMBL ξεκίνησε το 1999, μερικά χρόνια πριν την ολοκλήρωση της καταγραφής του ανθρώπινου γονιδιώματος. Ο στόχος ήταν ο αυτόματος επεξηγηματικός σχολιασμός του γονιδιώματος, η ενσωμάτωση των επεξηγήσεων σε άλλα διαθέσιμα βιολογικά δεδομένα και η δημόσια διάθεση των παραπάνω μέσω του διαδικτύου. Η ensEMBL περιέχει δεδομένα για το σύνολο γονιδίων διαφόρων οργανισμών σε βάσεις δεδομένων, ενώ διαθέτει και εργαλεία εξόρυξης δεδομένων (data mining), όπως το BioMart. Τέλος διαθέτει πληροφορίες για την σύγκριση,

ποικιλομορφία και τη ρύθμιση της έκφρασης γονιδίων. Όλα τα παραπάνω δεδομένα μπορούν να παρουσιαστούν στο χρήστη της ιστοσελίδας.

Μπορεί κανείς να κατεβάσει τις βάσεις δεδομένων της ensEMBL από το ftp site της, σε διάφορες μορφές. Τα αρχεία αυτά μπορεί να έχουν μέγεθος της τάξης των GB. Κάθε κατάλογος στο ftp.ensembl.org περιέχει ένα readme αρχείο που περιγράφει τη δομή του καταλόγου. Τα αρχεία που καταγράφουν γονιδιακά δεδομένα για τους διάφορους οργανισμούς μπορεί να έχουν μια από τις ακόλουθες δομές: FASTA, EMBL, GenBank, GTF, MySQL, EMF, GVF, VEP, GFF. Τα τέσσερα τελευταία είδη δεν αφορούν καθαρά τα δεδομένα γονιδίων, αλλά δεδομένα ποικιλότητας γονιδίων, ή ρύθμισης της έκφρασης γονιδίων. Τα αρχεία με δεδομένα σε EMBL και GenBank, αποτελούν database dumps ακολουθιών γονιδίων.

Η βάση ensEMBL ανανεώνεται με νέα δεδομένα κάθε μήνα. Οι σελίδες αρχείου της ensEMBL δημιουργήθηκαν ώστε να υπάρχει σταθερός σύνδεσμος, για τρία χρόνια, προς σελίδες μιας συγκεκριμένης έκδοσης. Οι σύνδεσμοι αυτοί είναι κατάλληλοι ώστε να γίνεται σωστά αναφορά στην ensEMBL σε δημοσιεύσεις. Μερικά ακόμα παλιότερα αρχεία μπορεί να διατηρούνται, αν συμπεριλαμβάνουν σημαντικά δεδομένα για βασικά είδη οργανισμών, όμως η διαθεσιμότητα των σελίδων αυτών δεν είναι εγγυημένη. Στο [18] μπορεί κανείς να βρει ποια αρχεία υπάρχουν διαθέσιμα ανά πάσα στιγμή. Τη στιγμή που γράφεται το κείμενο οι εκδόσεις που υπάρχουν αναρτημένες είναι η έκδοση 46 και οι εκδόσεις 50 έως 65, όπου η έκδοση 65 είναι η τρέχουσα.

Δεν καταγράφονται δεδομένα για τα ίδια είδη σε όλες τις εκδόσεις των βάσεων της ensEMBL. Για κάθε σελίδα αρχείου, επεκτείνοντας στον browser τη διεύθυνση ως “info/about/species.html”, όπως για παράδειγμα στο [19], μπορούμε να δούμε έναν κατάλογο με τα είδη των οποίων γονίδια καταγράφονται στη δεδομένη έκδοση.

Ενδιαφέρον παρουσιάζει μια υποσημείωση στην ίδια σελίδα αρχείων της ensEMBL, η οποία αναφέρει τα ακόλουθα:

“Ensembl aims to maintain stable identifiers for genes (ENSG), transcripts (ENST), proteins (ENSP) and exons (ENSE) as long as possible. Changes within the genome sequence assembly or an updated genome annotation may dramatically change a gene model. In these cases, the old set of stable IDs is retired and a new one assigned. Gene and transcript pages both have an ID History view which maps changes in the ID from the earliest version in Ensembl.”

Με άλλα λόγια, βάσει της σημείωσης υπάρχουν περιπτώσεις αλλαγών στην ακολουθία ενός γονιδίου, οι οποίες συνεπάγονται και αλλαγή στο όνομα. Επομένως το ζήτημα της καταγραφής της εξέλιξης των δεδομένων από έκδοση σε έκδοση είναι ανοιχτό και για γονίδια που καταγράφονται στην βάση ensEMBL.

Τέλος, βλέποντας τη σελίδα ftp.ensembl.org, παρατηρούμε ότι υπάρχουν διαθέσιμα δεδομένα από διάφορους οργανισμούς, για τις εκδόσεις 19 και 21-65. Δεν υπάρχουν δεδομένα για το ίδιο σύνολο οργανισμών σε κάθε έκδοση. Τα είδη των αρχείων που είναι διαθέσιμα σε κάθε έκδοση και για κάθε οργανισμό, επίσης διαφέρουν. Επιπλέον, δε δίνονται συγκεντρωτικά όλες οι πληροφορίες μιας έκδοσης. Αυτές είναι ταξινομημένες ανά οργανισμό, ενώ για κάθε οργανισμό οι καταγεγραμμένες πληροφορίες είναι κατανεμημένες σε αρχεία που περιέχουν ένα συγκεκριμένο αριθμό γονιδίων.

3

Σχεδίαση συστήματος I: Χτίζοντας το backbone της εφαρμογής

3.1 Μοντελοποίηση της εξέλιξης δεδομένων με σχεσιακή βάση mySQL

Σκοπός της εργασίας είναι η ανάπτυξη μιας εφαρμογής ιστού που να δίνει την δυνατότητα προβολής της ιστορικής εξέλιξης των δεδομένων που καταγράφονται για κάθε βιομόριο της βάσης mirbase. Η πληροφορία για την εξέλιξη αυτή βρίσκεται έμμεσα στα αρχεία .dat και τα αρχεία .diff, όπως περιγράφθηκε στο κεφάλαιο 2. Έμμεσα, διότι αφενός οι αλλαγές που υφίστανται οι εγγραφές στα αρχεία .dat μπορούν να παρατηρηθούν με σύγκριση των αρχείων .dat δύο διαδοχικών εκδόσεων. Η διαδικασία πρέπει να γίνει για όλες τις διαδοχικές εκδόσεις και για όλες τις περιεχόμενες σε κάθε έκδοση εγγραφές. Αφετέρου στα αρχεία .diff καταγράφεται για κάθε έκδοση μόνο το παρατηρούμενο είδος της αλλαγής των δεδομένων μιας εγγραφής σε σχέση με την προηγούμενη έκδοση και όχι η αλλαγή καθεαυτή. Προκύπτει έτσι η ανάγκη να δημιουργηθεί μια συγκεντρωτική δομή που εμπεριέχει

πληροφορίες για τις αλλαγές όλων των εγγραφών σε όλες τις εκδόσεις. Αυτή θα αποτελεί και τη βάση της εφαρμογής. Ως τέτοια δομή αναπτύχθηκε μια βάση δεδομένων mySQL. Στο παρόν κεφάλαιο περιγράφονται οι πίνακες που την αποτελούν, η διαδικασία ανάπτυξης και υλοποίησής της, καθώς και σενάρια χρήσης της από έναν διαχειριστή.

3.1.1 Ο πίνακας hairpinhistory

Η βάση δεδομένων που αναπτύχθηκε αποτελείται από έναν πίνακα για κάθε τύπο βιομορίων που αποθηκεύει. Ο πρώτος από τους δύο πίνακες που προκύπτουν ονομάζεται hairpinhistory και συγκεντρώνει την ιστορία των εγγραφών μορίων τύπου hairpin. Μια εγγραφή σε αρχεία .dat έχει την ακόλουθη μορφή:

```
ID   cel-let-7       standard; RNA; CEL; 99 BP.
XX
AC   MI0000001;
XX
DE   Caenorhabditis elegans let-7 stem-loop
XX
RN   [1]
RX   PUBMED; 11679671.
RA   Lau NC, Lim LP, Weinstein EG, Bartel DP;
RT   "An abundant class of tiny RNAs with probable regulatory roles in
RT   Caenorhabditis elegans";
RL   Science. 294:858-862(2001).
XX
RN   [2]
RX   PUBMED; 12672692.
RA   Lim LP, Lau NC, Weinstein EG, Abdelhakim A, Yekta S, Rhoades MW, Burge CB,
RA   Bartel DP;
RT   "The microRNAs of Caenorhabditis elegans";
RL   Genes Dev. 17:991-1008(2003).
XX
RN   [3]
RX   PUBMED; 12747828.
RA   Ambros V, Lee RC, Lavanway A, Williams PT, Jewell D;
RT   "MicroRNAs and other tiny endogenous RNAs in C. elegans";
RL   Curr Biol. 13:807-818(2003).
XX
RN   [4]
RX   PUBMED; 12769849.
RA   Grad Y, Aach J, Hayes GD, Reinhart BJ, Church GM, Ruvkun G, Kim J;
RT   "Computational and experimental identification of C. elegans microRNAs";
RL   Mol Cell. 11:1253-1263(2003).
XX
```

```
RN   [5]
RX   PUBMED; 17174894.
RA   Ruby JG, Jan C, Player C, Axtell MJ, Lee W, Nusbaum C, Ge H, Bartel DP;
RT   "Large-scale sequencing reveals 21U-RNAs and additional microRNAs and
RT   endogenous siRNAs in C. elegans";
RL   Cell. 127:1193-1207(2006).
XX
RN   [6]
RX   PUBMED; 19460142.
RA   Kato M, de Lencastre A, Pincus Z, Slack FJ;
RT   "Dynamic expression of small non-coding RNAs, including novel microRNAs
RT   and piRNAs/21U-RNAs, during Caenorhabditis elegans development";
RL   Genome Biol. 10:R54(2009).
XX
RN   [7]
RX   PUBMED; 20062054.
RA   Zisoulis DG, Lovci MT, Wilbert ML, Hutt KR, Liang TY, Pasquinelli AE, Yeo
RA   GW;
RT   "Comprehensive discovery of endogenous Argonaute binding sites in
RT   Caenorhabditis elegans";
RL   Nat Struct Mol Biol. 17:173-179(2010).
XX
DR   RFAM; RF00027; let-7.
DR   WORMBASE; C05G5/12462-12364; .
XX
CC   let-7 is found on chromosome X in Caenorhabditis elegans [1] and pairs to
CC   sites within the 3' untranslated region (UTR) of target mRNAs, specifying
CC   the translational repression of these mRNAs and triggering the transition
CC   to late-larval and adult stages [2].
XX
FH   Key            Location/Qualifiers
FH
FT   miRNA           17..38
FT                   /accession="MIMAT0000001"
FT                   /product="cel-let-7"
FT                   /evidence=experimental
FT                   /experiment="cloned [1-3,5], Northern [1], PCR [4], Solexa
FT                   [6], CLIPseq [7]"
FT   miRNA           56..80
FT                   /accession="MIMAT0015091"
FT                   /product="cel-let-7*"
FT                   /evidence=experimental
FT                   /experiment="CLIPseq [7]"
XX
SQ   Sequence 99 BP; 26 A; 19 C; 24 G; 0 T; 30 other;
     uacacugugg auccggugag guaguagguu guauaguuug gaauauuacc accggugaac        60
     uaugcaauuu ucuaccuuac cggagacaga acucuucga                               99
//
```

Στα παραπάνω όσο αφορά την πληροφορία που ενδιαφέρει για τα hairpins μπορούμε να αγνοήσουμε όλες τις γραμμές FH και FT. Επίσης ενδιαφερόμαστε να καταγράψουμε κυρίως τα στοιχεία ID και SQ, που είναι και τα στοιχεία που μεταβάλλονται με την εμφάνιση νέων εκδόσεων. Φυσικά δε μπορεί να παραληφθεί το στοιχείο AC, το οποίο χαρακτηρίζει μοναδικά μια εγγραφή. Επιπλέον στοιχεία που ενδιαφέρουν είναι η έκδοση για την οποία αποτυπώνουμε το ID και το SQ, καθώς και το πιθανό είδος αλλαγής που συνέβη.

Επειδή μια εγγραφή μπορεί για μια σειρά εκδόσεων να μην παρουσιάζει καμία αλλαγή, η καταγραφή των στοιχείων ID και SQ για κάθε έκδοση θα δημιουργούσε ένα πλήθος εγγραφών με επαναλαμβανόμενη πληροφορία. Αντίθετα μπορούμε να πάρουμε ποιοτική πληροφορία για το διάστημα στο οποίο μια εγγραφή έχει αμετάβλητα ID και SQ διατηρώντας από ένα πεδίο για την πρώτη και την τελευταία έκδοση στην οποία εμφανίζεται με τα δεδομένα στοιχεία. Ένα πρόβλημα που προκύπτει από αυτή τη μοντελοποίηση είναι ότι υπάρχουν 33 εκδόσεις αρχείων .dat, οι οποίες έχουν ονόματα 1.0 έως 18. Οι εκδόσεις που εμφανίζονται στα ενδιάμεσα των ακέραιων αριθμών εκδόσεως (π.χ. 1.2) δεν είναι σταθερές, ή προδιαγεγραμμένες. Έτσι δεν μπορεί κανείς να γνωρίζει αν μετά την έκδοση 2.1 θα έπρεπε να ακολουθεί η 2.2, ή η έκδοση 3.0. Για το λόγο αυτό χωρίζουμε τα πεδία που οριοθετούν το διάστημα όπου μια εγγραφή εμφανίζεται με σταθερά ID και SQ σε πεδία που αναφέρονται στον απόλυτο αύξοντα αριθμό έκδοσης και σε πεδία που διατηρούν την τιμή έκδοσης όπως εμφανίζεται στον τίτλο των αρχείων καταλόγου της mirbase.

Για λόγους παρακολούθησης της αλλαγής που συνέβη μπορούμε να διαθέσουμε επιπλέον πεδία που περιέχουν τις προηγούμενες τιμές ID και SQ. Επιπλέον σε περίπτωση διαγραφής ενός hairpin υπάρχει πληροφορία με το accession στο οποίο προωθείται, εφόσον υπάρχει προώθηση, και πληροφορία που αναφέρεται στους λόγους της διαγραφής. Την πληροφορία αυτή μπορούμε να την αντλήσουμε από τα αρχεία .dead και να την προσθέσουμε στις εγγραφές της βάσης.

Τέλος, για λόγους χρησιμότητας του πίνακα της βάσης και σε εφαρμογές πέραν της παρούσας, μπορούμε να διατηρούμε από ένα πεδίο που περιέχει την έκδοση πρώτης εμφάνισης, την έκδοση διαγραφής, ή την έκδοση προώθησης ενός hairpin.

Καταλήγουμε έτσι στο σχήμα του πίνακα hairpinhistory που περιέχει τα ακόλουθα πεδία:

- DBid - Είναι ένας ακέραιος αύξων αριθμός, μοναδικός για κάθε εγγραφή του πίνακα. Επιβάλλεται ένα τέτοιο πεδίο, για να διαχωρίζονται όλες οι εγγραφές του πίνακα μεταξύ τους και να έχουν ένα μοναδικό αναγνωριστικό.
- AC - Είναι το χαρακτηριστικό accession ενός hairpin (π.χ. MI0000001), τύπου VARCHAR.
- ID - Είναι το όνομα που αντιστοιχεί στο παραπάνω accession (π.χ. cel-let-7) για το συγκεκριμένο διάστημα εκδόσεων της εγγραφής. Ο τύπος δεδομένων είναι VARCHAR.
- first_appearance - Είναι ένας ακέραιος που αντιστοιχεί στην έκδοση όπου για πρώτη φορά εμφανίζεται το συγκεκριμένο accession της εγγραφής με δεδομένες τιμές στα πεδία ID και SQ.
- last_appearance - Είναι ένας ακέραιος που αντιστοιχεί στην τελευταία έκδοση που εμφανίζεται το accession με τις δεδομένες τιμές στα πεδία ID και SQ.
- actual_first_appearance - Είναι η πραγματική τιμή της πρώτης έκδοσης, όπως δίνεται από την mirbase, όπου η εγγραφή έχει τα δεδομένα στοιχεία στα πεδία ID και SQ (π.χ. η 10η έκδοση της mirbase αντιστοιχεί στην 3.0). Ο τύπος δεδομένων που χρησιμοποιείται είναι VARCHAR.
- actual_last_appearance - Είναι η πραγματική τιμή της έκδοσης, όπου για τελευταία φορά έχουμε δεδομένα στοιχεία στα πεδία ID και SQ. Ο τύπος δεδομένων που χρησιμοποιείται είναι VARCHAR.
- DE - Είναι ένα μικρό κείμενο περιγραφής που συνοδεύει κάθε εγγραφή hairpin στα αρχεία .dat. Είναι τύπου TEXT.
- SQ - Το πεδίο αυτό περιέχει πληροφορία σχετική με την ακολουθία του hairpin. Ο τύπος δεδομένων του είναι BLOB (binary large object). Ο τύπος αυτός επιλέχθηκε για λόγους συνοχής με πιθανές μελλοντικές επεκτάσεις της βάσης όπου θα συμπεριλαμβάνονται γονίδια μήκους πολλών χιλιάδων βάσεων.
- Change - Το πεδίο αυτό αναγράφει με κείμενο ποιό είδος αλλαγής υπέστη ένα hairpin στην έκδοση που αναγράφεται στο πεδίο first_appearance/ actual_first_appearance. Τα είδη των αλλαγών για hairpins έχουν κωδικοποιηθεί στο κεφάλαιο 2. Το πεδίο αυτό είναι τύπου TEXT.

- prev_name - Το όνομα (ID) που είχε το hairpin πριν από πιθανή αλλαγή NAME ή NAME SEQUENCE. Η τιμή του είναι null όταν δεν υπάρχει αλλαγή NAME. Είναι τύπου VARCHAR.
- prev_sq - Η ακολουθία που είχε το hairpin πριν από αλλαγή SEQUENCE ή NAME SEQUENCE. Η τιμή του είναι null όταν δεν υπάρχει αλλαγή SEQUENCE. Το πεδίο αυτό είναι τύπου BLOB.
- forward - Το πεδίο αυτό συμπληρώνεται μόνο για εγγραφές που έχουν στο πεδίο Change την τιμή FORWARD. Στο πεδίο forward υπάρχει η τιμή ενός accession, και δηλώνει σε ποιό hairpin (δηλαδή σε ποιο hairpin accession) γίνεται προώθηση μετά τη διαγραφή. Διαφορετικά η τιμή του πεδίου, είναι NULL. Ο τύπος δεδομένων του πεδίου είναι VARCHAR.
- comment - Το πεδίο αυτό επίσης συμπληρώνεται μόνο για εγγραφές που έχουν στο πεδίο Change την τιμή DELETE ή FORWARD. Είναι μια εξήγηση σε μορφή κειμένου του λόγου για τον οποίο έγινε η διαγραφή του hairpin. Ο τύπος δεδομένων του πεδίου είναι TEXT.
- new_appearance - Είναι ένας ακέραιος που αντιστοιχεί στην πρώτη έκδοση όπου εμφανίζεται το δεδομένο accession, ανεξάρτητα από τιμές ID και SQ, δηλαδή αντιστοιχεί στην πρώτη του εμφάνιση.
- forward_appearance - Είναι ένας ακέραιος που αντιστοιχεί στην έκδοση όπου γίνεται forward το δεδομένο accession. Αν ένα accession δεν γίνεται forward, ή δεν έχει καν διαγραφεί από τη βάση, η τιμή αυτή είναι NULL.
- delete_appearance - είναι ένας ακέραιος που αντιστοιχεί στην έκδοση όπου γίνεται delete το δεδομένο accession. Αν ένα accession δεν έχει ακόμα διαγραφεί, ή αν η διαγραφή του συνοδεύεται από forward, η τιμή του πεδίου είναι NULL.

Παραδείγματα εγγραφών και η ερμηνεία τους δίνονται στην ακόλουθη σελίδα.

DBid	AC	ID	first_ appearance	last_ appearance	actual_ first_ appearance	actual_ last_ appearance	DE	SQ	Change
1	MI0000001	cel-let-7	7	33	2.0	18.0	Caenorhabditis elegans let-7 stem-loop	[BLOB - 200B]	NAME

prev_name	prev_sq	forward	comment	new_ appearance	forward_appearance	delete_ appearance
cel-let-7L	[BLOB - 0B]	*NULL*	*NULL*	1	*NULL*	*NULL*

Η παραπάνω εγγραφή μας πληροφορεί ότι στο διάστημα των εκδόσεων 2.0 μέχρι και 18.0, το hairpin με accession MI0000001 έχει όνομα cel-let-7 και η ακολουθία των νουκλεοτιδίων του δίνεται από το πεδίο SQ. Πληροφορούμαστε επίσης ότι το accession αυτό στην έκδοση 2.0 εμφάνισε αλλαγή στο ID του, δηλαδή το όνομά του. Το αμέσως προηγούμενο όνομα πριν την έκδοση 2.0 δίνεται στο πεδίο prev_name και είναι cel-let-7L. Ακόμα πληροφορούμαστε ότι η πρώτη εμφάνιση σε αρχεία .dat, του accession MI0000001 είναι στην έκδοση με αύξων αριθμό 1. Τέλος επειδή τόσο το πεδίο forward_appearance, όσο και το πεδίο delete_appearance έχουν την τιμή null, γνωρίζουμε ότι το accession αυτό εξακολουθεί να καταγράφεται στη βάση και δεν έχει διαγραφεί.

Παρουσιάζουμε και την αμέσως προηγούμενη μορφή του hairpin αυτού, όπως καταγράφεται στον πίνακα hairpins της βάσης στην επόμενη σελίδα:

DBid	AC	ID	first_ appearance	last_ appearance	actual_ first_ appearance	actual_ last_ appearance	DE	SQ	Change
2	MI0000001	cel-let-7L	1	6	1.0	1.5	Caenorhabditis elegans let-7 precursor RNA	[BLOB - 200B]	NEW

prev_name	prev_sq	forward	comment	new_ appearance	forward_appearance	delete_ appearance
NULL	[BLOB - 0B]	*NULL*	*NULL*	1	*NULL*	*NULL*

Η παραπάνω εγγραφή μας πληροφορεί ότι το accession MI0000001 έχει όνομα cel-let-7L στο διάστημα εκδόσεων 1.0 έως 1.5. Η ένδειξη NEW στο πεδίο Change μας πληροφορεί ότι στην έκδοση 1.0 έχουμε και την συνολικά πρώτη εμφάνιση του hairpin, η οποία παραμένει αμετάβλητη μέχρι και την έκδοση 1.5. Η ακολουθία νουκλεοτιδίων δίνεται όπως και πριν στο πεδίο SQ. Τα πεδία prev_name, prev_sq είναι NULL, γιατί έχουμε την πρώτη εμφάνιση και δεν έχουν υπάρξει προηγούμενες τιμές ID και SQ. Επίσης τα πεδία forward, comment, forward_appearance, delete_appearance είναι NULL, διότι αφενός η εγγραφή δεν αναφέρεται σε έκδοση όπου έχουμε διαγραφή ενός accession και αφετέρου το accession αυτό καταγράφεται ακόμα στην τρέχουσα έκδοση της βάσης δεδομένων.

Παρακάτω δίνεται και η μορφή μιας εγγραφής η οποία αναφέρεται σε διαγραφή ενός accession από τη βάση:

DBid	AC	ID	first_ appearance	last_ appearance	actual_first_ appearance	actual_last_ appearance	DE	SQ	Change
18374	MI0002102	*NULL*	20	*NULL*	8.2	*NULL*	*NULL*	[BLOB - 0B]	FORWARD

prev_name	prev_sq	forward	comment	new_ appearance	forward_ appearance	delete_ appearance
dre-mir-430c-22	[BLOB - 200B]	MI0002079	28 identical zebrafish mir-430b precursor sequenc...	16	20	*NULL*

Τι πληροφορία δίνει η παραπάνω εγγραφή; Το hairpin με accession MI0002102 διαγράφεται από τη βάση στην έκδοση με αύξων αριθμό 20, που αντιστοιχεί στην πραγματική έκδοση 8.2. Η διαγραφή που γίνεται είναι τύπου FORWARD, δηλαδή αναφερόμαστε πλέον στη θέση αυτού του hairpin σε αυτό που έχει accession MI0002079, δηλαδή του accession που δίνεται από το πεδίο forward. Το όνομα και η ακολουθία του hairpin που είχαμε πριν τη διαγραφή δίνονται στα πεδία prev_name και prev_sq. Ο λόγος για τον οποίο αφαιρείται το hairpin από τη βάση δίνεται στο πεδίο comment. Τέλος μαθαίνουμε ότι η πρώτη εμφάνιση του hairpin με accession MI0002102 είναι στην έκδοση με αύξων αριθμό 16.

3.1.2 Ο πίνακας maturehistory

Ο δεύτερος από τους δύο πίνακες που προκύπτουν ονομάζεται maturehistory και συγκεντρώνει την ιστορία των εγγραφών μορίων τύπου mature. Τα δεδομένα που αφορούν τα mature micro RNAs δίνονται στα αρχεία .dat μέσω των γραμμών FT, δηλαδή των γραμμών feature table κάθε εγγραφής ενός hairpin. Έχουμε δηλαδή δοσμένα τα δεδομένα ενός mature micro RNA σε σύζευξη με το αντίστοιχο hairpin που το παράγει. Το feature table της εγγραφής που παρουσιάσαμε παραπάνω έχει την μορφή

```
FT   miRNA          17..38
FT                   /accession="MIMAT0000001"
FT                   /product="cel-let-7"
FT                   /evidence=experimental
FT                   /experiment="cloned [1-3,5], Northern [1], PCR [4], Solexa
FT                   [6], CLIPseq [7]"
FT   miRNA          56..80
FT                   /accession="MIMAT0015091"
FT                   /product="cel-let-7*"
FT                   /evidence=experimental
FT                   /experiment="CLIPseq [7]"
```

Παρατηρούμε ότι για κάθε miRNA που παράγεται από την δοσμένη εγγραφή έχουμε μια γραμμή FT που ξεκινάει με τη λέξη "miRNA". Στη συνέχεια δίνεται η αρχή και το τέλος του mature σε σχέση με τα νουκλεοτίδια που αποτελούν το hairpin που το παράγει. Οι σειρές που ακολουθούν περιέχουν το accession, το ID και ορισμένα δεδομένα που αφορούν τον τρόπο, ή τη δημοσίευση που αποδεικνύει την ύπαρξη του mature. Οι πληροφορίες που χρειαζόμαστε αφορούν την ακολουθία και το όνομα ενός mature, επομένως μας είναι απαραίτητο το accession, το ID καθώς και τα όρια του mature πάνω στο hairpin, ώστε να αντλήσουμε σωστά την ακολουθία που το αποτελεί από το τμήμα SQ της εγγραφής του hairpin. Επιπλέον θα χρησιμοποιήσουμε ως συνοδευτική πληροφορία το hairpin accession της εγγραφής που περιέχει το mature, ώστε να παρακολουθήσουμε ξεχωριστά την εξέλιξη ίδιων mature που παράγονται από διαφορετικά hairpins. Τέλος, υπενθυμίζεται ότι για mature micro RNAs η πληροφορία accession δίνεται από την έκδοση 6.0 και μετά των αρχείων .dat και επομένως δεν μπορούμε να καταγράψουμε την ιστορία τους νωρίτερα.

Όσες άλλες πληροφορίες είναι διαθέσιμες τις χειριζόμαστε όπως και στην περίπτωση των hairpins. Το σχήμα του πίνακα στο οποίο καταλήγουμε αποτελείται από τα ακόλουθα πεδία:

- mID - Ένας μοναδικός για κάθε εγγραφή του πίνακα αύξων αριθμός. Επιβάλλεται ώστε κάθε εγγραφή της βάσης να έχει μια ταυτότητα.
- AC - Το αναγνωριστικό accession ενός miRNA (π.χ. MIMAT0000001). Είναι τύπου VARCHAR.
- ID - Το όνομα του παραπάνω accession (π.χ. cel-let-7). Είναι τύπου VARCHAR.
- mother_hairpin – Το accession του hairpin που παράγει το συγκεκριμένο miRNA. Είναι τύπο VARCHAR.
- sequence_part - Το πεδίο αυτό υποδηλώνει ποιό τμήμα του hairpin αποτελεί το mature. Δίνονται δύο αριθμοί που αποτελούν την αρχή και το τέλος του mature στην ακολουθία του hairpin (π.χ. 17..38). Είναι τύπου VARCHAR.
- SQ - Είναι η ακολουθία του miRNA. Είναι τύπου BLOB. Αυτό έγινε όπως αναφέρθηκε για λόγους συνοχής της βάσης με πιθανές μελλοντικές επεκτάσεις της.
- first_appearance - Είναι ένας ακέραιος που αντιστοιχεί στην έκδοση όπου για πρώτη φορά εμφανίζεται το συγκεκριμένο accession της εγγραφής με δεδομένες τιμές στα πεδία ID και SQ. Στην περίπτωση των matures η πρώτη φορά όπου υπάρχει πληροφορία για accessions στα αρχεία .dat της mirbase είναι η έκδοση 15/6.0 και έτσι αυτή είναι η ελάχιστη τιμή που μπορεί να πάρει το πεδίο.
- last_appearance - Είναι ένας ακέραιος που αντιστοιχεί στην τελευταία φορά που εμφανίζεται το accession με τις δεδομένες τιμές στα πεδία ID και SQ.
- actual_first_appearance - Είναι η πραγματική τιμή της πρώτης έκδοσης, όπως δίνεται από την mirbase, όπου η εγγραφή έχει τα δεδομένα στοιχεία στα πεδία ID και SQ (π.χ. η 10η έκδοση της mirbase αντιστοιχεί στην 3.0). Το πεδίο είναι τύπου VARCHAR.
- actual_last_appearance - Είναι η πραγματική τιμή της έκδοσης, όπου για τελευταία φορά έχουμε δεδομένα στοιχεία στα πεδία ID και SQ. Το πεδίο είναι τύπου VARCHAR.

- Change - Είναι το πεδίο που με μορφή κειμένου αναγράφει τι αλλαγή υπέστη μια εγγραφή. Τα είδη των αλλαγών για matures έχουν περιγραφθεί στο κεφάλαιο 2.
- prev_name - Περιέχει την τιμή του ονόματος που είχαμε πριν συμβεί αλλαγή NAME, ή NAME SEQUENCE (σε διαφορετική περίπτωση είναι NULL).
- prev_sq - Περιέχει την ακολουθία που είχαμε πριν συμβεί αλλαγή SEQUENCE, ή NAME SEQUENCE (σε διαφορετική περίπτωση είναι NULL).
- new_appearance - Είναι ένας ακέραιος που αντιστοιχεί στην πρώτη έκδοση εμφάνισης του δεδομένου accession.
- delete_appearance - Είναι ένας ακέραιος που αντιστοιχεί στην έκδοση όπου διαγράφτηκε το δεδομένο accession. Σε διαφορετική περίπτωση είναι NULL.

Σημειώνουμε ότι όταν έχουμε την περίπτωση μιας εγγραφής που ταυτόχρονα παρουσιάζει την αλλαγή ADD PARENT HAIRPIN και NAME, SEQUENCE, ή τις δύο τελευταίες ταυτόχρονα, επιλέχθηκε να εισάγονται δύο εγγραφές στη βάση. Μια με ένδειξη ADD PARENT HAIRPIN και μια δεύτερη που προσδιορίζει το είδος αλλαγής του accession σε σχέση με προηγούμενες εκδόσεις. Επιπλέον επιλέξαμε να έχουμε μόνο μια εγγραφή που να δηλώνει την πρώτη συνολικά εμφάνιση ενός accession, με την ένδειξη NEW. Επιπλέον για κάθε πρώτη εμφάνιση του ίδιου accession παραγόμενου από κάποιο άλλο hairpin έχουμε μια εγγραφή με την ένδειξη ADD PARENT HAIRPIN. Την ίδια στρατηγική ακολουθούμε και στην περίπτωση των διαγραφών όπου έχουμε μια εγγραφή με ένδειξη DELETE αν διαγραφεί συνολικά ένα accession, ενώ για κάθε επιμέρους διαγραφή βάσει hairpin έχουμε μια εγγραφή με την ένδειξη REMOVE PARENT HAIRPIN.

Παραδείγματα εγγραφών του πίνακα maturehistory που αφορούν το μόριο με accession MIMAT0000001 δίνονται στην σελίδα που ακολουθεί.

mID	AC	ID	mother_ hairpin	sequence _ part	SQ	first_ appearance	last_ appearance	actual_ first_ appearance	actual_ last_ appearance
23520	MIMAT0000001	cel-let-7	*NULL*		[BLOB - 22B]	15	*NULL*	6.0	*NULL*

Change	prev_name	prev_sq	new_appearance	delete_appearance
NEW	*NULL*	[BLOB - 0B]	15	*NULL*

Η παραπάνω εγγραφή μας πληροφορεί ότι για πρώτη φορά το mature με accession MIMAT0000001 καταγράφεται στη βάση στην έκδοση 6.0.

Επίσης μας πληροφορεί ότι το mature αυτό υπάρχει ακόμα στη βάση και δεν έχει διαγραφεί.

Περισσότερες πληροφορίες παίρνουμε από τις ακόλουθες εγγραφές:

mID	AC	ID	mother_ hairpin	sequence_ part	SQ	first_ appearance	last_ appearance	actual_ first_ appearance	actual_ last_ appearance
1830	MIMAT0000001	cel-let-7	MI0000001	17..38	[BLOB - 22B]	15	32	6.0	17.0

Change	prev_name	prev_sq	new_appearance	delete_appearance
ADD PARENT HAIRPIN	*NULL*	[BLOB - 0B]	15	*NULL*

mID	AC	ID	mother_ hairpin	sequence_ part	SQ	first_ appearance	last_ appearance	actual_ first_ appearance	actual_ last_ appearance
48468	MIMAT0000001	cel-let-7-5p	MI0000001	17..38	[BLOB - 22B]	33	33	18.0	18.0

Change	prev_name	prev_sq	new_appearance	delete_appearance
NAME	cel-let-7	[BLOB - 0B]	15	*NULL*

Τι μας πληροφορούν οι παραπάνω εγγραφές; Αρχικά το MIMAT0000001 παράγεται από το hairpin με accession MI0000001. Το όνομά του είναι cel-let-7 στις εκδόσεις 6.0 ως 17.0. Αποτελείται από το 17^{ο} ως το 38^{ο} νουκλεοτίδιο του hairpin που το παράγει. Τέλος έχουμε αλλαγή στο όνομα, η οποία παρατηρείται στην έκδοση 18.0.

3.1.3 Υλοποίηση script για το χτίσιμο της βάσης

Αφού ορίστηκαν τα πεδία από τα οποία θα πρέπει να αποτελείται η βάση δεδομένων και αφού περιγράφθηκε η δομή μιας τυπικής εγγραφής, μπορεί πλέον να συγγραφεί ένα script που τη δημιουργεί. Για κάθε πίνακα της βάσης χρησιμοποιήθηκε ένα διαφορετικό script. Στην περίπτωση των hairpins εστιάζει στην απομόνωση πληροφοριών από όλη την εγγραφή του αρχείου .dat, ενώ στην περίπτωση των matures εστιάζει στην άντληση πληροφοριών από το feature table (γραμμές FT) της κάθε εγγραφής. Τα αρχεία .dat βρίσκονται στην ίδια τοποθεσία με το script που θα τρέξει και έχουν μετονομαστεί ως DBX.X, όπου "X.X" είναι ο αριθμός έκδοσης του αρχείου όπως δίνεται από την mirbase. Ορίζεται επίσης στο script ένας πίνακας που περιέχει όλες τις εκδόσεις με την παραπάνω μορφή στη σειρά, ώστε να μπορούν να συνδεθούν με τους αύξοντες αριθμούς στους οποίους αντιστοιχούν. Ορίζονται επίσης πίνακες που στο τέλος της εκτέλεσης θα περιλαμβάνουν την πρώτη εμφάνιση ενός accession, καθώς ενημερώνονται κάθε φορά που μια εγγραφή διαβάζεται σε κάποια παλιότερη έκδοση των αρχείων .dat.

Το πρόγραμμα για τα hairpins εκτελείται στις ακόλουθες φάσεις:

1. Αρχικά διαβάζει όλα τα αρχεία .dat ξεκινώντας από το πιο πρόσφατο και καταλήγοντας στο πιο παλιό.
2. Κάθε αρχείο .dat που διαβάζεται χωρίζεται σε εγγραφές.
3. Για κάθε εγγραφή απομονώνονται με χρήση των regular expressions τα πεδία που ενδιαφέρουν και που μπορούν να ληφθούν με μια πρώτη ανάγνωση: AC, ID, DE, actual_last_appearance, last_appearance και SQ. Τα στοιχεία αυτά, εφόσον το accession εμφανίζεται για πρώτη φορά προστίθενται σε μια μεταβλητή string, όπου τα δεδομένα είναι οργανωμένα σε μορφή εντολής INSERT της SQL.
4. Αν το accession στο οποίο γίνεται επεξεργασία υπάρχει ήδη στην παραπάνω μεταβλητή, ελέγχεται αν συνοδεύεται από ίδιες τιμές ID και SQ.
5. Αν κάποια από τις παραπάνω διαφέρει, εισάγεται στην διαμορφούμενη εντολή SQL.
6. Όταν ολοκληρωθεί η διαδικασία εκτελείται η εντολή INSERT προς τη βάση. Πλέον έχουμε όλες τις εγγραφές, αλλά με συμπληρωμένα μόνο ορισμένα πεδία. Πρέπει ακόμα να συμπληρωθούν τα πεδία Change, first_appearance,

actual_first_appearance, comment, forward, new_appearance, forward_appearance, delete_appearance.

7. Στη συνέχεια για κάθε καταγεγραμμένο accession διαβάζουμε το ιστορικό του βάσει φθίνουσας τιμής στο last_appearance.
8. Ελέγχουμε σε ποιο πεδίο υπάρχει αλλαγή και διαμορφώνουμε αντίστοιχα την τιμή του πεδίου Change. Επίσης συγκρίνοντας με τις τιμές last_appearance και actual_last_appearance, της επόμενης σε σειρά εγγραφής συμπληρώνουμε τα πεδία first_appearance, actual_first_appearance. Επίσης μπορούμε να συμπληρώσουμε τα πεδία prev_sq και prev_name. Αν διαβάζουμε την τελευταία εγγραφή του accession (δηλαδή αυτή που αντιστοιχεί στην πρώτη εμφάνισή του) συμπληρώνουμε το πεδίο first_appearance, με την πρώτη συνολικά εμφάνιση από τον πίνακα που περιέχει αυτή την πληροφορία.
9. Μένουν πλέον οι περιπτώσεις διαγραφών: Διαλέγουμε το σύνολο των εγγραφών ομαδοποιημένων κατά accession, για τις οποίες η μέγιστη τιμή last_appearance είναι μικρότερη από αυτή που αντιστοιχεί στην τρέχουσα έκδοση.
10. Για κάθε μια από τις παραπάνω εγγραφές ελέγχουμε αν υπάρχουν στοιχεία της στο τρέχον αρχείο .dead. Τα στοιχεία αυτά απομονώνονται για να συμπληρωθεί το πεδίο comment και forward αντίστοιχα και δημιουργείται μια νέα εγγραφή με τιμή DELETE, ή FORWARD στο πεδίο Change και συμπληρωμένο μόνο το πεδίο first_appearance και actual_first_appearance.
11. Τέλος επιλέγουμε την ελάχιστη τιμή first_appearance και τη μέγιστη τιμή last_appearance για κάθε accession και συμπληρώνουμε ανάλογα τα πεδία new_appearance, delete_appearance, forward_appearance.

Αξίζει να σημειωθεί ότι ο όγκος των δεδομένων που είναι προς εισαγωγή στη βάση στο βήμα 6 είναι πολύ μεγάλος και ενδέχεται να χρειαστεί κάποια ρύθμιση ώστε ο server της βάσης να διαβάσει αυτά τα δεδομένα. Στην περίπτωσή μας το πρόβλημα λύθηκε με τη χρήση της εντολής:

```
set global max_allowed_packet = 16* 1024 * 1024
```

Η εντολή αυτή δίνεται στην mySQL και αυξάνει τον όγκο των δεδομένων που μπορούν να διαβαστούν στα 16MB.

Παρόμοια διαδικασία ακολουθείται και στην περίπτωση των mature. Η διαδικασία είναι πιο σύνθετη διότι κάθε εγγραφή ενός mature πρέπει να συσχετιστεί και με το

AC του hairpin που το παράγει. Αυτή τη φορά χρησιμοποιούμε δύο πίνακες που περιέχουν την έκδοση πρώτης εμφάνισης ενός mature. Ο ένας από αυτούς περιέχει την πρώτη συνολικά εμφάνιση ενός accession, ο δεύτερος την πρώτη έκδοση όπου εμφανίζεται σε σχέση με το hairpin που το παράγει.

Το πρόγραμμα για τα mature εκτελείται στις ακόλουθες φάσεις:

1. Διαβάζει όλα τα αρχεία .dat ξεκινώντας από το πιο πρόσφατο και καταλήγοντας στο πιο παλιό.
2. Κάθε αρχείο .dat που διαβάζεται χωρίζεται σε εγγραφές.
3. Για κάθε εγγραφή απομονώνονται το AC και το feature table της.
4. Για κάθε feature table απομονώνονται με χρήση των regular expressions τα πεδία που ενδιαφέρουν και που μπορούν να ληφθούν με μια πρώτη ανάγνωση: AC, ID, sequence_part, actual_last_appearance, last_appearance και SQ. Αυτό γίνεται για κάθε παραγόμενο mature. Τα στοιχεία αυτά, εφόσον το accession εμφανίζεται για πρώτη φορά παραγόμενο από το συγκεκριμένο hairpin, αποθηκεύονται σε πίνακες από τους οποίους στο τέλος θα συντεθούν ερωτήματα mySQL.
5. Αν το accession στο οποίο γίνεται επεξεργασία υπάρχει ήδη, παραγόμενο από το συγκεκριμένο hairpin, ελέγχεται αν συνοδεύεται από ίδιες τιμές ID και SQ.
6. Αν κάποια από τις παραπάνω διαφέρει, εισάγεται στον πίνακα που περιέχει δεδομένα από τα οποία θα δημιουργηθεί τελικά ένα ερώτημα SQL. Ταυτόχρονα προστίθεται η αλλαγή που έγινε σε σχέση με την προηγούμενη εγγραφή στην αντίστοιχη θέση το πίνακα. Σε περίπτωση που αναφερόμαστε σε ζεύγος hairpin- mature που ήδη καταγράφεται, αλλά τώρα έχει αλλαγή στα δεδομένα του, προσθέτουμε τιμή και στα πεδία actual_first_appearance και first_appearance.
7. Όταν ολοκληρωθεί η διαδικασία δημιουργείται και εκτελείται η εντολή INSERT προς τη βάση. Πλέον έχουμε όλες τις εγγραφές, αλλά με συμπληρωμένα μόνο ορισμένα πεδία. Πρέπει ακόμα να συμπληρωθούν τα πεδία, new_appearance, delete_appearance.
8. Στη συνέχεια για κάθε καταγεγραμμένο mature accession χρησιμοποιούμε τους πίνακες όπου αποθηκεύαμε τις τιμές των εκδόσεων πρώτης εμφάνισης. Συμπληρώνονται έτσι τα πεδία first_appearance και actual_first_appearance για τις περιπτώσεις του ιστορικού ενός mature βάσει του κάθε hairpin που το

παράγει, ενώ δημιουργείται και μια σειρά εγγραφών που αφορούν τη συνολική πρώτη εμφάνιση ενός mature accession και θα έχουν στο πεδίο Change την τιμή NEW. Στις περιπτώσεις όπου έχουμε την πρώτη εμφάνιση ενός mature βάσει του hairpin accession που το συνοδεύει, γίνεται επιπλέον έλεγχος αν το mature παραγόταν από άλλα hairpins σε προηγούμενες εκδόσεις και αν έχουν υπάρξει αλλαγές και στο ID, ή το SQ σε σχέση με αυτές, ώστε να προστεθούν οι αντίστοιχες εγγραφές.

9. Τέλος πρέπει να δημιουργηθούν οι εγγραφές DELETE. Αρχικά βρίσκουμε όλες τις εγγραφές με ζεύγος από accessions hairpin-mature, των οποίων η μέγιστη τιμή last_appearance είναι μικρότερη από την τρέχουσα. Για όλες αυτές τις περιπτώσεις θα πρέπει να δημιουργηθεί μια εγγραφή που θα έχει στο πεδίο Change την τιμή REMOVE PARENT HAIRPIN.
10. Η διαδικασία ολοκληρώνει με τη δημιουργία εγγραφών DELETE, οι οποίες προκύπτουν βρίσκοντας κάθε συνολικά τελική εμφάνιση ενός mature accession που δεν αντιστοιχεί στην τρέχουσα έκδοση.

Οι παραπάνω αλγόριθμοι χτίζουν τους πίνακες hairpinhistory και maturehistory γεμίζοντας τους με εγγραφές όπως περιγράφθηκαν στις παραγράφους 3.1.1 και 3.1.2.

3.1.4 Σύγκριση αποτελεσμάτων με τα αρχεία της mirbase

Στο σημείο αυτό προκύπτει η ανάγκη να ελεγχθεί η ορθότητα των δεδομένων που καταγράφονται στη βάση mySQL που δημιουργήσαμε. Η mirbase μας έχει ήδη διαθέσει τα κατάλληλα εργαλεία για αυτή τη δουλειά: τα αρχεία .diff. Μπορούμε πλέον να μελετήσουμε την δομή των αρχείων .diff και να εξάγουμε αντίστοιχα αρχεία από τη βάση που δημιουργήσαμε. Στη συνέχεια μπορούμε να κάνουμε συγκρίσεις των αντίστοιχων αρχείων. Με αυτόν τον τρόπο μπορούμε να εντοπίσουμε τυχόν παραλείψεις των αρχείων της mirbase, ή προβλήματα της βάσης που δημιουργήσαμε.

Ένα αρχείο .diff αποτελείται από γραμμές σαν την ακόλουθη:

MI0000395 mmu-mir-297-1 SEQUENCE

Έχουμε δηλαδή τρεις διακριτές στήλες. Η πρώτη αφορά το accession για το οποίο παρατηρείται αλλαγή, η δεύτερη το ID του και η τρίτη αναφέρει το είδος της αλλαγής. Σε περίπτωση που έχουμε αλλαγή NAME και SEQUENCE ταυτόχρονα έχουμε τέσσερις τέτοιες στήλες, καθώς χρειάζονται δύο για τις αλλαγές.

Η βάση που δημιουργήθηκε περιέχει αυτή τη πληροφορία και μπορούμε να εξάγουμε αρχεία .diff από αυτήν για κάθε έκδοση. Μπορούμε επίσης να εξάγουμε τέτοια αρχεία και για τις εκδόσεις στις οποίες η mirbase δεν τα διαθέτει. Τα αρχεία που δημιουργήθηκαν δεν περιέχουν την τιμή του ID καθώς το accession αποτελεί μοναδικό χαρακτηριστικό και ενδιαφέρει η καταγραφή της αλλαγής που έγινε. Εν συνεχεία με χρήση ενός script μπορούν να βρεθούν τυχόν διαφορές μεταξύ των αρχείων της mirbase και της βάσης που χτίστηκε, τουλάχιστον όσο αφορά τις εκδόσεις για τις οποίες η mirbase διαθέτει τέτοια αρχεία.

Ένα τέτοιο πρόγραμμα εκτελεί τα ακόλουθα βήματα:

1. Διαβάζει κάθε γραμμή των παρεχόμενων από την mirbase αρχείων .diff και αναζητά μια γραμμή με τα ίδια δεδομένα στα αρχεία που προέκυψαν από τη βάση.
2. Εκτελεί την ίδια διαδικασία συγκρίνοντας τις γραμμές των αρχείων .diff που εξήχθησαν από τη βάση με αυτές που υπάρχουν στα αρχεία της mirbase.
3. Για κάθε ασυνέπεια ενημερώνει με αντίστοιχο μήνυμα.

3.1.5 Υλοποίηση script για την αναβάθμιση της βάσης σε νέες εκδόσεις

Η mirbase τηρεί αρχείο για τα δεδομένα όλων των εκδόσεών της. Κατά τη χρήση της ιστοσελίδας η αναζητήσεις γίνονται πάντα με βάσει την τρέχουσα έκδοση. Μέχρι τη στιγμή που γράφεται το κείμενο έχουν κυκλοφορήσει 33 εκδόσεις. Δεδομένου ότι η διαδικασία που χρησιμοποιήθηκε για το χτίσιμο της βάσης πρέπει να διαβάζει τα αρχεία .dat όλων των εκδόσεων και να κάνει συγκρίσεις όλων των εγγραφών μια προς μια, είναι φανερό ότι δε μπορεί να εφαρμόζεται κάθε φορά εκ νέου όταν ανανεώνεται η mirbase. Η διαδικασία αυτή θα ήταν πολύ χρονοβόρα, ιδίως όταν οι εκδόσεις των αρχείων .dat έχουν εγγραφές της τάξης των μερικών χιλιάδων, ή δεκάδων χιλιάδων το κάθε ένα, οι οποίες πρέπει να συγκριθούν τόσες φορές, όσος είναι και ο αριθμός των εκδόσεων.

Προκύπτει συνεπώς η ανάγκη να δημιουργηθεί ένα πρόγραμμα που αναβαθμίζει την τρέχουσα μορφή της βάσης δεδομένων, λαμβάνοντας υπ’ όψιν όλα τα δεδομένα που περιέχονται σε καινούρια αρχεία .dat. Στη διαδικασία αυτή δε θα χρειάζεται να δημιουργεί τη βάση εκ νέου. Το πρόγραμμα αυτό θα εισάγει στη βάση τις νέες εγγραφές, ενώ θα αναβαθμίζει όσες από τις ήδη υπάρχουσες έχουν στην τιμή του

πεδίου last_appearance και actual_last_appearance την τιμή που αντιστοιχεί στην αμέσως προηγούμενη έκδοση.

Ενδεικτικά, το πρόγραμμα όσο αφορά στα hairpins εκτελεί τα ακόλουθα βήματα:

1. Διαβάζει όλες τις εγγραφές της βάσης που έχουν στην τιμή του πεδίου last_appearance και actual_last_appearance την τιμή που αντιστοιχεί στην αμέσως προηγούμενη έκδοση.
2. Διαβάζει όλες τις εγγραφές του καινούριου αρχείου .dat.
3. Συγκρίνει μια προς μια τις εγγραφές της βάσης με αυτές του αρχείου .dat. Ανάλογα με το αποτέλεσμα της σύγκρισης εκτελεί ένα από τα επόμενα βήματα:
    a. Σε περίπτωση που υπάρχει αλλαγή, δημιουργεί μια νέα εγγραφή με την ένδειξη της αλλαγής που παρατηρήθηκε.
    b. Σε περίπτωση που δεν υπάρχει αλλαγή θα πρέπει να γίνει update το πεδίο last_appearance και actual_last_appearance στην τιμή της νέας έκδοσης.
    c. Σε περίπτωση που κάποια εγγραφή της βάσης δε βρεθεί στην καινούρια έκδοση του αρχείου .dat δημιουργεί μια εγγραφή DELETE, ή FORWARD ανάλογα με τα δεδομένα που υπάρχουν στο καινούριο αρχείο .dead. Παράλληλα ενημερώνει όλα τα πεδία delete_appearance που αφορούν το συγκεκριμένο accession.
4. Συγκρίνει μια προς μια τις εγγραφές του αρχείου .dat με αυτές που υπάρχουν στη βάση. Έτσι εντοπίζονται εγγραφές που δεν υπάρχουν στη βάση και δημιουργούνται οι κατάλληλες εγγραφές NEW.
5. Στη συνέχεια παράγει ένα αρχείο .diff μόνο για την καινούρια έκδοση.
6. Τέλος συγκρίνει το αρχείο .diff που παράγεται με αυτό που δίνει η νέα έκδοση της mirbase, ειδοποιώντας κατάλληλα για τυχόν διαφορές.
7. Ένας επιπλέον έλεγχος γίνεται στα αρχεία .dead των δύο τελευταίων εκδόσεων ώστε να βρεθούν τυχόν διαφορές.

Η ίδια διαδικασία ακολουθείται και για τα mature micro RNAs. Η μόνη διαφορά είναι ότι ελέγχεται κάθε φορά μια εγγραφή mature σε σχέση με το hairpin που το παράγει. Τέλος, επειδή δεν υπάρχουν αρχεία .dead για τα mature micro RNAs το βήμα 7 της προηγούμενης διαδικασίας είναι περιττό.

3.1.6 Σενάρια χρήσης διαχειριστή της βάσης δεδομένων

Έχοντας πλέον συγκεντρωμένη μια δομή που περιέχει όλη την πληροφορία που αφορά την εξέλιξη των δεδομένων των εγγραφών για hairpins και matures μπορούμε να δούμε ορισμένα ερωτήματα που μπορεί να κάνει ένας διαχειριστής στη βάση, καθώς και να ερμηνεύσουμε τα αποτελέσματά τους.

Μια πρώτη βασική λειτουργία είναι η αναζήτηση όλης της ιστορίας που αφορά ένα συγκεκριμένο accession. Η λειτουργία αυτή ενδιαφέρει τόσο στον πίνακα hairpinhistory, όσο και στον πίνακα maturehistory. Έστω ότι θέλουμε να μάθουμε την ιστορία του hairpin με accession MI0000001. Το ερώτημα μπορεί να διατυπωθεί είναι:

```
SELECT AC, ID, first_appearance, last_appearance, actual_first_appearance,
                actual_last_appearance, DE, SQ, hairpinhistory.Change, forward,
               comment
FROM  `hairpinhistory`
WHERE AC =  'MI0000001'
ORDER BY first_appearance ASC
```

Τα πεδία που επιλέγονται είναι ακριβώς αυτά που χρειάζονται για να πάρουμε όλη την πληροφορία που αφορά την ιστορία του hairpin με accession MI0000001. Η απαίτηση να εμφανιστούν με σειρά αύξουσας πρώτης εμφάνισης τοποθετεί τις εγγραφές που αφορούν το accession σε σειρά εμφάνισης.

Με παρόμοιο τρόπο εκφράζεται το ερώτημα και για τα mature. Έστω ότι ενδιαφέρει το accession MIMAT0000001. Το ερώτημα διατυπώνεται ως:

```
SELECT AC, ID, mother_hairpin, first_appearance, last_appearance,
actual_first_appearance, actual_last_appearance, SQ, maturehistory.Change
FROM  `maturehistory`
WHERE AC =  'MIMAT0000001'
ORDER BY mother_hairpin, first_appearance ASC
```

AC	ID	first_appearance	last_appearance	actual_first_appearance	actual_last_appearance	DE	SQ	Change	forward	comment
MI0000001	cel-let-7L	1	6	1.0	1.5	Caenorhabditis elegans let-7 precursor RNA	[BLOB - 200B]	NEW	*NULL*	*NULL*
MI0000001	cel-let-7	7	33	2.0	18.0	Caenorhabditis elegans let-7 stem-loop	[BLOB - 200B]	NAME	*NULL*	*NULL*

Εικόνα 1 - Ιστορικό του hairpin MI0000001

AC	ID	mother_hairpin	first_appearance	last_appearance	actual_first_appearance	actual_last_appearance	SQ	Change
MIMAT0000001	cel-let-7	*NULL*	15	*NULL*	6.0	*NULL*	[BLOB - 22B]	NEW
MIMAT0000001	cel-let-7	MI0000001	15	32	6.0	17.0	[BLOB - 22B]	ADD PARENT HAIRPIN
MIMAT0000001	cel-let-7-5p	MI0000001	33	33	18.0	18.0	[BLOB - 22B]	NAME

Εικόνα 2 - Ιστορικό του mature MIMAT0000001

Στην εικόνα 1 δίνεται το αποτέλεσμα του πρώτου ερωτήματος. Η πληροφορία που δίνει το σύνολο αποτελεσμάτων είναι η ακόλουθη: Το hairpin με accession MI0000001 εμφανίζεται για πρώτη φορά στην έκδοση 1.0, με ID cel-let-7L. Τα στοιχεία του παραμένουν αμετάβλητα μέχρι και την έκδοση 1.5. Στην έκδοση 2.0 εμφανίζεται αλλαγή στο όνομα του hairpin, το οποίο πλέον είναι cel-let-7. Τα στοιχεία αυτά παραμένουν μέχρι και την έκδοση 18.0. Δεδομένου ότι δεν έχουμε εγγραφή που να συνοδεύεται με την τιμή DELETE, ή FORWARD στο πεδίο Change, μπορούμε να βγάλουμε το συμπέρασμα ότι η έκδοση 18.0 είναι και η τρέχουσα έκδοση της mirbase.

Στην εικόνα 2 δίνεται το αποτέλεσμα του δεύτερου ερωτήματος. Ποια πληροφορία περιέχεται στο σύνολο αποτελεσμάτων; Αρχικά μαθαίνουμε ότι η πρώτη εμφάνιση του accession MIMAT0000001 έγινε στην έκδοση 6.0, όπως πληροφορεί η ειδική εγγραφή με την τιμή NEW στο πεδίο Change. (Υπενθυμίζουμε ότι η έκδοση 6.0 είναι η πρώτη στην οποία καταγράφονται accessions για mature micro RNAs). Επιπλέον επειδή ζητήθηκε ομαδοποίηση των αποτελεσμάτων με βάση το πεδίο mother_hairpin και έχουμε μόνο μια τιμή στο πεδίο αυτό, πληροφορούμαστε ότι το mature αυτό παράγεται μόνο από ένα hairpin, αυτό δηλαδή που έχει accession MI0000001. Το mature εμφανίζεται για πρώτη φορά με όνομα cel-let-7 στην έκδοση 6.0 και διατηρεί τα στοιχεία του μέχρι και την έκδοση 17.0. Στην έκδοση 18.0 έχουμε αλλαγή στο όνομα, το οποίο πλέον είναι cel-let-7-5p. Όπως και στην περίπτωση του hairpin, έτσι και εδώ, λόγω έλλειψης εγγραφής με την τιμή DELETE στο πεδίο Change, μαθαίνουμε ότι η έκδοση 18.0 είναι η τρέχουσα και το mature εξακολουθεί να υπάρχει.

Ένα άλλο ερώτημα που μπορεί να ενδιαφέρει είναι η παρουσίαση όλων των αλλαγών που αφορούν μια συγκεκριμένη έκδοση. Για παράδειγμα μπορεί να ενδιαφέρει η καταγραφή όλων των αλλαγών που έγιναν από την έκδοση 1.3 στην έκδοση 1.4. Το ερώτημα διατυπώνεται ως εξής:

```
SELECT AC, ID, first_appearance, last_appearance, actual_first_appearance,
        actual_last_appearance, SQ, hairpinhistory.Change, prev_name, prev_sq
FROM hairpinhistory
WHERE actual_first_appearance =  '1.4'
```

AC	ID	first_appearance	last_appearance	actual_first_appearance	actual_last_appearance	SQ	Change	prev_name	prev_sq
MI0000342	hsa-mir-200b	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000343	dme-mir-263	5	6	1.4	1.5	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000344	cel-mir-264	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000345	cel-mir-265	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000346	cel-mir-266	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000347	cel-mir-267	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000348	cel-mir-268	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000349	cel-mir-269	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000350	cel-mir-270	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]
MI0000351	cel-mir-271	5	33	1.4	18.0	[BLOB - 200B]	NEW	*NULL*	[BLOB - 0B]

Εικόνα 3 - Αλλαγές hairpins στην έκδοση 1.4

Στην προηγούμενη εικόνα δίνεται ένα τμήμα του συνόλου αποτελεσμάτων του ερωτήματος που διατυπώθηκε. Παρατηρούμε ένα αναμενόμενο αποτέλεσμα: το μεγαλύτερο μέρος των αλλαγών που καταγράφονται για νέες εκδόσεις αφορούν καινούριες εγγραφές που δεν υπήρχαν προηγούμενα στη βάση. Η εμφάνιση των πεδίων prev_name και prev_sq γίνεται ώστε να φαίνονται απευθείας οι αλλαγές που παρατηρούνται στις περιπτώσεις όπου στο πεδίο Change έχουμε τις τιμές NAME, ή SEQUENCE.

Αντίστοιχο ερώτημα μπορεί να διατυπωθεί και στην περίπτωση που θέλουμε να βρούμε όλες τις αλλαγές που παρατηρούνται για μια δεδομένη έκδοση και αφορούν matures. Σε αυτή τη περίπτωση ενδέχεται να θέλουμε να δούμε ομαδοποιημένες τις αλλαγές, όσο αφορά τα accessions. Δηλαδή ενδέχεται να μην ενδιαφέρει λεπτομερώς ποιες αλλαγές παρατηρούνται σε κάθε ζευγάρι accession hairpin- mature. Στην περίπτωση αυτή, αν αναζητάμε το σύνολο των αλλαγών από την έκδοση 6.0 στην 7.0, το ερώτημα πρέπει να διατυπωθεί ως εξής:

```
SELECT AC, ID, mother_hairpin, first_appearance, last_appearance,
        actual_first_appearance, actual_last_appearance,
        SQ, maturehistory.Change, prev_name, prev_sq
FROM maturehistory
WHERE actual_first_appearance =  '7.0'
AND maturehistory.Change <>  'ADD PARENT HAIRPIN'
GROUP BY AC
```

Στο παραπάνω ερώτημα υπάρχει απαίτηση να μη λαμβάνονται υπ’ όψιν οι εγγραφές που έχουν την τιμή 'ADD PARENT HAIRPIN' στο πεδίο Change. Αυτό έγινε, γιατί όπως έχει αναφερθεί, σε περίπτωση που έχουμε ταυτόχρονα την πρώτη εμφάνιση όσο αφορά ένα νέο hairpin- παραγωγό του mature, και ταυτόχρονα αλλαγή του ID, ή SQ σε σχέση με προηγούμενες εγγραφές του mature accession, στη βάση υπάρχουν δύο διακριτές εγγραφές. Από αυτές ενδιαφέρει αυτή που αφορά την αλλαγή στο ID, ή SQ.

Τέλος γίνεται GROUP BY, ώστε μια αλλαγή NAME, ή SQ που αφορά το ίδιο mature accession, από όσα hairpins και αν παράγεται, να εμφανίζεται μόνο μια φορά.

Ένα άλλο ερώτημα που μπορεί να ενδιαφέρει είναι να βρεθούν όλες οι εγγραφές των hairpins, ή mature που εμφανίζονται σε μια συγκεκριμένη έκδοση. Δηλαδή να

παρουσιαστεί ισοδύναμη πληροφορία με αυτή που περιέχεται σε ένα αρχείο .dat. Έστω ότι μας ενδιαφέρει η έκδοση 7.0. Η έκδοση αυτή αντιστοιχεί στον αύξοντα αριθμό 16. Το ερώτημα επομένως όσο αφορά τα hairpins διατυπώνεται ως εξής:

```
SELECT AC, ID, first_appearance, last_appearance, actual_first_appearance,
        actual_last_appearance, DE, SQ, hairpinhistory.Change
FROM hairpinhistory
WHERE first_appearance <= 16
AND last_appearance >= 16
```

Με αντίστοιχο τρόπο διατυπώνεται και το ερώτημα όσο αφορά τα matures:

```
SELECT AC, ID, mother_hairpin, first_appearance, last_appearance,
        actual_first_appearance, actual_last_appearance, SQ, maturehistory.Change
FROM maturehistory
WHERE first_appearance <=16
AND last_appearance >=16
ORDER BY AC
```

Στην περίπτωση αυτή προστέθηκε το ORDER BY AC, το οποίο θα παρουσιάσει σε διαδοχικές εγγραφές τα mature που παράγονται από διαφορετικά hairpins.

Τμήματα των αποτελεσμάτων των παραπάνω ερωτημάτων παρουσιάζονται στις εικόνες 4 και 5.

AC	ID	first_appearance	last_appearance	actual_first_appearance	actual_last_appearance	DE	SQ	Change
MI0000001	cel-let-7	7	33	2.0	18.0	Caenorhabditis elegans let-7 stem-loop	[BLOB - 200B]	NAME
MI0000002	cel-lin-4	10	33	3.0	18.0	Caenorhabditis elegans lin-4 stem-loop	[BLOB - 200B]	NAME
MI0000003	cel-mir-1	1	33	1.0	18.0	Caenorhabditis elegans miR-1 stem-loop	[BLOB - 200B]	NEW
MI0000004	cel-mir-2	1	33	1.0	18.0	Caenorhabditis elegans miR-2 stem-loop	[BLOB - 200B]	NEW
MI0000005	cel-mir-34	1	33	1.0	18.0	Caenorhabditis elegans miR-34 stem-loop	[BLOB - 200B]	NEW
MI0000006	cel-mir-35	1	33	1.0	18.0	Caenorhabditis elegans miR-35 stem-loop	[BLOB - 200B]	NEW
MI0000007	cel-mir-36	1	33	1.0	18.0	Caenorhabditis elegans miR-36 stem-loop	[BLOB - 200B]	NEW
MI0000008	cel-mir-37	1	33	1.0	18.0	Caenorhabditis elegans miR-37 stem-loop	[BLOB - 200B]	NEW
MI0000009	cel-mir-38	1	33	1.0	18.0	Caenorhabditis elegans miR-38 stem-loop	[BLOB - 200B]	NEW
MI0000010	cel-mir-39	10	33	3.0	18.0	Caenorhabditis elegans miR-39 stem-loop	[BLOB - 200B]	SEQUENCE

Εικόνα 4 - hairpins της έκδοσης 7.0

AC ▲	ID	mother_hairpin	first_appearance	last_appearance	actual_first_appearance	actual_last_appearance	SQ	Change
MIMAT0000344	dme-miR-281-1*	MI0000366	15	31	6.0	16.0	[BLOB - 22B]	ADD PARENT HAIRPIN
MIMAT0000345	dme-miR-281	MI0000366	15	31	6.0	16.0	[BLOB - 23B]	ADD PARENT HAIRPIN
MIMAT0000345	dme-miR-281	MI0000370	15	31	6.0	16.0	[BLOB - 23B]	ADD PARENT HAIRPIN
MIMAT0000346	dme-miR-282	MI0000367	15	31	6.0	16.0	[BLOB - 28B]	ADD PARENT HAIRPIN
MIMAT0000347	dme-miR-283	MI0000368	15	31	6.0	16.0	[BLOB - 21B]	ADD PARENT HAIRPIN
MIMAT0000348	dme-miR-284	MI0000369	15	31	6.0	16.0	[BLOB - 29B]	ADD PARENT HAIRPIN
MIMAT0000349	dme-miR-281-2*	MI0000370	15	31	6.0	16.0	[BLOB - 22B]	ADD PARENT HAIRPIN
MIMAT0000350	dme-miR-34	MI0000371	15	23	6.0	9.2	[BLOB - 22B]	ADD PARENT HAIRPIN
MIMAT0000351	dme-miR-124	MI0000373	15	31	6.0	16.0	[BLOB - 23B]	ADD PARENT HAIRPIN
MIMAT0000352	dme-miR-79	MI0000374	15	31	6.0	16.0	[BLOB - 22B]	ADD PARENT HAIRPIN

Εικόνα 5- matures της έκδοσης 7.0 1

Στα παραπάνω αρκεί η έκδοση 7.0 να βρίσκεται μεταξύ των εκδόσεων που αναγράφονται στα πεδία actual_first_appearance και actual_last_appearance, καθώς η κάθε εγγραφή της βάσης αναφέρεται σε διάστημα εκδόσεων όπου ένα accession έχει σταθερά στοιχεία ID και SQ.

Τέλος μπορεί να ενδιαφέρει η απεικόνιση όλων των hairpins, ή matures που έχουν ένα συγκεκριμένο όνομα σε κάποια στιγμή της καταγραφής τους. Έστω ότι αναζητάμε τα hairpins που φέρουν σε κάποια έκδοση το όνομα mmu-mir-1b. Το ερώτημα που διατυπώνεται είναι το ακόλουθο:

```
SELECT AC, ID, first_appearance, last_appearance, actual_first_appearance,
        actual_last_appearance, SQ, hairpinhistory.Change
FROM  `hairpinhistory`
WHERE ID =  'mmu-mir-1b'
ORDER BY AC, first_appearance ASC
```

Αντίστοιχα μπορεί να αναζητάμε τα mature που φέρουν σε κάποια έκδοση το όνομα kshv-miR-K12-12*. Το ερώτημα διατυπώνεται ως:

```
SELECT AC, ID, mother_hairpin, first_appearance, last_appearance,
        actual_first_appearance, actual_last_appearance, SQ, maturehistory.Change
FROM  `maturehistory`
WHERE ID =  'kshv-miR-K12-12*'
ORDER BY AC, mother_hairpin, first_appearance ASC
```

Στην εικόνα 6 δίνεται το αποτέλεσμα του πρώτου ερωτήματος. Το σύνολο των εγγραφών που επιστράφηκε μας πληροφορεί ότι το ID mmu-mir-1b στις εκδόσεις 1.1 ως 2.0 ανήκε στο accession MI0000258. Στην έκδοση 2.2, το ID συσχετιζόταν με το hairpin με accession MI0000652, ενώ στις εκδόσεις 17.0 ως 18.0 το ID αποδίδεται στο haipin MI0006283.

Στην εικόνα 7 έχουμε τα αποτελέσματα του δεύτερου ερωτήματος. Το ID kshv-miR-K12-12* συνδέεται με δύο matures. Στην έκδοση 15.0 το φέρει το mature με accession MIMAT0015238 που παράγεται από το hairpin MI0004987. Στις εκδόσεις 16.0 ως 18.0 το όνομα ανήκει το MIMAT0003712 που παράγεται από το hairpin MI0004897.

AC	ID	first_appearance	last_appearance	actual_first_appearance	actual_last_appearance	SQ	Change
MI0000258	mmu-mir-1b	2	7	1.1	2.0	[BLOB - 200B]	NEW
MI0000652	mmu-mir-1b	9	9	2.2	2.2	[BLOB - 200B]	NEW
MI0006283	mmu-mir-1b	32	33	17.0	18.0	[BLOB - 281B]	NAME

Εικόνα 6 – Σύνολο hairpins που συνδέονται με το όνομα mmu-mir-1b

AC	ID	mother_hairpin	first_appearance	last_appearance	actual_first_appearance	actual_last_appearance	SQ	Change
MIMAT0003712	kshv-miR-K12-12*	MI0004987	31	33	16.0	18.0	[BLOB - 23B]	NAME
MIMAT0015238	kshv-miR-K12-12*	*NULL*	30	*NULL*	15.0	*NULL*	[BLOB - 22B]	NEW
MIMAT0015238	kshv-miR-K12-12*	MI0004987	30	30	15.0	15.0	[BLOB - 22B]	ADD PARENT HAIRPIN

Εικόνα 7 – Σύνολο mature που σχετίζονται με το όνομα kshv-miR-K12-12*

3.1.7 Περιπτώσεις εσφαλμένων αποτελεσμάτων λόγω ασυνέπειας των καταγεγραμμένων δεδομένων της mirBase

Όπως αναφέρθηκε οι αλλαγές που καταγράφονται στα αρχεία .diff της mirbase συγκρίθηκαν με αντίστοιχα αρχεία που εξήχθηκαν από τη βάση mySQL που δημιουργήσαμε. Η σύγκριση αυτή έδειξε ορισμένες περιπτώσεις όπου τα δεδομένα μιας εγγραφής καταγράφονται με προβληματικό τρόπο στα αρχεία .dat. Στις περιπτώσεις αυτές ενδέχεται τα ερωτήματα SQL προς τη βάση να παρουσιάζουν ασαφή, ή εσφαλμένα αποτελέσματα.

Για παράδειγμα υπάρχουν τρεις περιπτώσεις εγγραφών που αφορούν matures, όπου τα accessions τους εμφανίζονται στο ίδιο διάστημα εκδόσεων με περισσότερα από ένα ID. Αναλυτικά οι περιπτώσεις αυτές είναι:

1. Το MIMAT0004283 καταγράφεται στα αρχεία .dat των εκδόσεων 9.1-18.0 με ID ath-miR854d. Καταγράφεται ωστόσο στην έκδοση 15.0 και με ένα **δεύτερο,** διαφορετικό ID, το ath-miR854e.
2. Το MIMAT0005413 καταγράφεται στην έκδοση 15.0 με IDs cre-miR1161a αλλά και με ID cre-miR1161b.
3. Το MIMAT0002860 καταγράφεται στις εκδόσεις 10.0-10.1 με ID hsa-miR-516a-3p. Ταυτόχρονα καταγράφεται στις εκδόσεις 10.0 – 17.0 με ID has-miR-516b*. Δηλαδή υπάρχουν δύο διαφορετικά ID για το ίδιο accession στις εκδόσεις 10.0-10.1 σύμφωνα με τα αρχεία .dat.

Σύμφωνα με τα παραπάνω, θα έπρεπε στην πρώτη περίπτωση να καταγράφεται μια αλλαγή NAME στην έκδοση 15.0, αλλά και στην έκδοση 16.0, καθώς στη μια περίπτωση εμφανίζεται ένα νέο όνομα για το accession, ενώ στη δεύτερη περίπτωση εμφανίζεται ξανά το παλιό όνομα για το MIMAT0004283. Οι αλλαγές αυτές καταγράφονται στα αρχεία .diff της mibrase. Ωστόσο δε φαίνονται στη βάση.

Στη δεύτερη περίπτωση επίσης θα έπρεπε να έχουμε αλλαγές NAME στις εκδόσεις 15.0 και 16.0 για τους ίδιους λόγους που αφορούν και την πρώτη περίπτωση. Οι αλλαγές αυτές και πάλι δε φαίνονται στη βάση.

Στην τελευταία περίπτωση αντίστοιχα θα έπρεπε να έχουμε μια αλλαγή NAME στην έκδοση 11.0 όπου χάνεται το ένα από τα δύο καταγεγραμμένα ονόματα.

Σε κάθε μια από τις παραπάνω περιπτώσεις ωστόσο τίθεται το ζήτημα κατά πόσο είναι επιτρεπτή η διπλή ονομασία της ίδιας οντότητας, αφού ως μοναδικό χαρακτηριστικό κάθε εγγραφής θεωρείται ο κωδικός αριθμός accession.

Στα παραπάνω προστίθεται μια ιδιαίτερη περίπτωση που αφορά το mature με accession MIMAT0000687. Για το miRNA αυτό, τα δεδομένα που καταγράφονται παρουσιάζουν την ακόλουθη περίεργη συμπεριφορά: Στο αρχείο .dat της έκδοσης 6.0 καταγράφεται ότι παράγεται από το hairpin με accession MI0000744. Στην έκδοση αυτή έχει ID hsa-miR-299 και αντιστοιχεί το τμήμα 7..28 της αλληλουχίας του hairpin. Ωστόσο στο αρχείο .dat της έκδοσης 7.0 το hairpin MI0000744 καταγράφεται ότι παράγει **δύο** miRNA, τα οποία εμφανίζονται και τα δύο με το ίδιο accession, MIMAT0000687. Αντιστοιχούν στα τμήματα 7..28 και 39..60 της ακολουθίας του hairpin. Στην έκδοση 7.1 το hairpin MI0000744 παράγει και πάλι δύο miRNAs, σύμφωνα με το αρχείο .dat, τα οποία έχουν accessions MIMAT0002890 και MIMAT0000687 αντίστοιχα. Εδώ όμως το MIMAT0000687 αντιστοιχεί πλέον στο τμήμα ακολουθίας 39..60 του hairpin MI0000744.

Τα προβλήματα που προκύπτουν είναι δύο: Αφενός φαίνεται ότι υπάρχουν δύο miRNA με το ίδιο accession τα οποία παράγονται από διαφορετικά τμήματα ακολουθίας από το ίδιο hairpin στην έκδοση 7.0. Αφετέρου φαίνεται ότι το hairpin με accession MI0000744 στην έκδοση 6.0 καταγράφεται να παράγει το MIMAT0000687 από το τμήμα 7..28 της ακολουθίας του, ενώ στην έκδοση 7.1 το ίδιο τμήμα ακολουθίας του hairpin αποδίδεται στο MIMAT0002890. Προκύπτει λοιπόν το ερώτημα αν είναι δυνατόν να αποδοθεί το ίδιο accession σε διαφορετικά matures που παράγονται από το ίδιο hairpin. Επιπλέον πρέπει να εξεταστεί αν έχει υπάρχει ανταλλαγή μεταξύ των δεδομένων των δύο προϊόντων του hairpin MI0000744 στις εκδόσεις 6.0-7.1.

Εξαιτίας των παραπάνω η βάση σε περίπτωση που ζητηθεί η ιστορία για το MIMAT0000687 θα δώσει εσφαλμένα αποτελέσματα, καθότι θα περιέχει μεταξύ των άλλων και εγγραφές στις οποίες η τιμή του πεδίου actual_last_appearance αντιστοιχεί σε μικρότερη έκδοση από αυτή του actual_first_appearance.

Η λύση των προβλημάτων που περιγράφηκαν παραπάνω απαιτεί αρχικά τη διερεύνηση κατά πόσον είναι ορθά τα καταγεγραμμένα στοιχεία των αρχείων .dat. Η απάντηση στο ερώτημα αυτό μπορεί να δοθεί από τους ερευνητές- βιολόγους που κατέγραψαν τα δεδομένα αυτά. Στη συνέχεια, σε περίπτωση που πράγματι πρόκειται

για εσφαλμένη καταγραφή, μπορεί να διορθωθούν τα δεδομένα της βάσης. Αυτό μπορεί να γίνει είτε την επέμβαση με απευθείας εκτέλεση εντολών UPDATE στις κατάλληλες εγγραφές, ή με διόρθωση των αρχείων .dat και εν συνεχεία εκτέλεση εκ νέου των προγραμμάτων που δημιουργούν τη βάση mySQL.

4

Σχεδίαση (ΙΙ) και ανάπτυξη εφαρμογής παρουσίασης εξελικτικών γράφων

Στο κεφάλαιο αυτό παρουσιάζουμε τις λειτουργίες που πρέπει να εκτελεί η εφαρμογή προς υλοποίηση, καθώς και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη και τη σχεδίαση της. Στη συνέχεια παρουσιάζονται οι λεπτομέρειες της υλοποίησής της, ενώ η τελευταία παράγραφος παρουσιάζει τη χρήση της ολοκληρωμένης εφαρμογής.

4.1 Ανάλυση απαιτήσεων συστήματος

Το σύστημα που θέλουμε να κατασκευάσουμε αποτελεί στην αφηρημένη του μορφή ένα σύστημα παρουσίασης της εξέλιξης ορολογιών, όπως καταγράφονται κατά την εξέλιξη μιας βάσης δεδομένων. Το σύστημα θα πρέπει δηλαδή να παρουσιάζει την διαφορετική μορφή με την οποία καταγράφονται κάποιες ορολογίες, ή δεδομένα, σε διαφορετικές εκδόσεις. Οι ορολογίες στην συγκεκριμένη τους μορφή θα είναι τα ID και οι ακολουθίες που καταγράφονται για μόρια micro RNA.

Η εφαρμογή καθεαυτή στοχεύουμε να μπορεί να ενσωματωθεί στην ιστοσελίδα DIANA. Δηλαδή να μπορεί να επεκτείνει το ήδη υπάρχον λογισμικό (για παράδειγμα το microT v4.0), αλλά να μπορεί να αποτελέσει και τμήμα λογισμικού που θα

αναπτυχθεί στο μέλλον. Η ιστοσελίδα DIANA και οι εφαρμογές που διαθέτει έχουν αναπτυχθεί χρησιμοποιώντας την Yii, μια δημοφιλή PHP πλατφόρμα. Επομένως για την ενσωμάτωση της προς ανάπτυξη εφαρμογής, απαιτείται η χρήση της πλατφόρμας αυτής. Συνεπώς, επιλέξαμε η εφαρμογή παρουσίασης γράφων εξέλιξης να αναπτυχθεί ως component της πλατφόρμας Yii. Συγκεκριμένα ο στόχος ήταν να αναπτυχθεί ένα Yii component, το οποίο στη συνέχεια θα ενσωματωνόταν στην διεπαφή του εργαλείου microT v4.0 της ιστοσελίδας DIANA.

Οι εφαρμογές της ιστοσελίδας DIANA ενδέχεται να διατηρούν δεδομένα που αφορούν τα micro RNAs μόνο για κάποια δεδομένη έκδοση της βάσης mirbase. Η εφαρμογή που θα αναπτυχθεί πρέπει να διευκολύνει στην αναζήτηση πληροφοριών για ένα συγκεκριμένο micro RNA, στην περίπτωση που συμβαίνει αυτό. Αυτό θα επιτυγχάνεται με τη συνοπτική παρουσίαση όλου του ιστορικού της εξέλιξης των δεδομένων που καταγράφονται για το micro RNA αυτό. Προκύπτουν συνεπώς οι ακόλουθες λειτουργικές απαιτήσεις για την εφαρμογή.

Αναζήτηση βάσει accession

Ο χρήστης έχει τη δυνατότητα να εισάγει σε μια φόρμα το accession ενός micro RNA είτε αυτό είναι hairpin, είτε είναι mature. Μια τέτοια φόρμα δίνεται ήδη από την εφαρμογή DIANA microT v4.0. Ο γράφος που θα προβληθεί θα παρουσιάζει την εξέλιξη των δεδομένων που αφορούν την εγγραφή με το συγκεκριμένο accession.

Αναζήτηση βάσει ID

Ο χρήστης έχει τη δυνατότητα να εισάγει σε μια φόρμα το ID ενός micro RNA είτε αυτό είναι hairpin, είτε είναι mature. Ο γράφος που θα προβληθεί θα παρουσιάζει την εξέλιξη των δεδομένων όλων των εγγραφών που σε κάποια έκδοση, ή σε κάποιο διάστημα εκδόσεων είχαν το ID το οποίο αναζήτησε ο χρήστης. Το ID ενδέχεται να αφορά τόσο hairpins, όσο και matures, με αποτέλεσμα να προβάλλονται ξεχωριστά γράφοι που αφορούν την κάθε κατηγορία βιομορίων.

Παρουσίαση γράφου εξέλιξης δεδομένων

Η αναζήτηση που μπορεί να γίνει στη φόρμα όπου εισάγεται κλειδί αναζήτησης μπορεί να αποτελείται από περισσότερες από μια λέξεις. Κάθε λέξη που δίνεται προς αναζήτηση μπορεί να είναι ένα ID, ή ένα accession. Για κάθε τέτοια λέξη δίνεται η δυνατότητα προβολής του γράφου εξέλιξης δεδομένων, αφού ο χρήστης πατήσει ένα

κατάλληλο κουμπί. Δίνεται επιπλέον η δυνατότητα εναλλαγής της προβολής, ή απόκρυψης των γράφων που προκύπτουν για κάθε λέξη-κλειδί.

Ο γράφος που προβάλλεται αποτελείται από κόμβους που αφορούν την πρώτη έκδοση, όπου ένα micro RNA εμφανίζεται με συγκεκριμένα δεδομένα (ID και SQ). Οι κόμβοι συνδέονται μεταξύ τους με κατευθυνόμενες ακμές, οι οποίες αναγράφουν το είδος της αλλαγής που παρατηρείται. Σε περίπτωση που η τελευταία αλλαγή δεν αντιστοιχεί στην τρέχουσα έκδοση, αλλά σε παλιότερη, ενώ παράλληλα έχουμε αμετάβλητη καταγραφή δεδομένων έχουμε μια ακμή προς την τρέχουσα έκδοση, στην οποία αναγράφεται ότι δεν έχει πραγματοποιηθεί αλλαγή.

Οι κόμβοι του γράφου που αντιστοιχούν σε έκδοση όπου υπάρχει κάποιο είδος διαγραφής ενός accession χρωματίζονται με διαφορετικό τρόπο, ώστε να ξεχωρίζουν μεταξύ τους. Για παράδειγμα, σε περίπτωση συνολικής διαγραφής ενός mature accession οι κόμβοι θα χρωματίζονται κόκκινοι, ενώ σε περίπτωση που ένα mature παύει να παράγεται από συγκεκριμένο hairpin και παράλληλα εξακολουθεί να καταγράφεται στη βάση παραγόμενο από άλλα, το χρώμα είναι ιώδες. Στην περίπτωση που έχουμε προώθηση ενός hairpin σε νέο accession ο κόμβος περιέχει την εικόνα ενός πράσινου βέλους.

Προβολή λεπτομερειών

Κάθε κόμβος του γράφου εξέλιξης δεδομένων πρέπει να παρέχει τη δυνατότητα, με το πάτημα ενός κατάλληλου κουμπιού, να προβάλλει όλα τα καταγεγραμμένα από τη βάση δεδομένα για τη συγκεκριμένη έκδοση στην οποία αντιστοιχεί. Τα στοιχεία αυτά αφορούν το διάστημα των εκδόσεων στο οποίο η εγγραφή έχει αμετάβλητα δεδομένα, δηλαδή μέχρι την έκδοση αμέσως πριν τον επόμενο κόμβο.

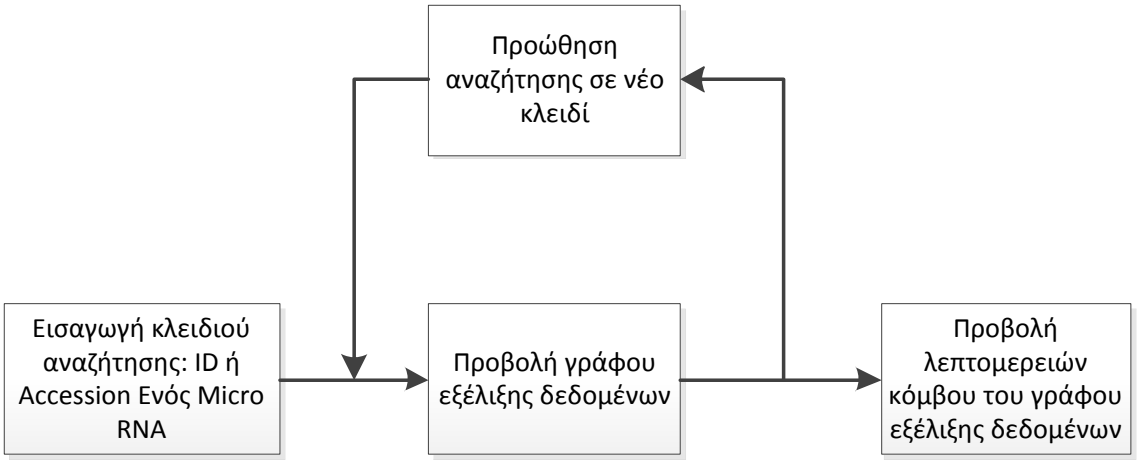
Τα δεδομένα αυτά θα παρουσιάζονται με τη μορφή ενός αναδυόμενου παράθυρου.

Προώθηση αναζήτησης σε νέα λέξη- κλειδί

Κάθε κόμβος του γράφου εξέλιξης που προβάλλεται περιέχει το ID του micro RNA. Το ID που περιέχεται αντιστοιχεί στο micro RNA από την έκδοση του κόμβου, μέχρι την έκδοση του επόμενου κόμβου. Εξαίρεση αποτελούν κόμβοι που αφορούν την έκδοση στην οποία έχουμε FORWARD για ένα hairpin. Σε αυτή τη περίπτωση ο κόμβος περιέχει το accession του hairpin στο οποίο γίνεται η προώθηση.

Κάθε κόμβος θα πρέπει να δίνει τη δυνατότητα με το πάτημα του κατάλληλου κουμπιού να γίνεται προώθηση της αναζήτησης στο περιεχόμενο (ID, ή accession) του κόμβου.

Συνοπτικά ο χρήστης της εφαρμογής θα δίνει λέξεις προς αναζήτηση, θα προβάλλει τα αποτελέσματα, θα προωθεί τις αναζητήσεις του σε άλλες λέξεις και τέλος, θα προβάλλει τα λεπτομερώς καταγεγραμμένα δεδομένα ενός micro RNA για ένα διάστημα εκδόσεων. Οι λειτουργίες αυτές δίνονται σχηματικά στο ακόλουθο σχήμα:



4.1 Σχηματική αναπαράσταση λειτουργιών προβολής γράφων εξέλιξης

4.2 Τεχνολογίες, πλατφόρμες και προγραμματιστικά εργαλεία

Στη συνέχεια περιγράφουμε τις προγραμματιστικές πλατφόρμες, τις τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Το βασικό εργαλείο ανάπτυξης της εφαρμογής ήταν το XAMPP. Το XAMPP είναι μια διανομή του apache web server που συμπεριλαμβάνει τη mySQL, τις γλώσσες php και Perl. Το εργαλείο αυτό διατίθεται δωρεάν. Η λειτουργικότητά του είναι να επιτρέπει την ανάπτυξη και τον έλεγχο διαδικτυακών εφαρμογών, παρέχοντας ένα περιβάλλον τοπικού εξυπηρετητή αιτήσεων.

Η επιλογή αυτού του εργαλείου έγινε διότι παρέχει το σύνολο των απαιτούμενων συστατικών της εφαρμογής (βάση δεδομένων mySQL, http εξυπηρετητής, php για web development) σε ένα ενιαίο πακέτο.

4.2.1. Apache http server

Ο apache http web server είναι ένας εξυπηρετητής του διαδικτύου. Αποτελεί μια πλατφόρμα λογισμικού ανοιχτού κώδικα που χρησιμοποιείται σε ένα μεγάλο ποσοστό των ιστοσελίδων που βρίσκονται σήμερα στο διαδίκτυο. Ο τρόπος με τον οποίο ο Apache εξυπηρετεί αυτές τις αιτήσεις, είναι σύμφωνος με τα πρότυπα που ορίζει το πρωτόκολλο http. Επειδή αποτελεί εφαρμογή ανοιχτού λογισμικού μπορεί να ρυθμιστεί, να επεκταθεί και να προσαρμοστεί με τη συγγραφή κώδικα 'modules' που κάνουν χρήση του apache module API.

Η κοινότητα ανοιχτού λογισμικού που αναπτύσσει σήμερα τον apache web server εξακολουθεί να προσθέτει νέες λειτουργίες, ώστε να παρέχει ο apache μια ασφαλή και επεκτάσιμη πλατφόρμα εξυπηρετητή διαδικτύου. Υπάρχει μια πληθώρα από modules που έχουν αναπτυχθεί για τον apache και τα οποία του δίνουν μια ευρεία γκάμα δυνατοτήτων, όπως user authentication, URL redirection και άλλες. Επιπλέον σημαντικό χαρακτηριστικό του είναι η υποστήριξη εφαρμογών και γλωσσών προγραμματισμού, όπως mySQL και php.

4.2.2. mySQL

Η βάση mySQL είναι ένα από τα πιο δημοφιλή συστήματα σχεσιακών βάσεων δεδομένων ανοιχτού κώδικα, εξαιτίας της υψηλής απόδοσης, μεγάλης αξιοπιστίας και της ευκολίας στη χρήση. Η βάση mySQL μπορεί να τρέξει σε πολλά λειτουργικά συστήματα, συμπεριλαμβανομένων των Linux, Windows, Mac OS, Solaris κλπ. Χρησιμοποιείται επίσης από πολλές μεγάλες εταιρίες σε διαδικτυακές τους εφαρμογές και ιστοσελίδες.

Ανάμεσα στις κύριες δυνατότητες της mySQL λογαριάζονται οι ακόλουθες:

- **Αρχιτεκτονική πελάτη- εξυπηρετητή**: Υπάρχει ο εξυπηρετητής (MySQL) και ένας αυθαίρετος αριθμός πελατών (προγράμματα εφαρμογών), που επικοινωνούν με αυτόν. Δηλαδή στέλνουν ερωτήματα προς τη βάση, αποθηκεύουν αλλαγές κλπ. Τα προγράμματα- πελάτες μπορεί να τρέχουν στο ίδιο, ή σε διαφορετικό μηχάνημα με τον εξυπηρετητή.
- **Συμβατότητα με τη γλώσσα SQL**: Η mySQL χρησιμοποιεί ως γλώσσα της την SQL. Η SQL είναι μια τυποποιημένη γλώσσα για τη διατύπωση ερωτημάτων και την τροποποίηση δεδομένων, αλλά και για τη διαχείριση μιας βάσης δεδομένων.

- **Stored procedures**: Πρόκειται για κώδικα SQL, ο οποίος αποθηκεύεται από το σύστημα. Οι Stored procedures χρησιμοποιούνται συνήθως για να απλοποιήσουν ορισμένα βήματα, όπως την εισαγωγή, ή τη διαγραφή μιας εγγραφής δεδομένων. Αυτό δίνει σε χρήστες- πελάτες της βάσης το πλεονέκτημα ότι δε χρειάζεται να έρθουν απ' ευθείας σε επαφή με τη βάση, ή να τροποποιήσουν άμεσα πίνακές της. Η mySQL υποστηρίζει τις stored procedures από την έκδοση 5.0 και έπειτα.
- **Unicode:** Η MySQL υποστηρίζει όλα τα υπαρκτά σύνολα χαρακτήρων από την έκδοση 4.1 και έπειτα.
- **Ανεξαρτησία πλατφόρμας:** Ο server της mySQL μπορεί να τρέξει σε πολλά διαφορετικά λειτουργικά συστήματα. Τα πιο σημαντικά από αυτά είναι τα Apple Macintosh OS X, Linux, Microsoft Windows, οι διάφορες εκδοχές των Unix, καθώς και το Solaris της Sun.
- **Ταχύτητα:** Η MySQL θεωρείται πολύ γρήγορο πρόγραμμα βάσης δεδομένων, κάτι το οποίο έχουν δείξει πολλά benchmark τεστ.

4.2.3. PHP

Η γλώσσα προγραμματισμού PHP είναι μια ευρέως χρησιμοποιούμενη γλώσσα γενικού σκοπού και ανοιχτού κώδικα, η οποία είναι ιδιαίτερα κατάλληλη για δικτυακές εφαρμογές. Μπορεί επιπλέον να ενσωματωθεί μέσα σε κώδικα HTML. Ο κώδικας της php είναι κώδικας που εκτελείται σε εξυπηρετητή, παράγει κώδικα HTML και στη συνέχεια στέλνεται στον πελάτη- χρήστη. Ο πελάτης λαμβάνει επομένως τα αποτελέσματα της εκτέλεσης του κώδικα ενός αρχείου php χωρίς να γνωρίζει ποιος ήταν αυτός ο κώδικας.

Οι δυνατότητες της php δεν περιορίζονται μόνο στην παραγωγή html. Μπορεί επίσης να παραγάγει εικόνες ή pdf, καθώς και flash videos. Η php μπορεί επίσης να δημιουργεί αυτόματα αρχεία κειμένου διαφόρων τύπων, όπως XHTML, ή γενικότερα κείμενα τύπου XML και να τα αποθηκεύει στο σύστημα αρχείων, ώστε να δημιουργεί μια cache στην πλευρά του εξυπηρετητή για παρουσίαση δυναμικών περιεχομένων. Η γλώσσα επίσης υποστηρίζει πολλά συστήματα βάσεων δεδομένων. Τέλος, έχει μια σειρά από χρήσιμες δυνατότητες που αφορούν την επεξεργασία κειμένου. Σ' αυτές περιλαμβάνονται οι regular expressions της perl, καθώς και πολλές επεκτάσεις και εργαλεία για parsing αρχείων XML.

Ένα πολύ σημαντικό χαρακτηριστικό της γλώσσας είναι ότι μπορεί να χρησιμοποιεί είτε διαδικαστικό, είτε αντικειμενοστραφές μοντέλο προγραμματισμού, είτε μείγμα και των δύο.

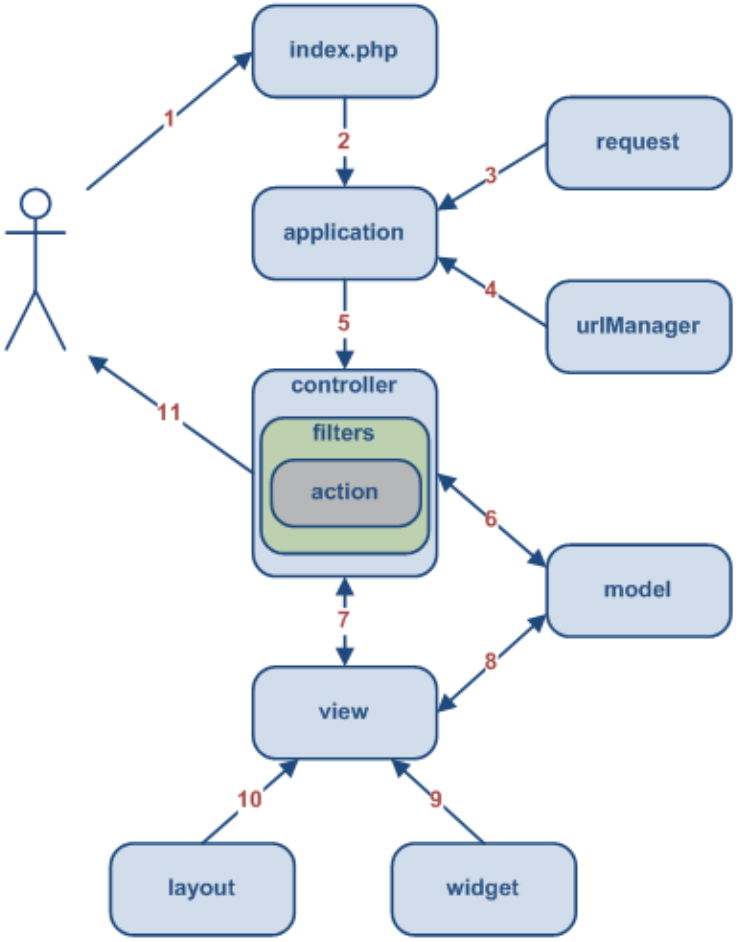
Η php μπορεί να χρησιμοποιηθεί στα κυριότερα λειτουργικά συστήματα, όπως Linux, διάφορες παραλλαγές unix, Windows, Mac OS κ.α. Τέλος υποστηρίζει και τους περισσότερους εξυπηρετητές που χρησιμοποιούνται σήμερα, όπως ο apache, IIS και πολλούς άλλους.

4.2.3.1 Yii framework της php

Η εφαρμογή παρουσίασης γράφων εξέλιξης δεδομένων των micro RNA αποτελεί, όπως αναφέρθηκε, επέκταση του λογισμικού που μπορεί κανείς να χρησιμοποιήσει από την ιστοσελίδα DIANA. Επομένως ακολουθεί και το μοντέλο ανάπτυξης στο οποίο είναι βασισμένη η ιστοσελίδα αυτή. Το μοντέλο αυτό είναι το Yii framework της php.

Το Yii framework υλοποιεί το σχεδιαστικό πρότυπο MVC (Model- View-Controller). Το πρότυπο αυτό χρησιμοποιείται ευρέως σε διαδικτυακές εφαρμογές. Ο σκοπός του είναι να διαχωρίζει τη λογική της εφαρμογής από τη διεπιφάνεια του χρήστη, έτσι ώστε οι αλλαγές που εφαρμόζονται στο ένα τμήμα να μην επηρεάζουν το άλλο. Στο μοντέλο MVC τα δεδομένα αποτελούν το τμήμα model. Το View εμπεριέχει στοιχεία της διεπιφάνειας του χρήστη, ενώ οι ελεγκτές (controllers) είναι υπεύθυνοι για την επικοινωνία των δύο προηγούμενων τμημάτων.

Στο Yii υπάρχει ένας βασικός ελεγκτής (controller) που ονομάζεται application. Αυτός ευθύνεται για την επεξεργασία των αιτήσεων που γίνονται, ώστε να ενεργοποιεί άλλους κατάλληλους ελεγκτές, οι οποίοι εκτελούν τις απαιτούμενες ενέργειες. Μια τυπική ροή εργασιών σε μια εφαρμογή αναπτυγμένη με το framework Yii δίνεται στο ακόλουθο σχήμα:



4.2 Ροή εργασιών εφαρμογής Yii

Τα βήματα της εκτέλεσης των εργασιών συνοψίζονται ως εξής:

1. Όταν ο χρήστης κάνει μια αίτηση στη διεύθυνση όπου βρίσκεται η εφαρμογή Yii, εκτελείται το script με όνομα index.php που βρίσκεται σε αυτή τη τοποθεσία.
2. Δημιουργείται ένας μοναδικός application controller (application).
3. Το application επεξεργάζεται την αίτηση και προσδιορίζει την κατάλληλη ενέργεια (action) του ελεγκτή (controller) που αφορά η αίτηση, με τη βοήθεια ενός συστατικού που ονομάζεται urlManager.
4. Δημιουργείται ο κατάλληλος ελεγκτής και καθορίζεται η ενέργεια που πρέπει να πραγματοποιηθεί.
5. Η ενέργεια διαβάζει το κατάλληλο μοντέλο (δεδομένα).
6. Ενεργοποιείται το κατάλληλο view (διεπιφάνεια χρήστη).
7. Το view διαβάζει τα κατάλληλα δεδομένα από το model.
8. Εκτελούνται ορισμένα widgets, αν υπάρχουν.

9. Εφαρμόζεται κάποιο layout στα views, δηλαδή διαμορφώνεται με ένα κατάλληλο σχήμα η διεπιφάνεια.
10. Τα αποτελέσματα των παραπάνω ενεργειών εμφανίζονται στον χρήστη.

Στη συνέχεια παρουσιάζονται ξεχωριστά τα κύρια στοιχεία μιας εφαρμογής του Yii framework, δηλαδή τα model, view και controller. Περιγράφουμε επιπλέον τις κλάσεις widget, καθώς αποτελούν το κύριο κομμάτι της εφαρμογής που αναπτύχθηκε.

4.2.3.1.1 Model

Οι κλάσεις model αναπαριστούν ένα αντικείμενο δεδομένων, όπως μια φόρμα εισαγωγής δεδομένων, ή μια εγγραφή από βάση δεδομένων. Το Yii framework έχει βασικά δύο ειδών model, τα active records και τα form models.

Ένα form model χρησιμοποιείται συνήθως για να αποθηκεύει δεδομένα που εισάγουν οι χρήστες. Τέτοια δεδομένα συνήθως απορρίπτονται αφού χρησιμοποιηθούν μια φορά, δηλαδή δεν είναι επαναχρησιμοποιούμενα.

Ένα active record είναι ένα σχεδιαστικό πρότυπο που χρησιμοποιείται για την πρόσβαση σε βάσεις δεδομένων με αντικειμενοστραφή τρόπο. Κάθε active record αναπαριστά μια εγγραφή σε μια βάση και τα πεδία μιας τέτοιας εγγραφής αποτελούν ιδιότητες του αντικειμένου.

4.2.3.1.2 View

Τα τμήματα μιας εφαρμογής Yii που αποτελούν το view είναι php scripts, τα οποία σχετίζονται κυρίως με την εμφάνιση της διεπιφάνειας χρήστη. Προτείνεται να μην αλλοιώνουν δεδομένα των model και να έχουν σχετικά απλή μορφή. Εάν η εφαρμογή απαιτεί την εκτέλεση μιας σειράς λογικών πράξεων, αυτές θα πρέπει να πραγματοποιούνται στον controller και όχι στο view. Τα views καλούνται από τους controllers με τους οποίους σχετίζονται, μετά από την εκτέλεση κατάλληλης μεθόδου.

Με τον τρόπο αυτό διαχωρίζεται η εμφάνιση των δεδομένων από την εκτέλεση των πράξεων που απαιτείται κατά την επεξεργασία τους.

4.2.3.1.3 Controller

Οι ελεγκτές δημιουργούνται από το αντικείμενο application όταν γίνονται αιτήσεις από το χρήστη. Ένας ελεγκτής περιέχει μια σειρά μεθόδων που ονομάζονται actions,

οι οποίες όταν εκτελούνται συνήθως επεξεργάζονται τα κατάλληλα δεδομένα που βρίσκονται σε αντικείμενα model και εμφανίζουν κατόπιν το view που απαιτείται. Κάθε controller έχει και μια βασική μέθοδο, η οποία εκτελείται όταν καλείται ο ελεγκτής χωρίς να καθορίζεται συγκεκριμένα κάποιο action από την αίτηση του χρήστη.

4.2.3.1.4 Widget

Ένα Widget αποτελεί ένα συστατικό στοιχείο μιας εφαρμογής που χρησιμοποιείται κυρίως για λόγους παρουσίασης. Τα widget ενσωματώνονται συνήθως σε ένα view και αποτελούν ένα πολύπλοκο, αλλά σχετικά αυτοτελές κομμάτι της διεπιφάνειας χρήστη. Για παράδειγμα μπορεί να χρησιμοποιείται ένα widget για την εμφάνιση ενός ημερολογίου σε μια ιστοσελίδα. Ένα widget επίσης μπορεί να διαθέτει views τα οποία συσχετίζονται με αυτό, όπως συμβαίνει και με τους controllers. Το πλεονέκτημα των widget είναι ότι δίνουν τη δυνατότητα επαναχρησιμοποίησης κώδικα στις διεπιφάνειες χρήστη.

4.2.4. jQuery και τεχνολογία AJAX

Η jQuery είναι μια βιβλιοθήκη της γλώσσας προγραμματισμού javascript. Η javascript είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού, που σχεδιάστηκε κυρίως για να προσθέτει διαδραστικότητα σε ιστοσελίδες και για την ανάπτυξη διαδικτυακών εφαρμογών. Η javascript «τρέχει» στον browser ενός χρήστη μιας ιστοσελίδας, είναι δηλαδή μια client- side γλώσσα.

Η jQuery αποτελεί μια βιβλιοθήκη συναρτήσεων της javascript. Σχεδιάστηκε για να απλοποιεί την διάσχιση εγγράφων HTML, το χειρισμό γεγονότων, την παρουσίαση κίνησης σε ιστοσελίδες, την τροποποίηση του εφαρμοζόμενου στην ιστοσελίδα κώδικα css και τέλος, να διευκολύνει κλίσεις AJAX σε εφαρμογές. Η φιλοσοφία της είναι να δίνει τη δυνατότητα να γράφεται λιγότερος κώδικας για σύνθετες λειτουργίες.

Στα πλεονεκτήματα της χρήσης της βιβλιοθήκης περιλαμβάνονται τα ακόλουθα:

- **Ευκολία χρήσης:** Η βιβλιοθήκη επιτρέπει την πραγματοποίηση σύνθετων λειτουργιών κάνοντας χρήση απλής σύνταξης και περιορισμένων γραμμών κώδικα.

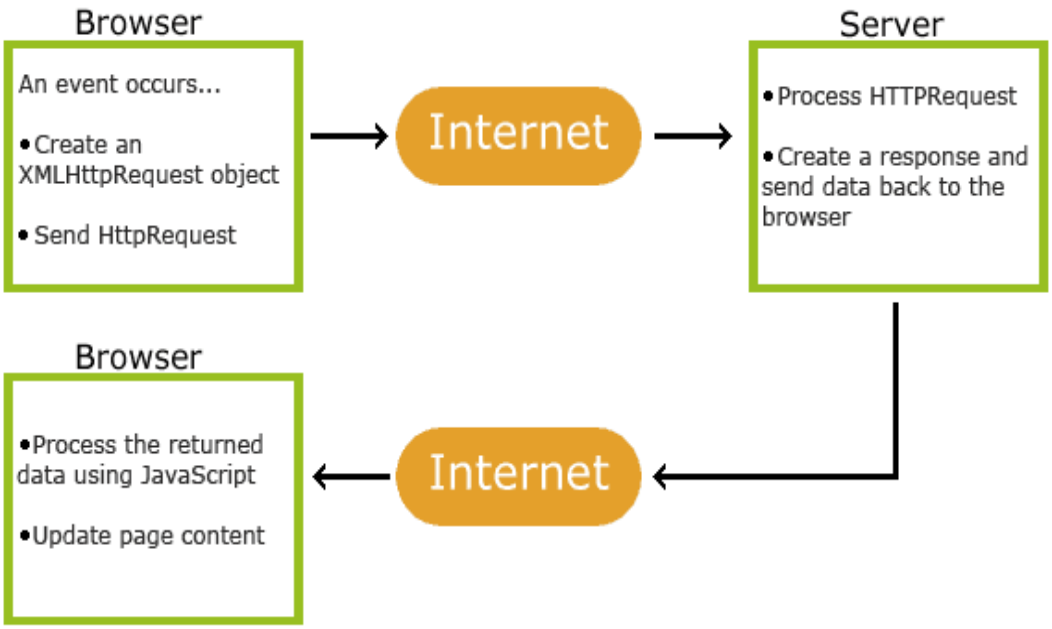
- **Cross- browser συμβατότητα**: Η βιβλιοθήκη φροντίζει οι συναρτήσεις της να λειτουργούν με τον ίδιο τρόπο σε όλους τους browsers.
- **Κοινότητα ανοιχτού λογισμικού:** Αν και η βιβλιοθήκη είναι σχετικά νέα, χρησιμοποιείται ευρέως. Πολλοί χρήστες της την αναβαθμίζουν με τη δημιουργία plug-ins που μπορούν να χρησιμοποιηθούν για να αναβαθμίσουν τις λειτουργίες της.

Ένα από τα βασικά πλεονεκτήματα της jQuery είναι η υποστήριξη λειτουργιών AJAX. Η τεχνολογία AJAX (τα αρχικά σημαίνουν: Asynchronous Javascript and XML) παρέχει έναν τρόπο ανταλλαγής δεδομένων με το server. Στη διαδικασία αυτή δίνεται η δυνατότητα να αναβαθμιστούν κάποια δεδομένα μιας ιστοσελίδας, χωρίς να είναι απαραίτητο να ξαναφορτωθεί εκ νέου ολόκληρη η σελίδα. Η διαδικασία αυτή γίνεται ασύγχρονα.

Ο κύκλος λειτουργίας μιας κλήσης AJAX είναι ο ακόλουθος: Αρχικά πραγματοποιείται ένα γεγονός στον browser του χρήστη. Στη συνέχεια δημιουργείται ένα αντικείμενο XMLHttpRequest, το οποίο στέλνει μια αίτηση (Request) στον εξυπηρετητή μέσω του διαδικτύου. Ο εξυπηρετητής επεξεργάζεται την αίτηση και επιστρέφει δεδομένα μέσω του διαδικτύου στον browser. Ο browser επεξεργάζεται τα δεδομένα που του επιστράφηκαν με χρήση της javascript και αναβαθμίζει το περιεχόμενο της σελίδας. Σχηματικά αυτό φαίνεται στην εικόνα 4.3.

Η εφαρμογή που αναπτύχθηκε πραγματοποιεί μια κλήση AJAX για να παρουσιάσει το γράφο εξέλιξης σε συγκεκριμένο σημείο της ιστοσελίδας.

Η γλώσσα javascript χρησιμοποιήθηκε επιπλέον για τη μορφοποίηση των περιεχομένων των κόμβων των γράφων εξέλιξης και τη σωστή τους τοποθέτηση. Τέλος η javascript ήταν απαραίτητη για την υλοποίηση των αναδυόμενων παράθυρων δεδομένων, καθώς και της λειτουργίας προώθησης αναζήτησης με κλικ του ποντικιού στα αντίστοιχα κουμπιά. Οι λειτουργίες αυτές υλοποιήθηκαν με χρήση δυνατοτήτων που δίνει η βιβλιοθήκη jQuery.



4.3 Λειτουργία κλήσης AJAX

4.3 *Μοντελοποίηση και υλοποίηση εφαρμογής εξελικτικών γράφων*

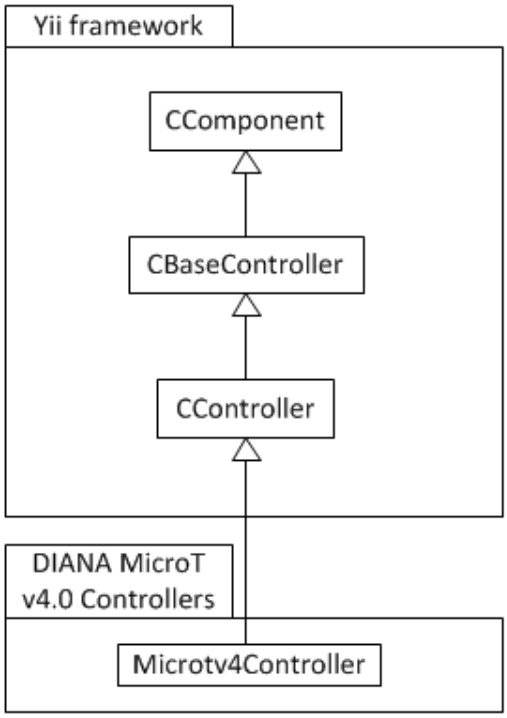
Στην παράγραφο αυτή περιγράφεται συνοπτικά το αρχιτεκτονικό μοντέλο που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής. Αναλύονται επιγραμματικά τα τμήματα από τα οποία αποτελείται, ενώ στη συνέχεια παρουσιάζονται οι συγκεκριμένες κλάσεις και μέθοδοι που αναπτύχθηκαν στα πλαίσια της εργασίας. Τέλος δίνονται επιγραμματικά ορισμένες λεπτομέρειες υλοποίησης, όπως κάποια ερωτήματα επικοινωνίας με τη βάση και ορισμένοι αλγόριθμοι για την κατασκευή του γράφου εξέλιξης.

4.3.1 Περιγραφή κλάσεων

Παρουσιάζεται στη συνέχεια η περιγραφή των κλάσεων, βασισμένων στο Yii framework, οι οποίες αποτελούν την εφαρμογή προβολής γράφων εξέλιξης δεδομένων. Περιγράφονται κλάσεις οι οποίες δημιουργήθηκαν εξολοκλήρου, καθώς και κλάσεις που αποτελούσαν ήδη τμήμα της εφαρμογής DIANA microT v4.0 και οι οποίες τροποποιήθηκαν. Συνεπώς περιοριζόμαστε στην παρουσίαση μόνο δομικών στοιχείων που αφορούν την εφαρμογή.

4.3.1.1 Ελεγκτές

Ένας ελεγκτής είναι υπεύθυνος για την εφαρμογή microT v4.0. Οι κλάσεις από τις οποίες κληρονομεί δίνονται στο ακόλουθο σχήμα.



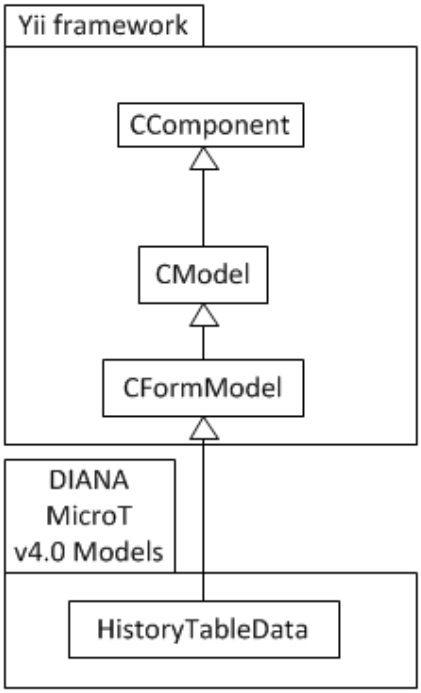
4.4 Controller της εφαρμογής microT v4.0

Ο ελεγκτής αυτός διαθέτει μια σειρά μεθόδων που είναι υπεύθυνες για την επεξεργασία και αναγνώριση των λέξεων κλειδιών που δίνονται στη φόρμα αναζήτησης, για την προβολή των αντίστοιχων αποτελεσμάτων που αφορούν γονίδια και micro RNAs, καθώς και την προβολή προτάσεων για κλειδιά αναζήτησης σε περίπτωση που η αναζήτηση του χρήστη δεν φέρει αποτελέσματα.

Στις μεθόδους που διατίθενται ήδη προστέθηκε η ενέργεια actionGetHistoryGraph. Η μέθοδος αυτή δημιουργεί τα δεδομένα του γράφου εξέλιξης μέσω ενός model με όνομα HistoryTableData. Στη συνέχεια καλεί το κατάλληλο view για την προβολή του γράφου που δημιουργήθηκε.

4.3.1.2 Κλάσεις form model

Μια κλάση μοντέλο προστέθηκε στο σύστημα για τις ανάγκες της εφαρμογής. Η κλάση αυτή ονομάζεται HistoryTableData. Δεν αποτελεί form model με τη συμβατική έννοια, καθώς δεν αναπαριστά τα δεδομένα μιας φόρμας που παρουσιάζεται σε κάποιο view. Οι κλάσεις από τις οποίες κληρονομεί φαίνονται στην εικόνα 4.5.



4.5 Model της εφαρμογής

Η κλάση αποτελεί ένα αντικείμενο που περιέχει διάφορες δομές πινάκων, τις οποίες, μετά από επικοινωνία με τη βάση δεδομένων και κατάλληλη επεξεργασία, θα διαμορφώσει. Οι δομές αυτές χρησιμοποιούνται τελικά από το view που είναι υπεύθυνο για την προβολή του γράφου εξέλιξης. Οι μέθοδοι που περιέχονται στην κλάση HistoryTableData, καθώς και οι λειτουργίες που πρέπει να εκτελούν, παρουσιάζονται ακολούθως:

Μέθοδος getLatestVersion

Η μέθοδος αυτή επικοινωνεί με τη βάση δεδομένων για να αντλήσει την τιμή της τρέχουσας έκδοσης της βάσης mirbase.

Μέθοδος getHistories

Η μέθοδος αυτή επεξεργάζεται την λέξη κλειδί που δόθηκε στην αναζήτηση του χρήστη. Εξάγει το συμπέρασμα αν πρόκειται για accession, ή για ID ενός micro RNA. Στην πρώτη περίπτωση εκτελεί το ερώτημα που αφορά το συγκεκριμένο accession στη βάση και επιστρέφει το αποτέλεσμα. Στη δεύτερη περίπτωση αναζητά όλα τα accessions, τόσο για hairpins, όσο και για mature micro RNAs, τα οποία έχουν σε κάποια στιγμή σχετιστεί με το ID που αναζητήθηκε. Για κάθε ένα από τα accessions αυτά επιστρέφει την καταγεγραμμένη στη βάση ιστορία του.

Μέθοδος getColumnNums

Η μέθοδος αυτή ελέγχει αν έχουμε ιστορικά για hairpins, ή matures, ή και τα δύο. Για κάθε τέτοια περίπτωση επιστρέφει σε ένα πίνακα το σύνολο των αριθμών εκδόσεων που πρέπει να εμφανίζονται στον γράφο εξέλιξης.

Μέθοδος getVersionNumArray

Καλείται από την παραπάνω μέθοδο για κάθε είδος βιομορίου για το οποίο πρόκειται να εμφανιστεί γράφος εξέλιξης και επιστρέφει τους αριθμούς εκδόσεων που θα πρέπει να εμφανίζονται για το συγκεκριμένο είδος βιομορίου.

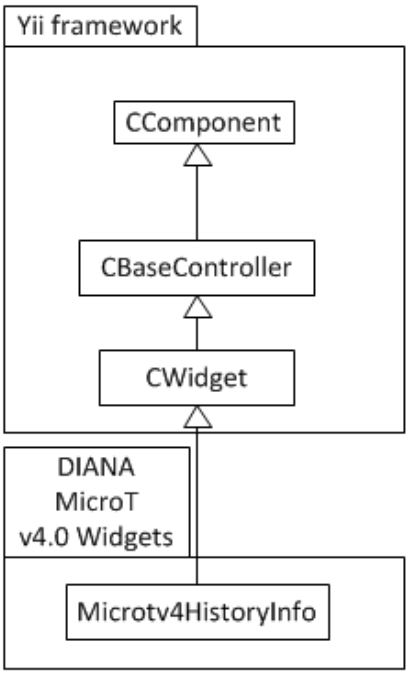
Μέθοδος getTableParts

Η μέθοδος αυτή μετράει πόσα hairpins σχετίζονται με ένα ID, ή accession στο οποίο γίνεται αναζήτηση. Επίσης μετράει τον αριθμό των mature, αλλά και των hairpins που τα παράγουν, τα οποία σχετίζονται με ένα mature accession, ή ID το οποίο αναζητήθηκε.

Μέθοδος makeRowData

Η μέθοδος αυτή είναι υπεύθυνη για τη δημιουργία μιας δομής πίνακα που προκύπτει από την ανάγνωση των δεδομένων που επιστρέφονται από τη βάση και η οποία θα μπορεί να διαβάζεται από ένα view. Ο πίνακας αυτός θα πρέπει να έχει για κάθε κελί του περιεχόμενα που διαμορφώνονται με ενιαία μορφή. Αυτό είναι απαραίτητο ώστε ένα view να μπορεί να αντλεί από κάθε κελί του πίνακα ακριβώς με τον ίδιο τρόπο δεδομένα και να τα προβάλλει.

4.3.1.3 Widget εφαρμογής



4.6 Widget εφαρμογής

Το widget της εφαρμογής αποτελεί τη διεπιφάνεια με την οποία έρχεται σε επαφή ο χρήστης. Καλείται σε views που αφορούν το αποτέλεσμα της αναζήτησης ενός hairpin, ή ενός mature, είτε αυτή έφερε αποτελέσματα, είτε ο χρήστης παραπέμπεται στη χρήση άλλων λέξεων- κλειδιών στην αναζήτησή του.

Αποτελείται από μια μπάρα στην οποία εμφανίζονται οι λέξεις- κλειδιά που αναζητήθηκαν. Δίπλα από κάθε λέξη εμφανίζεται ένα κουμπί, στο πάτημα του οποίου θα εμφανιστούν κάτω από τη μπάρα οι γράφοι εξέλιξης της αντίστοιχης λέξης. Αν ξαναπατηθεί το κουμπί, οι γράφοι εξέλιξης αποκρύπτονται πάλι.

4.3.2 Λεπτομέρειες υλοποίησης

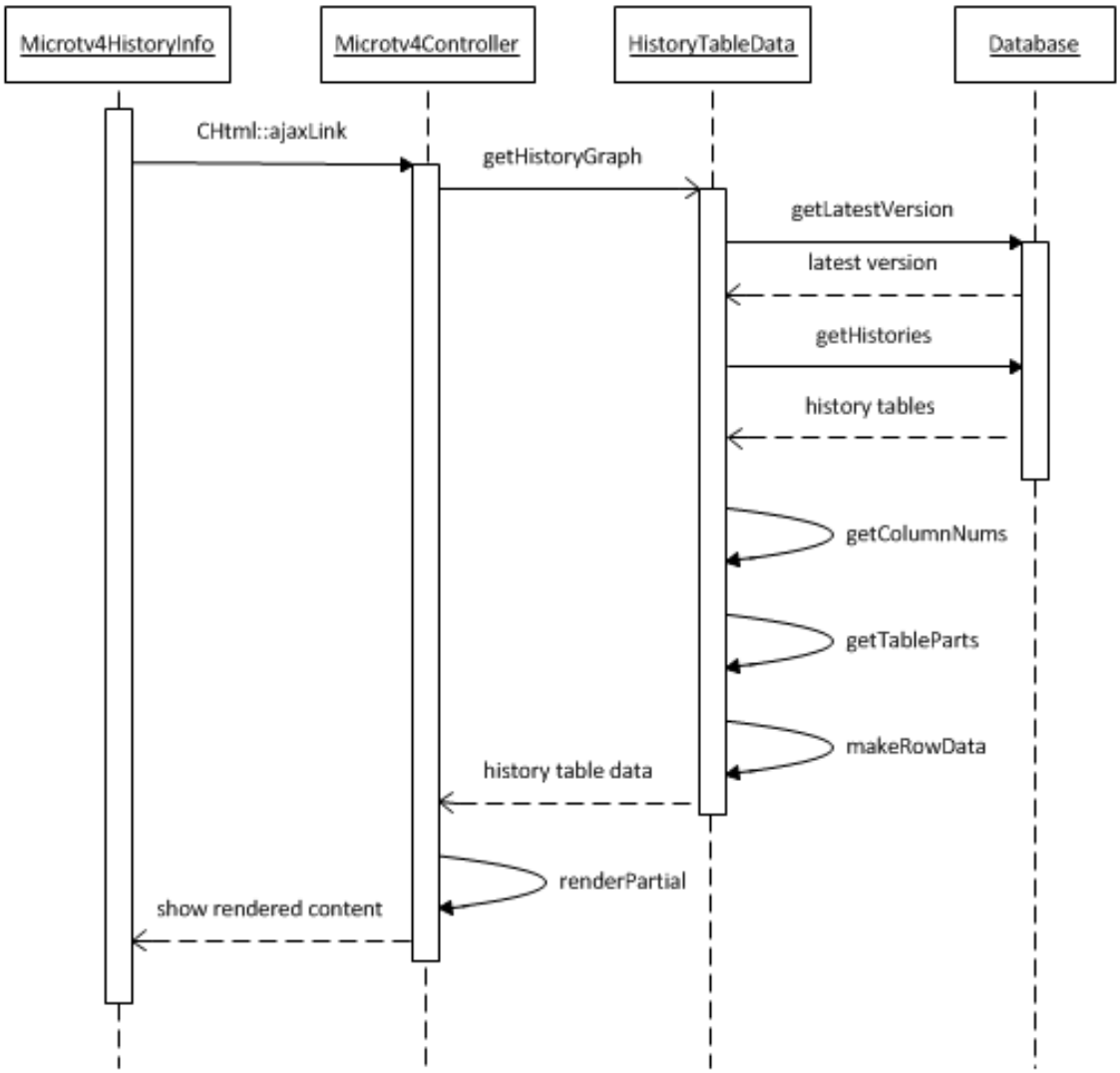
Η εφαρμογή προβολής των γράφων εξέλιξης δεδομένων αποτελείται από ένα widget του Yii framework. Η λειτουργία του στηρίζεται στη βάση δεδομένων που συμπυκνώνει την πληροφορία για την εξέλιξη των δεδομένων κάθε καταγεγραμμένης οντότητας της mirbase και της οποίας η κατασκευή περιγράφτηκε στο κεφάλαιο 3. Το widget αποτελεί τμήμα ενός view και παρουσιάζει όλες τις λέξεις που αναζητήθηκαν από το χρήστη και αντιστοιχούν σε micro RNAs. Κάθε τέτοια λέξη συνοδεύεται από

ένα κουμπί για την προβολή του γράφου εξέλιξης που της αντιστοιχεί. Για την προβολή του γράφου απαιτείται η επικοινωνία μεταξύ των διάφορων στοιχείων που απαρτίζουν το σύστημα: αρχικά γίνεται μια κλήση AJAX, η οποία ενεργοποιεί την ενέργεια getHistoryGraph του ελεγκτή της εφαρμογής microT v4.0. Ο ελεγκτής αυτός αρχικοποιεί ένα αντικείμενο model, τύπου HistoryTableData. Το μοντέλο HistoryTableData επικοινωνεί με τη βάση δεδομένων από την οποία αντλεί την απαιτούμενη πληροφορία. Στη συνέχεια διαμορφώνει τα δεδομένα που έχει αντλήσει με τέτοιο τρόπο ώστε να μπορούν να διαβαστούν με τη μορφή πίνακα από το view που θα τα παρουσιάσει. Η μορφή του πίνακα πρέπει να είναι τέτοια, ώστε στο view να μην απαιτούνται επιπλέον λογικές πράξεις, ενώ όλα τα περιεχόμενα κελιών έχουν παρόμοια δομή, ώστε να διαβάζονται με ενιαίο τρόπο. Επιπλέον το μοντέλο HistoryTableData αντλεί δεδομένα που αφορούν τις εκδόσεις που πρέπει να υπάρχουν ως επικεφαλίδες στο γράφο και την τιμή της τρέχουσας έκδοσης. Τα δεδομένα αυτά γίνονται διαθέσιμα στον αρμόδιο ελεγκτή, ο οποίος τα περνάει στο view script που ευθύνεται για την προβολή του γράφου.

Στην εικόνα 4.7 δίνεται συνοπτικά το ακολουθιακό διάγραμμα της εφαρμογής, όπου παρουσιάζεται η σειρά εκτέλεσης ενεργειών και η επικοινωνία μεταξύ των διαφορετικών τμημάτων που απαρτίζουν την εφαρμογή.

4.3.2.1 Επικοινωνία με τη βάση και καθορισμός του είδους της παραμέτρου αναζήτησης

Κατά την επικοινωνία της εφαρμογής με τη βάση δεδομένων ενδιαφέρουν τα εξής στοιχεία: Αρχικά η τρέχουσα έκδοση, δηλαδή η πιο πρόσφατη έκδοση που έχει κυκλοφορήσει για τη βάση mirbase. Επιπλέον, ενδιαφέρει το σύνολο του ιστορικού ενός hairpin, ή mature, εφόσον έχει γίνει αναζήτηση με βάσει κάποιο accession. Διαφορετικά, ενδιαφέρει το ιστορικό όλων των hairpins **και** mature που φέρουν σε κάποια έκδοση ένα συγκεκριμένο ID, εφόσον η αναζήτηση που έγινε αφορά ID.



4.7 Ακολουθιακό διάγραμμα εφαρμογής

Το πρώτο από τα δύο στοιχεία εξασφαλίζεται με ένα ερώτημα προς τη βάση που αρχικά επιστρέφει όλες τις εγγραφές που στο πεδίο last_appearance έχουν τη μέγιστη τιμή που συναντάται για αυτό το πεδίο στη βάση. Στη συνέχεια, επειδή απαιτείται μόνο μια τέτοια εγγραφή εφαρμόζουμε στο παραπάνω αποτέλεσμα μια συνάρτηση συνάθροισης. Το πεδίο που ενδιαφέρει να αντλήσουμε είναι το actual_last_appearance. Συγκεκριμένα το ερώτημα που χρησιμοποιήθηκε αφορά τον πίνακα hairpins και δίνεται παρακάτω:

```
SELECT * , COUNT( * )
FROM
 (
   SELECT AC, MAX( last_appearance ) , actual_last_appearance, last_appearance
   FROM maturehistory
   GROUP BY AC
   HAVING last_appearance = MAX( last_appearance )
 ) AS t1
```

Η συνθήκη max(last_appearance) = last_appearance μας εξασφαλίζει ότι θα πάρουμε εγγραφές με τιμή τον αριθμό που αντιστοιχεί στην τρέχουσα έκδοση της βάσης mirbase. Η απαίτηση αυτή δε μπορεί να γίνει στο πεδίο actual_last_appearance, διότι το περιεχόμενό του είναι τύπου VARCHAR. Έτσι οι μέγιστες τιμές των αριθμών που αναπαρίστανται δεν αντιστοιχούν στις τιμές που θα επέστρεφε η συνάρτηση max για το πεδίο αυτό.

Για την άντληση του ιστορικού ενός micro RNA από τη βάση, πρέπει πρώτα να ελεγχθεί αν πρόκειται για hairpin, ή mature. Για τον έλεγχο αυτό αρκεί να ελεγχθεί αρχικά, αν η λέξη που αποτελεί το κλειδί αναζήτησης αρχίζει με τους χαρακτήρες "MIMAT", ή "MI". Στις περιπτώσεις αυτές είναι δεδομένο ότι αναζητείται ένα συγκεκριμένο accession και άρα το ιστορικό μιας συγκεκριμένης εγγραφής. Τα ερωτήματα που διατυπώνονται σε αυτή τη περίπτωση είναι τα ακόλουθα:

```
SELECT AC, ID, first_appearance, actual_first_appearance, last_appearance,
        actual_last_appearance, hairpinhistory.Change, DE, SQ, forward, comment
FROM  hairpinhistory
WHERE AC = '" . mysql_escape_string($search_parameter) . "'
ORDER BY AC, first_appearance ASC
```

```
SELECT AC, ID, mother_hairpin, first_appearance, actual_first_appearance,
        last_appearance, actual_last_appearance, maturehistory.Change, SQ, sequence_part
FROM maturehistory
WHERE AC = '" . mysql_escape_string($search_parameter) . "'
GROUP BY AC, mother_hairpin, first_appearance, maturehistory.Change
ORDER BY AC, mother_hairpin, first_appearance ASC
```

Το πρώτο από τα παραπάνω αντιστοιχεί στην περίπτωση των hairpins. Εδώ αρκεί η ταξινόμηση βάσει της τιμής του πεδίου first_appearance για να πάρουμε τα αποτελέσματα σε σειρά αυξανόμενων εκδόσεων. Στη δεύτερη περίπτωση το ερώτημα διατυπώνεται για τα mature micro RNAs και η ταξινόμηση πρέπει επίσης να γίνεται βάσει του hairpin που παράγει το mature. Σε κάθε περίπτωση η μεταβλητή $search_parameter αποτελεί το συγκεκριμένο accession που αναζητήθηκε υπό τη μορφή συμβολοσειράς.

Στην περίπτωση που η αναζήτηση που γίνεται δεν αφορά κάποιο accession, αλλά κάποιο ID, θα πρέπει να προηγηθούν δύο ερωτήματα προς τη βάση, προτού λάβουμε το ιστορικό όλων των εγγραφών που ενδιαφέρουν. Τα ερωτήματα αυτά αφορούν την εύρεση όλων των accession που σχετίζονται με το ID που αναζητήθηκε. Επομένως θα πρέπει να γίνει ένα ερώτημα που αφορά τον πίνακα hairpinhistory και ένα που αφορά τον πίνακα maturehistory. Αυτό επιβάλλεται διότι ενδέχεται ένα ID να συναντάται

και στις δύο κατηγορίες βιομορίων. Τα ερωτήματα που βρίσκουν τα accessions που ενδιαφέρουν είναι τα ακόλουθα:

```
SELECT AC
FROM hairpinhistory
WHERE ID = '" . mysql_escape_string($search_parameter) . "'
GROUP BY AC
```

```
SELECT AC, mother_hairpin
FROM maturehistory
WHERE ID = '" . mysql_escape_string($search_parameter) . "'
GROUP BY AC, mother_hairpin
```

Όπως και παραπάνω, η μεταβλητή $search_parameter είναι η λέξη που αποτελεί το κλειδί αναζήτησης, στην περίπτωση αυτή κάποιο ID. Στην περίπτωση των matures η ομαδοποίηση γίνεται τόσο με βάση το accession, όσο και με βάση το hairpin από το οποίο παράγεται. Αυτό είναι απαραίτητο γιατί ένα mature μπορεί να παράγεται από περισσότερα από ένα hairpins.

Στη συνέχεια πρέπει να δημιουργηθεί μια συμβολοσειρά που θα χρησιμοποιηθεί στο τελικό ερώτημα προς τη βάση. Η συμβολοσειρά πρέπει να περιέχει όλα τα accessions που βρέθηκαν προηγουμένως, συζευγμένα με τη λέξη OR. Τα ερωτήματα που θα διατυπωθούν τελικά δίνονται παρακάτω:

```
SELECT AC, ID, first_appearance, actual_first_appearance, last_appearance,
        actual_last_appearance, hairpinhistory.Change, DE, SQ, comment, forward
FROM hairpinhistory
WHERE " . $hairpin_accession_search . "
ORDER BY AC, first_appearance ASC
```

Το ερώτημα αυτό αφορά τα hairpins. Η μεταβλητή $hairpin_accession_search αποτελείται από μια σειρά εκφράσεων "AC = 'τιμή' " συζευγμένες με τη λέξη OR. Η απαίτηση να ταξινομηθούν τα αποτελέσματα με βάση το accession και την έκδοση, θα μας δώσει όλο το ιστορικό όλων των accessions που ενδιαφέρουν ομαδοποιημένο κατά accession και με σειρά εμφάνισης.

Αντίστοιχα το ερώτημα για τα mature είναι το ακόλουθο:

```
SELECT AC, ID, mother_hairpin, first_appearance, actual_first_appearance, last_appearance,
        actual_last_appearance, maturehistory.Change, SQ, sequence_part
FROM maturehistory
WHERE " . $mirna_accession_search . "
GROUP BY AC, mother_hairpin, first_appearance, maturehistory.Change
ORDER BY AC, mother_hairpin, first_appearance ASC
```

Σε αυτήν την περίπτωση η μεταβλητή $mirna_accession_search αποτελεί μια σειρά εκφράσεων όπως και αυτή που χρησιμοποιείται στο ερώτημα για τα hairpins. Με αυτόν τον τρόπο βέβαια παρουσιάζεται η ιστορία του mature από όλα τα hairpin που το παράγουν, ανεξάρτητα αν είχε σε όλες τις περιπτώσεις κάποια στιγμή το συγκεκριμένο ID. Αυτή η σχεδιαστική επιλογή προτιμήθηκε ώστε να παρουσιάζονται πλήρη τα ιστορικά ενός mature κατά την αναζήτηση.

Σε περίπτωση που θέλαμε να εμφανίζεται η ιστορία ενός mature μόνο με βάση τα hairpin που σε κάποια στιγμή το παράγουν με το ζητούμενο όνομα, θα έπρεπε οι εκφράσεις στη συνθήκη WHERE να τροποποιηθούν. Οι μορφή που θα έπρεπε να πάρουν θα ήταν "(AC = 'τιμή' AND mother_hairpin = 'τιμή')", συζευγμένες με τη λέξη OR.

4.3.2.2 Προβολή γράφων εξέλιξης δεδομένων που αφορούν την παράμετρο αναζήτησης

Αφού σταλούν τα κατάλληλα ερωτήματα στη βάση και επιστραφούν τα σύνολα αποτελεσμάτων, τα οποία αφορούν το ιστορικό των εγγραφών που ενδιαφέρουν, θα πρέπει να γίνει μια επεξεργασία των συνόλων αυτών, ώστε να είναι σε μορφή που μπορεί άμεσα να προβληθεί. Το σύνολο αποτελεσμάτων δίνει την ιστορία ενός βιομορίου εκφρασμένη σε εγγραφές της μορφής της βάσης που περιγράφθηκε στο κεφάλαιο 3. Το σύνολο αυτό θα πρέπει να διαβαστεί και να μετατραπεί σε πίνακα που μπορεί να προβληθεί άμεσα με μια ανάγνωση από το κατάλληλο view.

Ο πίνακας που θα προκύψει θα αποτελείται από μια σειρά για κάθε accession που αφορά την αναζήτησή μας. Κάθε σειρά θα πρέπει να αποτελείται από ζευγάρια κελιών. Αυτά σε μια περίπτωση μπορεί να είναι και τα δύο κενά. Διαφορετικά, θα αποτελούνται εναλλάξ από ένα πεδίο που θα αναγράφει το είδος της παρατηρούμενης αλλαγής, το οποίο θα αντιστοιχεί σε ακμή του γράφου και ένα δεύτερο, που θα περιέχει το ID που διαβάζουμε και το οποίο θα αντιστοιχεί σε κόμβο του γράφου που θα προβληθεί. Εξαίρεση αποτελεί το πρώτο κελί της σειράς, το οποίο θα περιέχει είτε ένα ID, είτε θα είναι κενό. Δεν θα βρίσκεται σε ζεύγος. Όλες οι σειρές θα έχουν σταθερό μήκος. Για να υπάρχει σωστή στοίχιση ενδέχεται κάποια αρχικά κελιά μιας σειράς να πρέπει να παραμείνουν άδεια. Ένα τέτοιο παράδειγμα είναι η περίπτωση όπου γίνεται αναζήτηση με βάση κάποιο ID και προκύπτουν δύο hairpins (δηλαδή δύο accessions) εκ των οποίων το ένα πρωτοεμφανίζεται σε μεταγενέστερη έκδοση

από το άλλο. Επιπλέον σε περίπτωση διαγραφής ενός από τα δύο hairpins πριν από το άλλο, θα πρέπει να είναι και τα τελευταία κελιά της σειράς του πίνακα κενά.

Για να γίνουν τα παραπάνω κατανοητά, αναφέρουμε ως παράδειγμα την περίπτωση αναζήτησης του ID ‘dre-miR-430b-5’. Το ID αυτό αφορά μόνο μόρια τύπου hairpin. Παρουσιάζουμε το σύνολο των αποτελεσμάτων (τα απαραίτητα πεδία) που παίρνουμε με το αντίστοιχο ερώτημα:

AC	ID	actual_first_appearance	actual_last_appearance	Change
MI0002145	dre-mir-430b-5	7.0	8.1	NEW
MI0002145	NULL	8.2	NULL	FORWARD
MI0002166	dre-mir-430b-26	7.0	8.1	NEW
MI0002166	dre-mir-430b-5	8.2	12.0	NAME SEQUENCE
MI0002166	dre-mir-430b-5	13.0	18.0	SEQUENCE

Θέλουμε να μετασχηματίσουμε το παραπάνω σύνολο αποτελεσμάτων. Αυτό θα γίνει σε δύο φάσεις. Πρώτα σε έναν πίνακα της μορφής:

dre-mir-430b-5	No Change	dre-mir-430b-5	FORWARD	MI0001528				
dre-mir-430b-26			NAME SEQUENCE	dre-mir-430b-5	SEQUENCE	dre-mir-430b-5	No Change	dre-mir-430b-5

Στον παραπάνω πίνακα η πρώτη στήλη αντιστοιχεί στην έκδοση 7.0 της βάσης mirbase, η τρίτη στήλη αντιστοιχεί στην έκδοση 8.1 της βάσης mirbase, η πέμπτη στήλη αντιστοιχεί στην έκδοση 8.2 της βάσης, η έβδομη στήλη στην έκδοση 13.0 και η τελευταία στήλη στην έκδοση 18.0. Οι εκδόσεις αυτές που είναι απαραίτητες να απεικονίζονται έχουν αντληθεί από το σύνολο αποτελεσμάτων που επιστράφηκε από το ερώτημα στη βάση.

Ο αλγόριθμος που ακολουθούμε δίνεται με την μορφή ψευδοκώδικα ως εξής:

```
Read result set
    For each entry of result set
        If accession changes
            Create a new line in result array
        If recorded change is DELETE or FORWARD
            Write the type of change into a cell of the array line
            Store recorded data into next cell or array line
            Fill remaining cells of array line with empty values
        Else
            Fill empty cells as required
            Store recorded change into next cell of array line
            Store recorded micro RNA data into next cell of array line
```

Στη συνέχεια θέλουμε να το μετατρέψουμε σε έναν πίνακα της μορφής:

dre-mir-430b-5	No Change	dre-mir-430b-5	FORWARD	MI0001528				
dre-mir-430b-26	NAME SEQUENCE			dre-mir-430b-5	SEQUENCE	dre-mir-430b-5	No Change	dre-mir-430b-5

Συγχωνεύονται δηλαδή τα κελιά του πίνακα που είναι κενά με αυτά που ως περιεχόμενο αναγράφουν κάποιο είδος αλλαγής. Επιπλέον συγχωνεύονται όλα τα κενά κελιά στο τέλος μιας σειράς. Επειδή ο πίνακας θα πρέπει να εμπεριέχει όλα τα δεδομένα σε μορφή html, η οποία θα πρέπει να παρουσιαστεί από το κατάλληλο view script, θα μετασχηματίσουμε κάθε κελί του παραπάνω πίνακα σε έναν νέο πίνακα. Ο πίνακας αυτός θα περιέχει πληροφορίες για τον αριθμό στηλών που θα πρέπει να έχει έκταση το κελί, πληροφορία για την εικόνα που θα πρέπει να βρίσκεται στο κελί (βέλος- ακμή στην περίπτωση αλλαγής, κόμβος γράφου σε περίπτωση παρουσίασης δεδομένων, άδεια κελιά στην περίπτωση κενών στο τέλος μιας γραμμής), καθώς και τις λεπτομερείς πληροφορίες που πρέπει να εμφανίζονται σε περίπτωση που θέλουμε να δούμε όλα τα δεδομένα που καταγράφονται για κάποια έκδοση.

Ο αλγόριθμος που πραγματοποιεί την διαδικασία αυτή είναι ο παρακάτω:

```
Read unrefined graph array
    For each entry of array
        If it is empty
            Count number of empty fields
        If there is a recorded change type
            Create an array for this cell with fields for
            a) the number of columns it spans
            b) the type of recorded change
            c) an image for a graph edge
        Else
            Create an array fir this cell with fields for
            a) the number of columns it spans
            b) the recorded ID
            c) the image for a graph node
            d) detailed micro RNA data to be displayed on demand
```

Σε κάθε περίπτωση οι πίνακες που εισάγονται στα κελιά έχουν τα ίδια πεδία. Τα πεδία αυτά στην περίπτωση που δε χρειάζονται γεμίζονται με το κενό string.

Στην περίπτωση των mature ακολουθείται παρόμοια διαδικασία. Η μόνη διαφοροποίηση αφορά την παραγωγή του πρώτου πίνακα που προκύπτει από το αποτέλεσμα που επιστρέφει η βάση. Στην περίπτωση αυτή πρέπει να δημιουργείται μια γραμμή του πίνακα για κάθε hairpin που παράγει το mature (το ίδιο mature accession και διαφορετικό hairpin accession) και όχι μόνο για κάθε καινούριο mature (διαφορετικό mature accession) που ενδιαφέρει.

Επιπλέον υπάρχουν διακριτές εγγραφές στον πίνακα maturehistory που αφορούν την πρώτη εμφάνιση ενός mature accession γενικά (εγγραφές NEW). Επίσης έχουμε και εγγραφές που αφορούν τη συνολική διαγραφή του (εγγραφές DELETE). Οι εγγραφές του πρώτου τύπου πρέπει να αγνοούνται, ενώ οι εγγραφές του δεύτερου τύπου πρέπει να χρησιμοποιούνται για να καταγράφεται η έκδοση στην οποία γίνεται η διαγραφή. Το τελευταίο επιβάλλεται, ώστε να μπορούμε να διακρίνουμε μια τελική διαγραφή

ενός mature accession από την περίπτωση που παύει η καταγραφή της παραγωγής του mature μόνο για συγκεκριμένα hairpins.

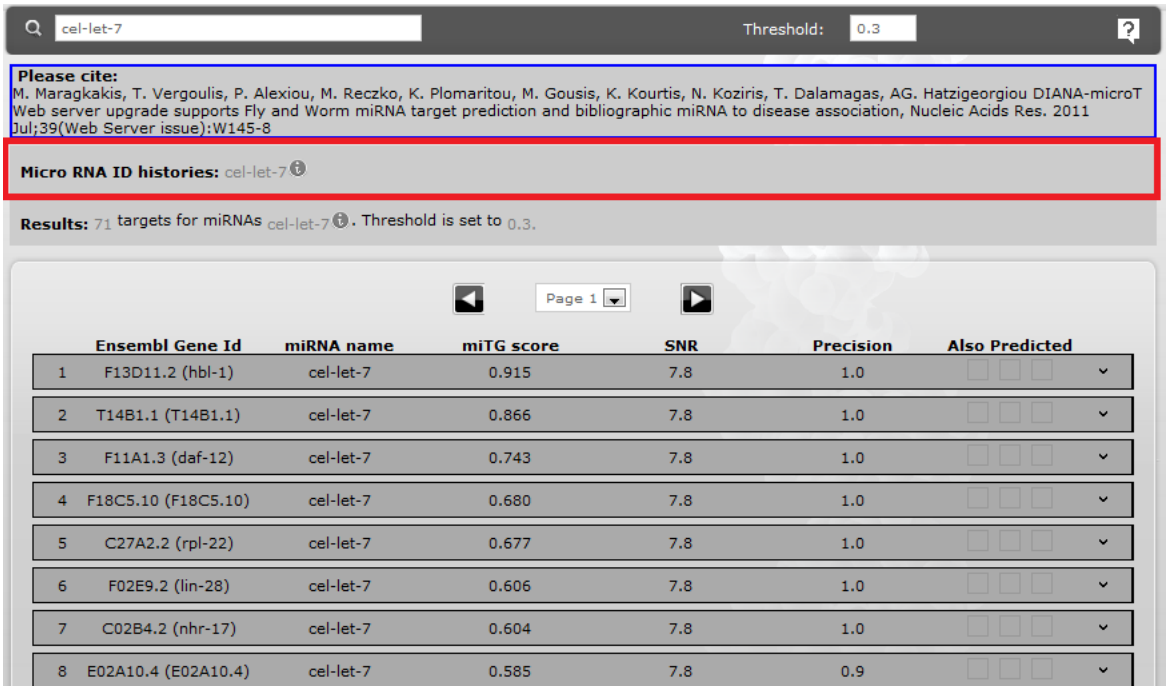
Συνολικά δηλαδή ο πρώτος αλγόριθμος τροποποιείται στο παρακάτω:

```
Read result set
    For each entry of result set
        If accession changes OR mother_hairpin changes
            create a new line in result array
        If recorded change is DELETE
            Store the version the DELETE is happening
        If recorded change is REMOVE PARENT HAIRPIN
            Write the type of change into a cell of the array line
            Store recorded data into next cell or array line
            Fill remaining cells of array line with empty values
        Else
            Fill empty cells as required
            Store recorded change into next cell of array line
            Store recorded micro RNA data into next cell of array line
```

Η κατασκευή του δεύτερου πίνακα γίνεται κατά τον ίδιο τρόπο όπως και για τα hairpins.

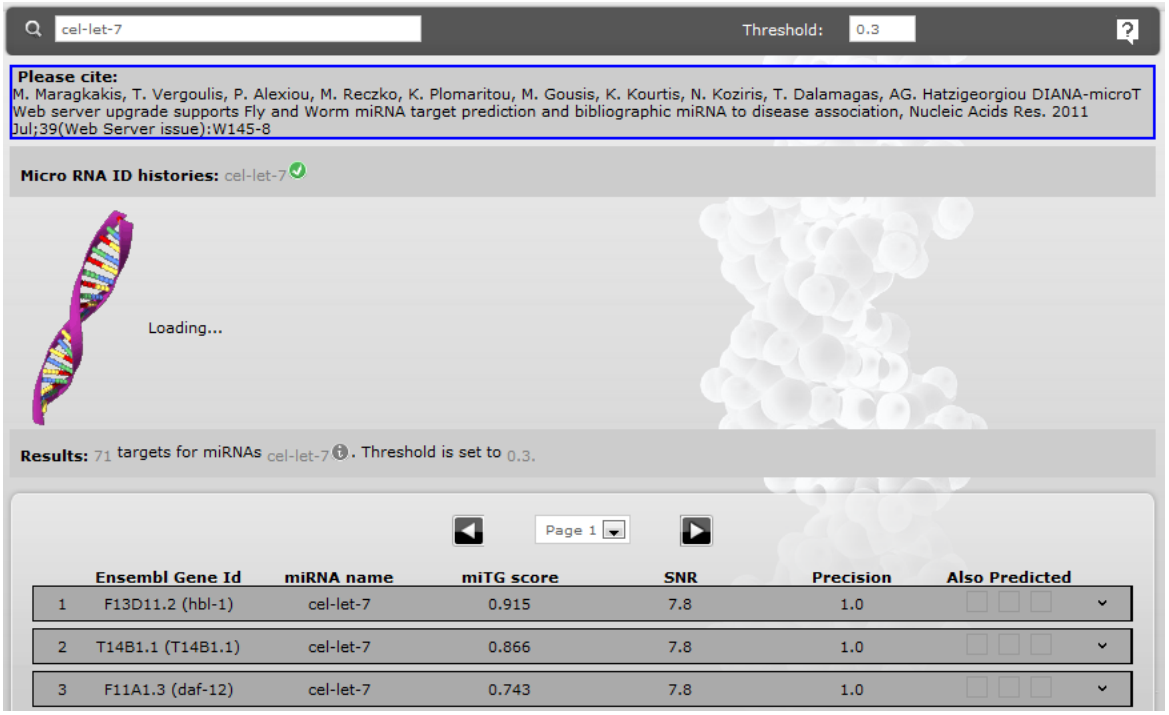
4.4 Παρουσίαση εφαρμογής

Στη συνέχεια δίνουμε παραδείγματα χρήσης της εφαρμογής και ερμηνεύουμε τα αποτελέσματα που μας δίνει. Στην εικόνα 4.8 δίνουμε την διεπιφάνεια χρήστη της εφαρμογής DIANA microT v4.0, στην οποία έχει προστεθεί το widget που παρουσιάζει τους γράφους εξέλιξης. Με κόκκινο χρώμα έχουμε σημειώσει το τμήμα της διεπιφάνειας που αποτελεί το widget καθεαυτό.



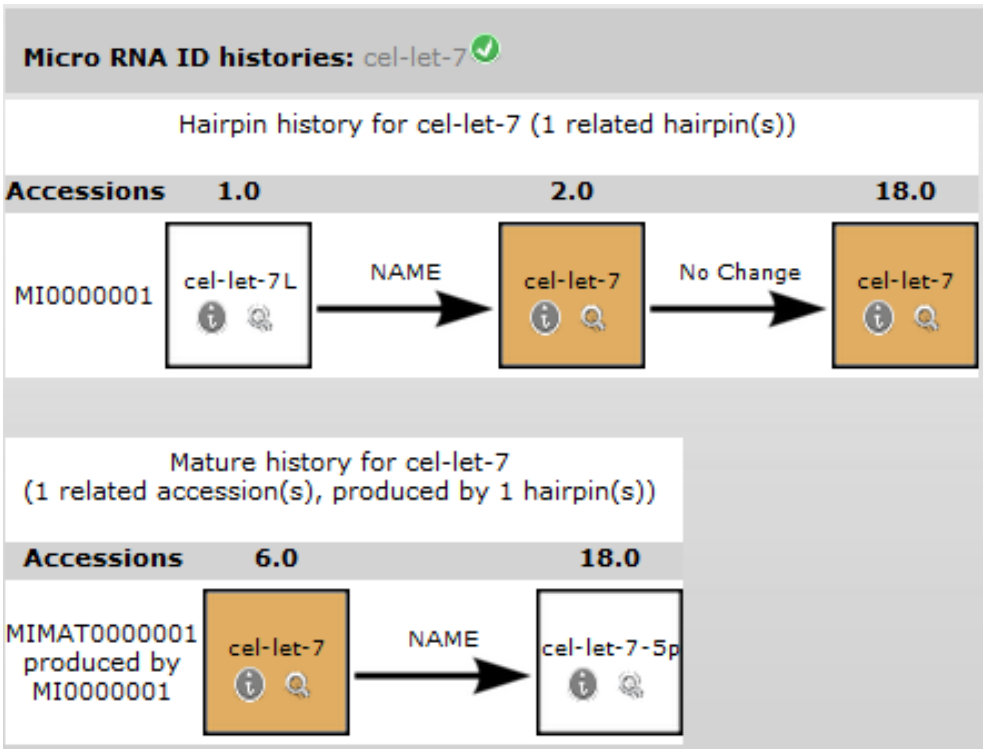
4.8 Διεπιφάνεια εφαρμογής microT v4.0

Στην εικόνα που φαίνεται έχει προηγηθεί αναζήτηση με λέξη-κλειδί το ID ‘cel-let-7’. Για την εμφάνιση του γράφου εξέλιξης δεδομένων ο χρήστης πρέπει να πατήσει με το ποντίκι το κουμπί με την ένδειξη “*i*”, το οποίο βρίσκεται δίπλα από τη λέξη αυτή. Με το πάτημα αυτού του κουμπιού, η εφαρμογή ξεκινά την επικοινωνία με τη βάση και το φόρτωμα του γράφου. Μέχρι να ολοκληρωθεί η διαδικασία αυτή έχουμε την ακόλουθη εικόνα αναμονής:



4.9 Loading screen εφαρμογής

Όταν ολοκληρώσει η κλήση AJAX που πραγματοποιείται κατά την αναμονή, παίρνουμε τα αποτελέσματα της αναζήτησης. Τα αποτελέσματα αυτά φαίνονται στην εικόνα 4.10:



4.10 Αποτελέσματα αναζήτησης για το ID "cel-let-7"

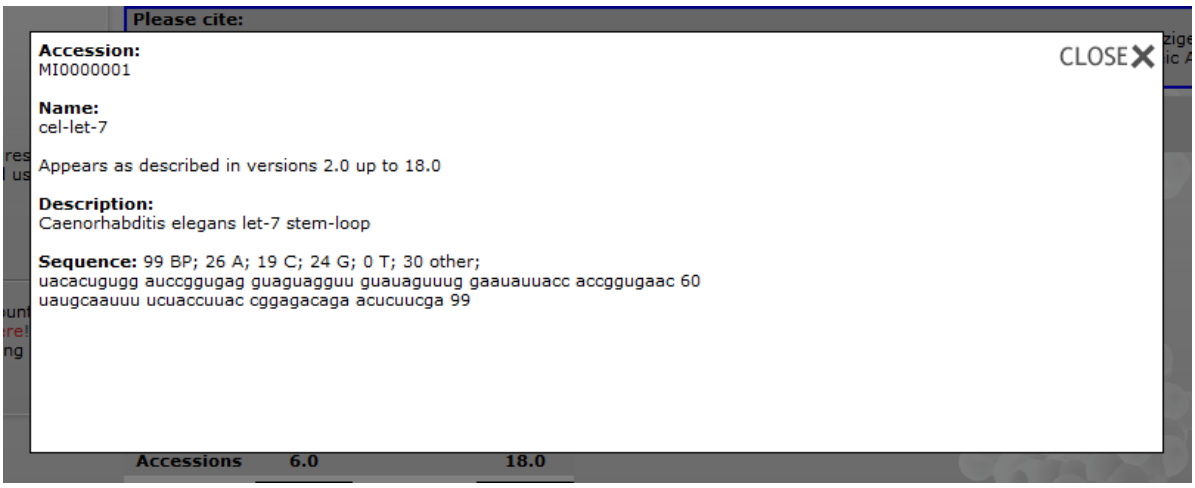
Τα αποτελέσματα αυτά μας δίνουν την ακόλουθη πληροφορία: Το ID 'cel-let-7' σχετίζεται με ένα hairpin και με ένα mature micro RNA. Για το λόγο αυτό έχουμε ως αποτέλεσμα την εμφάνιση δύο γράφων. Οι κόμβοι του γράφου που αφορούν το ID που αναζητήθηκε χρωματίζονται ξεχωριστά. Οι αριθμοί που αντιστοιχούν στις εκδόσεις όπου παρατηρούνται αλλαγές λειτουργούν και ως σύνδεσμοι προς την αντίστοιχη σελίδα ftp της mirbase, όπου μπορεί κανείς να βρει τα αρχεία .dat των αντίστοιχων εκδόσεων.

Ο πρώτος γράφος δίνει την ιστορία που αφορά το όνομα για τα hairpins. Υπάρχει μια εγγραφή που σχετίζεται με αυτό το όνομα, η εγγραφή με accession MI0000001. Η εγγραφή φέρει όνομα που αναζητήθηκε από την έκδοση 2.0 μέχρι την έκδοση 18.0. Η συνολική ιστορία του μορίου με accession MI0000001 είναι η εξής: Εμφανίζεται στην έκδοση 1.0 με ID 'cel-let-7L'. Διατηρεί αυτό το όνομα ως την έκδοση 2.0, όπου

έχουμε αλλαγή του ID στο ‘cel-let-7’. Το όνομα αυτό διατηρείται μέχρι την έκδοση 18.0 (τρέχουσα έκδοση).

Ο δεύτερος γράφος δίνει την ιστορία που αφορά το όνομα που αναζητήθηκε για τα matures. Υπάρχει ένα accession που σχετίζεται με το όνομα αυτό (MIMAT0000001) και ένα μόνο hairpin που παράγει το accession αυτό (MI0000001) . Το όνομα που αναζητήθηκε αφορούσε το mature με accession MIMAT0000001 από την έκδοση 6.0 (υπενθυμίζουμε ότι είναι η πρώτη έκδοση για την οποία διαθέτουμε accessions για τα matures), μέχρι την έκδοση 18.0, στην οποία για πρώτη φορά έχουμε αλλαγή ονόματος από ‘cel-let-7’ σε ‘cel-let-7-5p’.

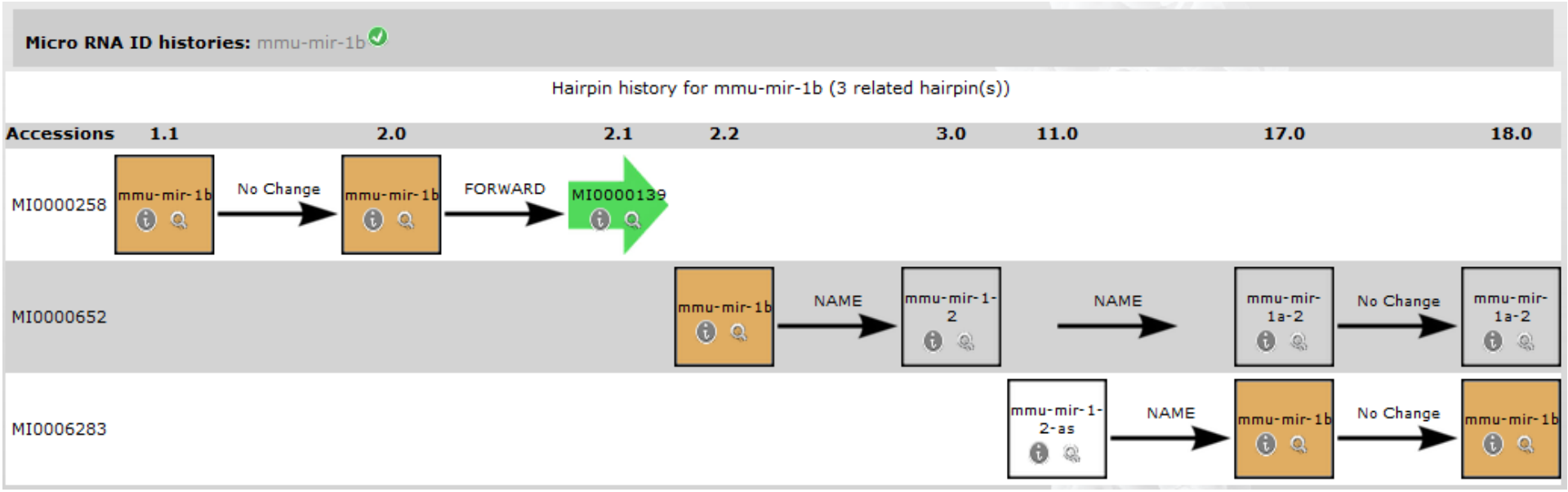
Μπορούμε να προβάλλουμε λεπτομέρειες για τα δεδομένα που καταγράφονται σε κάποιο διάστημα εκδόσεων, πατώντας με το ποντίκι στο πλήκτρο με την ένδειξη “*i* ” που βρίσκεται μέσα στον αντίστοιχο κόμβο. Για παράδειγμα πατώντας το πλήκτρο αυτό στον κόμβο που αντιστοιχεί στην έκδοση 2.0 για το hairpin MI0000001 εμφανίζουμε το ακόλουθο αναδυόμενο παράθυρο:



4.11 Αναδυόμενο παράθυρο πληροφοριών για το cel-let-7 στην έκδοση 2.0

Με το πάτημα στο δεύτερο κουμπί του κάθε κόμβου, ξεκινάμε μια νέα αναζήτηση με βάση το περιεχόμενο του αντίστοιχου κόμβου όπου βρίσκεται το κουμπί αυτό.

Παρουσιάζουμε μερικές ακόμα περιπτώσεις γράφων εξέλιξης και της ερμηνείας τους. Έστω ότι πραγματοποιούμε αναζήτηση στο ID ‘mmu-mir-1b’. Ο γράφος που προκύπτει δίνεται στην εικόνα 4.12.



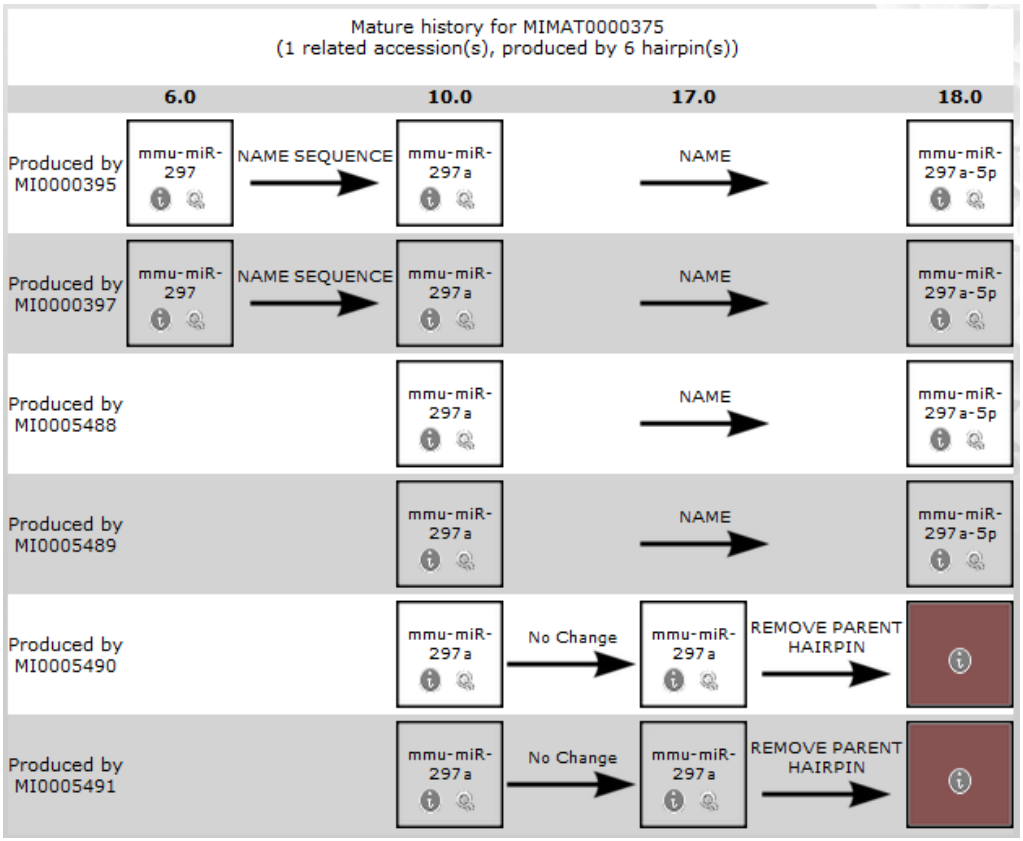
4.12 Αποτέλεσμα αναζήτησης για το 'mmu-mir-1b'

Η πληροφορία που δίνει η εικόνα 4.11 είναι η εξής: Το όνομα ‘mmu-mir-1b’ το φέρουν συνολικά τρία διαφορετικά hairpins σε κάποια έκδοση της mirbase. Αρχικά το hairpin MI0000258, από την έκδοση 1.1 ως την έκδοση 2.0. Στο διάστημα αυτό διατηρεί σταθερά δεδομένα. Εν συνεχεία διαγράφεται από τη βάση στην έκδοση 2.1 και γίνεται προώθησή του στο hairpin με accession MI0000139.

Στη συνέχεια το όνομα που αναζητήθηκε το φέρει το hairpin με accession MI0000652. Το hairpin αυτό διατηρεί το όνομα, μέχρις ότου στην έκδοση 3.0 αλλάζει σε ‘mmu-mir-1-2’. Στη συνέχεια έχουμε άλλη μια αλλαγή ονόματος, στην έκδοση 17.0, όπου το καινούριο ID είναι το ‘mmu-mir-1a-2’, το οποίο διατηρείται ως την τρέχουσα έκδοση.

Τέλος, το όνομα που αναζητήθηκε αντιστοιχεί και στο hairpin με accession MI0006283. Το όνομα αυτό το φέρει στις εκδόσεις 17.0 ως 18.0 (τρέχουσα), ενώ πριν από αυτές τις εκδόσεις το ID του ήταν ‘mmu-mir-1-2-as’, το οποίο έφερε από την έκδοση 11.0 - στην οποία καταγράφεται για πρώτη φορά - και έπειτα.

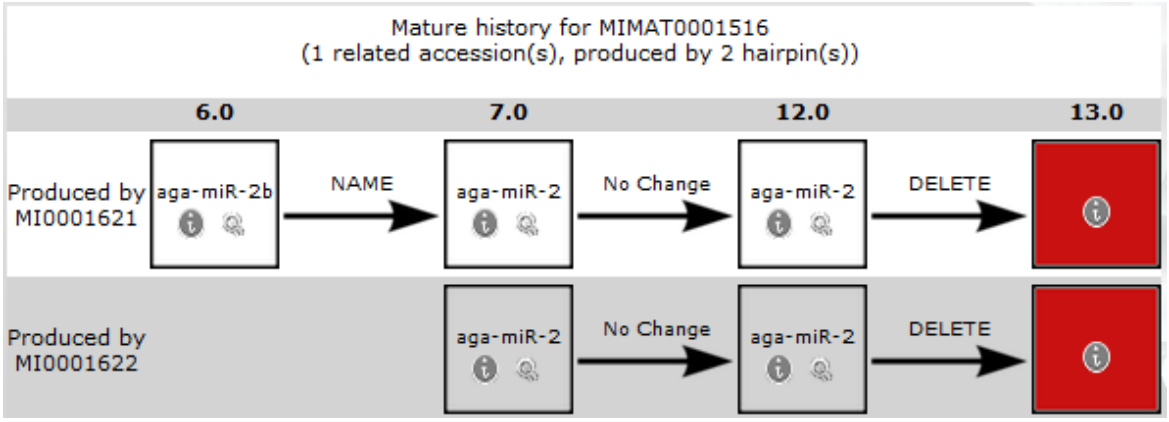
Η αναζήτηση που πραγματοποιούμε μπορεί να γίνει και βάσει κάποιου accession. Έστω ότι αναζητάμε την ιστορία του mature accession MIMAT0000375. Ο γράφος εξέλιξης δεδομένων που προκύπτει φαίνεται στην εικόνα 4.13.



4.13 Εξέλιξη δεδομένων του accession MIMAT0000375

Όπως προκύπτει από το γράφο, το mature αυτό παράγεται από 6 διαφορετικά hairpins. Δύο από αυτά καταγράφονται να το παράγουν αρχικά στην έκδοση 6.0. Στη συνέχεια αλλάζει το ID, καθώς και η αλληλουχία βάσεων που καταγράφονται για αυτά στην έκδοση 10.0. Παράλληλα καταγράφεται η παραγωγή του mature για πρώτη φορά από άλλα τέσσερα hairpins. Η καταγραφή της παραγωγής του mature παύει να καταγράφεται για δύο από αυτά τα hairpins στην έκδοση 18.0. Αυτό φαίνεται από την ένδειξη REMOVE PARENT HAIRPIN πάνω από την ακμή του γράφου και από το ιώδες χρώμα των κόμβων. Για το mature, όπως παράγεται από τα υπόλοιπα hairpins, έχουμε μια ακόμα αλλαγή στο καταγεγραμμένο ID στην έκδοση 18.0.

Τέλος παρουσιάζουμε την περίπτωση αναζήτησης του mature accession MIMAT0001516, για να διαχωρίσουμε την περίπτωση ολικής διαγραφής μιας εγγραφής από τη βάση, σε αντιδιαστολή με το προηγούμενο παράδειγμα. Ο γράφος εξέλιξης δεδομένων που λαμβάνουμε φαίνεται στην ακόλουθη εικόνα:



4.14 Εξέλιξη των δεδομένων του mature accession MIMAT0001516

Όπως βλέπουμε το mature αυτό καταγράφεται να παράγεται από δύο hairpins, με accessions MI0001621 και MI0001622. Η πρώτη περίπτωση καταγράφεται από την έκδοση 6.0, ενώ η δεύτερη από την έκδοση 7.0, όπου η πρώτη έχει και αλλαγή στο ID. Στη συνέχεια τα στοιχεία παραμένουν σταθερά και για τις δύο περιπτώσεις μέχρι την έκδοση 12.0. Στην έκδοση 13.0 έχουμε συνολική διαγραφή του mature από τη βάση, κάτι το οποίο φαίνεται οπτικά και με τη χρήση του κόκκινου χρώματος στους αντίστοιχους κόμβους.

5

Επίλογος

Ανακεφαλαιώνουμε σε αυτή τη παράγραφο το αντικείμενο της διπλωματικής εργασίας. Επισημαίνονται οι χρήσεις και οι δυνατότητες των προϊόντων που προέκυψαν από αυτήν. Γίνεται επίσης αναφορά σε πιθανές μελλοντικές επεκτάσεις τους, καθώς και σε νέους τομείς έρευνας που προκύπτουν πάνω στο αντικείμενο διαχείρισης και εξέλιξης βιολογικών δεδομένων.

5.1 Σύνοψη

Η παρούσα διπλωματική εργασία είχε ως αντικείμενο την καταγραφή και μοντελοποίηση των αλλαγών που υφίστανται τα δεδομένα που καταγράφονται για micro RNAs καθώς προχωράει η έρευνα για τα βιομόρια αυτά. Στη βάση αυτή μελετήθηκαν τα αρχεία της βάσης βιολογικών δεδομένων mirbase. Μετά από μελέτη του περιεχομένου τους αναπτύχθηκε κώδικας που εξάγει από αυτά την απαιτούμενη πληροφορία που αφορά τις αλλαγές αυτές. Η πληροφορία αυτή χρησιμοποιήθηκε για την ανάπτυξη βάσης δεδομένων που συμπυκνώνει τα δεδομένα με τα οποία καταγράφονταν όλες οι εγγραφές της mirbase από την πρώτη έκδοσή της, μέχρι σήμερα. Τελικό προϊόν της εργασίας αποτελεί εφαρμογή που ενσωματώθηκε στο

λογισμικό DIANA microT v4.0. Η εφαρμογή αυτή προβάλλει σε έναν ερευνητή βιολόγο ένα γράφο, που δίνει οπτικά την πορεία της αλλαγής των καταγεγραμμένων δεδομένων για έναν όρο αναζήτησης που τον ενδιαφέρει.

Στα πλαίσια της εργασίας επιτεύχθηκαν οι ακόλουθοι στόχοι:

- Δημιουργία βάσης δεδομένων που καταγράφει όλες τις αλλαγές στα δεδομένα για micro RNAs. Η βάση αποτελείται από δύο πίνακες, έναν που αφορά μόρια hairpins και έναν που αφορά μόρια mature micro RNA.
- Υλοποίηση κώδικα που συγκρίνει τις καταγεγραμμένες αλλαγές με αυτές που δίνονται σε ειδικά αρχεία της βάσης mirbase.
- Υλοποίηση κώδικα για την αναβάθμιση των δεδομένων της βάσης, όταν κυκλοφορούν νέες εκδόσεις της βάσης mirbase.
- Ανάπτυξη εφαρμογής widget που ενσωματώνεται στο λογισμικό DIANA microT v4.0 και προβάλλει γράφους εξέλιξης δεδομένων για κάθε όρο αναζήτησης που ενδιαφέρει το βιολόγο ερευνητή.

5.2 Μελλοντικές εργασίες

Η εφαρμογή που αναπτύχθηκε ενσωματώθηκε σε μια από τις πολλές εφαρμογές της ιστοσελίδας DIANA. Επιπλέον αφορά την εξέλιξη δεδομένων που σχετίζονται με συγκεκριμένο τύπο μορίων RNA, τα micro RNAs. Τα δύο αυτά στοιχεία δίνουν τα όρια των πλαισίων στα οποία κινήθηκε η εργασία, αλλά προσδιορίζουν και τους άξονες στους οποίος μπορεί να κινηθεί η μελλοντική έρευνα. Στη συνέχεια δίνονται ορισμένες τέτοιες κατευθύνσεις.

Επέκταση εφαρμογής γράφων εξέλιξης σε δεδομένα γονιδίων της βάσης ensEMBL

Τα προβλήματα που προκύπτουν από τις αλλαγές στην καταγραφή βιολογικών δεδομένων δεν περιορίζονται μόνο στα βιομόρια τύπου micro RNA. Μπορούν να αφορούν για παράδειγμα και ολόκληρα γονίδια οργανισμών που μελετώνται. Το πρόβλημα αυτό εκτέθηκε στο κεφάλαιο 2, με τη συνοπτική παρουσίαση της βάσης ensEMBL.

Για την υποστήριξη της έρευνας στο πεδίο της βιολογίας επομένως θα πρέπει να αναπτυχθούν αντίστοιχες εφαρμογές που ειδικεύονται σε άλλα είδη βιομορίων. Σε κάθε τέτοια περίπτωση θα πρέπει να αντιμετωπιστούν και τα επιμέρους ιδιαίτερα προβλήματα, που θα είναι συνάρτηση και του είδους των δεδομένων που θα

καταγραφούν. Για παράδειγμα, στην καταγραφή γονιδίων θα πρέπει να αντιμετωπιστεί το ζήτημα της επεξεργασίας του όγκου των δεδομένων, καθώς το καθένα από αυτά έχει μήκος ακολουθίας πολλά χιλιάδες νουκλεοτίδια. Επιπλέον κάθε αρχείο της βάσης ensEMBL έχει μέγεθος μερικά GB, ενώ καταγράφει μόνο ένα μέρος των γονιδίων που ενδιαφέρουν για έναν οργανισμό.

Ενσωμάτωση νεότερων εκδόσεων της mirbase στις εφαρμογές DIANA

Πολλές από τις εφαρμογές της ιστοσελίδας DIANA αντλούν στοιχεία για βιολογικά δεδομένα από βάσεις δεδομένων που διαθέτει η ιστοσελίδα. Τα δεδομένα αυτά μπορεί να αντανακλούν το περιεχόμενο από συγκεκριμένες εκδόσεις των βάσεων από τις οποίες έχουν ληφθεί. Το αποτέλεσμα είναι να χάνονται νεότερα δεδομένα, ή όσα υπάρχουν να μην είναι σύμφωνα με τις τρέχουσες εκδόσεις των βάσεων αυτών. Για παράδειγμα η εφαρμογή microT v4.0 διαθέτει δεδομένα για mature micro RNAs με accessions που εμφανίζονται το αργότερο στην έκδοση 13.0 της mirbase, ενώ τρέχουσα έκδοση είναι η 18.0. Τίθεται επομένως το ζήτημα της ενημέρωσης των δεδομένων αυτών για κάθε εφαρμογή που αντιμετωπίζει το πρόβλημα αυτό.

Πρόσθετες λειτουργίες και χρήση σε άλλες εφαρμογές DIANA

Οι υπηρεσίες που προσφέρει στην παρούσα φάση η εφαρμογή που αναπτύχθηκε είναι η προβολή των γράφων εξέλιξης, η παροχή ενός κουμπιού για την προώθηση της αναζήτησης στο περιεχόμενο ενός κόμβου του γράφου και η προβολή λεπτομερειών που αφορούν τα δεδομένα που καταγράφονται για ένα συγκεκριμένο διάστημα εκδόσεων. Οι λειτουργίες που είναι επιθυμητές, ή απαραίτητες για την υποστήριξη της έρευνας μπορεί να είναι περισσότερες.

Βασική ανάγκη στη διαδικασία της έρευνας είναι η μελέτη της αντίστοιχης βιβλιογραφίας. Κατά τη μελέτη ενός micro RNA, το οποίο είναι γνωστό με συγκεκριμένο όνομα (ID), ο ερευνητής ενδέχεται να παραμελήσει δημοσιεύσεις που αφορούν το ίδιο micro RNA, αλλά αναφέρονται σε αυτό με άλλο όνομα. Επέκταση της εφαρμογής μπορεί να αποτελέσει η χρήση του προβαλλόμενου ιστορικού για τη διεύρυνση της βιβλιογραφίας που παρουσιάζεται στον ερευνητή. Αυτό μπορεί να υλοποιηθεί με έναν μηχανισμό που προσφέρει συνδέσμους προς δημοσιεύσεις, οι οποίες σχετίζονται με όλα τα ID που είχε κάποια στιγμή ένα accession, το οποίο αναζητήθηκε. Επίσης μπορούν να δίνονται σύνδεσμοι προς δημοσιεύσεις που αφορούν όλα τα accession που φέρουν κάποιο συγκεκριμένο ID. Διευρύνεται έτσι η

δυνατότητα αναζήτησης δημοσιεύσεων για βιομόρια που ενδιαφέρουν το βιολόγο ερευνητή.

Επιπλέον, κάθε εφαρμογή της ιστοσελίδας DIANA εξυπηρετεί συγκεκριμένες ερευνητικές ανάγκες. Το πρόβλημα των αλλαγών στην καταγραφή δεδομένων αφορά όλες τις εφαρμογές. Η ενσωμάτωση όμως της εφαρμογής της διπλωματικής εργασίας, ως επέκταση στο υπάρχον λογισμικό της ιστοσελίδας, ενδέχεται να χρειάζεται να προσαρμοστεί διαφορετικά για τις ειδικές ανάγκες κάθε εφαρμογής. Ορισμένες πρόσθετες λειτουργίες που μπορεί να απαιτούνται ενδέχεται να διαφέρουν. Οι απαιτήσεις των λειτουργιών αυτών πρέπει να αναλυθούν και να ενσωματωθούν ξεχωριστά σε κάθε περίπτωση.

6

Βιβλιογραφία

[1]http://employees.csbsju.edu/hjakubowski/classes/chem%20and%20society/cent_dogma/olcentdogma.html

[2] http://www.cbs.dtu.dk/staff/dave/DNA_CenDog.html

[3] http://ghr.nlm.nih.gov/handbook/basics/dna

[4] Wigard P. Kloosterman and Ronald H.A. Plasterk. The Diverse Functions of MicroRNAs in Animal Development and Disease. Developmental Cell, Volume 11, Issue 4, 441-450, 1 October 2006.

[5] Y. Zeng. Principles of micro-RNA production and maturation. Oncogene. 2006 Oct 9;25(46):6156-62.

[6] http://www.mirbase.org/

[7] Ana Kozomara and Sam Griffiths-Jones. miRBase: integrating microRNA annotation and deep-sequencing data. D152–D157 Nucleic Acids Research, 2011, Vol. 39, Database issue doi:10.1093/nar/gkq1027. Published online 30 October 2010

[8] ftp://mirbase.org/pub/mirbase/18/README

[9] ftp://mirbase.org/pub/mirbase/

[10] http://www.mirbase.org/blog/2011/04/whats-in-a-name/

[11] http://www.ebi.ac.uk/embl/

[12] http://www.ebi.ac.uk/embl/Documentation/User_manual/usrman.html

[13] http://www.ddbj.nig.ac.jp/FT/full_index.html#2.3

[14] http://diana.cslab.ece.ntua.gr/

[15] http://diana.cslab.ece.ntua.gr/?sec=software

[16] http://www.ensembl.org/info/about/intro.html

[17] http://www.ensembl.org/info/data/ftp/index.html

[18] http://www.ensembl.org/info/website/archives/index.html

[19] http://aug2010.archive.ensembl.org/info/about/species.html

[20] http://www.mirbase.org/help/nomenclature.shtml

[21] http://www.apachefriends.org/en/xampp.html

[22] http://wiki.apache.org/httpd/FAQ#What_is_Apache.3F

[23] http://www.modulehosting.com/apache.html

[24] http://www.mysql.com/why-mysql/

[25] http://www.mysql.com/about/

[26] http://searchsystemschannel.techtarget.com/feature/MySQL-features

[27] http://php.net/manual/en/intro-whatis.php

[28] http://php.net/manual/en/intro-whatcando.php

[29] http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc

[30] http://www.yiiframework.com/doc/guide/1.1/en/basics.model

[31] http://www.yiiframework.com/doc/guide/1.1/en/basics.view

[32] http://www.yiiframework.com/doc/guide/1.1/en/basics.controller

[33] http://www.w3schools.com/jquery/jquery_intro.asp

[34] http://jquery.com/

[35] http://www.makeitspendit.com/jquery-advantages-and-disadvantages/

[36] http://www.jscripters.com/jquery-disadvantages-and-advantages/

[37] http://www.w3schools.com/ajax/default.asp

[38] http://www.w3schools.com/ajax/ajax_intro.asp

[39] Patrick J. Paddison, Amy A. Caudy, Emily Bernstein, et al. Short hairpin RNAs (shRNAs) induce sequence-specific silencing in mammalian cells. *Genes Dev.* 2002 16: 948-958.

[40] Tingting Du and Phillip D. Zamore. microPrimer: the biogenesis and function of microRNA. Published by The Company of Biologists 2005 doi:10.1242/dev.02070.

Επίσκεψη στους συνδέσμους που παρατίθενται έγινε τελευταία φορά στις 25/6/2012.