Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Διαδικτυακή εφαρμογή εξερεύνησης βιολογικών αλληλεπιδράσεων με τεχνικές οπτικοποίησης δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Τσιχριτζή Σ. Ιωάννη

Επιβλέπων: Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Διαδικτυακή εφαρμογή εξερεύνησης βιολογικών αλληλεπιδράσεων με τεχνικές οπτικοποίησης δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Τσιχριτζή Σ. Ιωάννη

Επιβλέπων: Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την κάτωθι τριμελή εξεταστική επιτροπή την 15η Φεβρουαρίου 2017.

.....	.....	.....
Ιωάννης Βασιλείου Καθηγητής Ε.Μ.Π.	Ανδρέας–Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Θεόδωρος Δαλαμάγκας Ερευνητής Β ΙΠΣΥ «Αθηνά»

Αθήνα, Φεβρουάριος 2017

.....

Τσιχριτζής Σ. Ιωάννης

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

Περίληψη

Τα micro–RNAs (miRNAs) ανακαλύφθηκαν το 1993 με αφορμή μελέτες στο νηματώδες Caenorhabditis elegans και βρέθηκε ότι αποτελούν έναν εκ των μηχανισμών ελέγχου της γονιδιακής έκφρασης. Τα miRNAs, όπως έχει καθιερωθεί να λέγεται, «αλληλεπιδρούν» με γονίδια μην επιτρέποντάς τους να εκφραστούν. Για τη μελέτη τέτοιων αλληλεπιδράσεων έχουν αναπτυχθεί διάφοροι αλγόριθμοι πρόβλεψης. Ωστόσο, δεν προβλέπουν όλοι οι αλγόριθμοι τις ίδιες αλληλεπιδράσεις καθώς καθένας βασίζει τις προβλέψεις του σε διαφορετικό συνδυασμό βιολογικών παραγόντων.

Στις περισσότερες περιπτώσεις, οι αλγόριθμοι πρόβλεψης διαθέτουν έναν ιστότοπο στον οποίο παρουσιάζονται τα αποτελέσματά τους. Ωστόσο, εάν ένας ερευνητής επιθυμεί να προβεί σε συγκρίσεις μεταξύ των αλγορίθμων ή επιθυμεί να μελετήσει αλληλεπιδράσεις συνδυάζοντας δεδομένα από πολλούς αλγορίθμους, θα αντιμετωπίσει μια βασική δυσκολία: πρώτα πρέπει να συλλέξει τα πρωτογενή δεδομένα (raw data) από όσους αλγορίθμους θέλει να μελετήσει, να τα φέρει σε μορφή κατάλληλη για συγκρίσεις και κατόπιν να προβεί στο κυρίως μέρος της μελέτης του και να εξάγει πιθανά συμπεράσματα.

Στόχος της παρούσης εργασίας είναι η δημιουργία μιας διαδικτυακής εφαρμογής εξερεύνησης αλληλεπιδράσεων γονιδίων–miRNAs με δύο σημαντικές καινοτομίες. Η πρώτη είναι η δυνατότητα συνδυαστικής μελέτης αλληλεπιδράσεων αντιπαραβάλλοντας προβλέψεις διαφορετικών αλγορίθμων. Η δεύτερη είναι οι γραφικές απεικονίσεις των μελετούμενων αλληλεπιδράσεων έτσι ώστε να διευκολυνθεί η ανάλυσή τους, η διαισθητική ερμηνεία των αποτελεσμάτων και η εξαγωγή συμπερασμάτων. Η χρήση των γραφημάτων, ειδικά, αποτελεί μία καινοφανή προσπάθεια να αξιοποιηθούν τεχνικές οπτικοποίησης πληροφοριών στο ερευνητικό πεδίο των miRNAs. Επιπρόσθετα, η εφαρμογή παρέχει συνδέσμους προς άλλες συναφείς βιολογικές βάσεις δεδομένων και ιστοτόπους. Έτσι, ο χρήστης μπορεί άμεσα να αντλήσει επιπλέον πληροφορίες για το εκάστοτε αντικείμενο που μελετά (γονίδιο, miRNA ή συγκεκριμένη αλληλεπίδραση) αξιοποιώντας τα δεδομένα από την αντίστοιχη πηγή.

Σε επίπεδο μεθοδολογίας, η εφαρμογή που δημιουργήθηκε θα μπορούσε να αποτελέσει ένα πρωτότυπο του α) πώς να σχεδιάζονται εργαλεία βιοπληροφορικής που υποστηρίζουν τη συνδυαστική εξερεύνηση πληροφοριών από διαφορετικές και, το κυριότερο, ετερογενείς μεταξύ τους βάσεις δεδομένων, β) πώς μπορούν να αξιοποιηθούν τεχνικές οπτικοποίησης πληροφορίας σε εργαλεία βιοπληροφορικής.

Λέξεις κλειδιά: βιοπληροφορική, γονίδιο, miRNA, αλληλεπιδράσεις, αλγόριθμοι πρόβλεψης αλληλεπιδράσεων, διαδικτυακή εφαρμογή, μεγάλος όγκος δεδομένων, οπτικοποίηση δεδομένων

Abstract

Micro–RNAs (miRNAs) were discovered in 1993 during studies executed upon the Caenorhabditis elegans nematode. Then, miRNAs were identified as a new gene expression regulative mechanism. miRNAs "interact" with genes (a phrase coined to describe that relationship) by downregulating their expression. Various "target prediction algorithms" (TPAs) have been developed to facilitate studying of such interactions. However, not all these algorithms predict the same interactions since each one of them depends on a different combination of biological factors to provide its predictions.

In most cases, TPAs provide a website where their results are available. However, if a researcher wishes to compare TPA results or needs to study interactions by combining data from different TPAs, they will encounter a fundamental difficulty: they must download raw data from all the algorithms into consideration, bring the data in a format that facilitates comparisons and then proceed with the actual study and draw any potential conclusions.

The subject of this diploma thesis is to develop a web application for exploring gene–miRNA interactions with two significant innovations. The first is the capability of investigating interactions combinatorially by comparing predictions from different TPAs. The second is the visualisation of TPA data so as to facilitate data analysis, enhance intuitive interpretation of results and hence make it easier for researchers to draw conclusions. The development of visualisations, especially, constitutes a pioneering effort to leverage visualisation techniques in the miRNA field of study. Moreover, the application provides links to relevant biological databases and websites. Thus, it enables the users utilise the data from those sources and acquire additional information about the items they study (a gene, miRNA or interaction).

Regarding the methodology followed, the developed application could be considered as a prototype on a) how to design bioinformatics tools that support the combinatorial exploration of data from different and, most importantly, heterogeneous databases, b) how to leverage data visualisation techniques on bioinformatics tools.

Keywords: bioinformatics, gene, micro-RNA (miRNA), interactions, target prediction algorithms, web application, big data, data visualisation

Περιεχόμενα

1

Εισαγωγή

1.1 micro–RNAs & αλγόριθμοι πρόβλεψης αλληλεπιδράσεων

1.1.1 Τι είναι τα miRNAs και γιατί είναι σημαντική η μελέτη τους

Τα micro–RNAs ανακαλύφθηκαν το 1993 στο νηματώδες Caenorabditis elegans με το πρώτο miRNA που εντοπίστηκε να είναι το lin-4 [2]. Πρόκειται για μικρά μόρια RNA μήκους περίπου 22 νουκλεοτιδίων (σε κάποιες περιπτώσεις από 19 έως 25 [3]) τα οποία δρουν καταστέλλοντας την έκφραση των γονιδίων στο μετα–μεταγραφικό στάδιο (post transcriptional gene regulation) [4]. Αυτό επιτυγχάνεται όταν τα miRNAs προκαλούν αλλοίωση του προς μετάφραση μεταγράφου ή όταν δεν επιτρέπουν τη μετάφρασή του σε πρωτεΐνη [4, 5]. Έτσι αποτελούν έναν σημαντικό μηχανισμό ελέγχου της έκφρασης των γονιδίων [3].

Η μελέτη των miRNAs έχει προσελκύσει έντονο ερευνητικό ενδιαφέρον καθώς πολλές μελέτες έχουν δείξει ότι τα miRNAs συνδέονται με διάφορες ασθένειες όπως καρδιαγγειακές παθήσεις, νευρολογικές παθήσεις, μεταβολικές διαταραχές [6] ενώ πολλές φορές καθορίζουν την ανάπτυξη κυττάρων (όπως κύτταρα του ανοσοποιητικού συστήματος, κάτι που επηρεάζει την απόκριση του ανοσοποιητικού) [3]. Το ακόμη σημαντικότερο, όμως, είναι ότι η λειτουργία των miRNAs έχει συνδεθεί επανειλημμένως με διάφορους τύπους καρκίνων [3, 6]. Τα miRNAs, σε αυτή την περίπτωση, επηρεάζουν με δύο τρόπους. Στη μία περίπτωση, η έλλειψη ενός miRNA μπορεί να επιτρέπει την έκφραση γονιδίων που δε θα έπρεπε να εκφράζονται και άρα να δημιουργείται ένας όγκος. Αντίστοιχα, η δράση ενός miRNA μπορεί να μην επιτρέπει την έκφραση ενός ογκοκατασταλτικού γονιδίου (tumour suppressor gene) [5] οπότε, σε αυτή την περίπτωση, δεν καταφέρνει ο οργανισμός να αντιμετωπίσει έναν όγκο ενώ θα μπορούσε.

Είναι σημαντικό, λοιπόν, να κατανοηθεί πλήρως ο τρόπος που λειτουργούν τα miRNAs έτσι ώστε να αξιοποιηθούν σε θεραπείες ασθενειών, μία προσπάθεια που ήδη έχει ξεκινήσει. Ορισμένα ενδεικτικά

παραδείγματα εφαρμογών των miRNAs στην ιατρική και τη φαρμακευτική είναι τα εξής:
Christopher et al. (2016) "MicroRNA therapeutics: Discovering novel targets and developing specific therapy", Ling et al. (2015) "MicroRNAs and other non-coding RNAs as targets for anticancer drug development", Schmidt (2014) "Drug target miRNAs: chances and challenges", Zhang et al. (2011) "Emerging role of microRNAs in drug response", Wu (2010) "MicroRNA: Potential Targets for the Development of Novel Drugs?".

1.1.2 Η σημασία των αλγορίθμων πρόβλεψης στη μελέτη των miRNAs

Παρά το γεγονός ότι η ύπαρξή των miRNAs είναι γνωστή σχεδόν δυόμιση δεκαετίες και παρά την εκτενή έρευνα που έχει πραγματοποιηθεί επάνω σε αυτά, ο *in vivo* μηχανισμός δράσης τους δεν έχει γίνει ακόμη πλήρως κατανοητός [3]. Η αποτελεσματική πρόβλεψη αλληλεπιδράσεων είναι ιδιαιτέρως απαιτητική διότι, αφενός, ο μηχανισμός των αλληλεπιδράσεων είναι περίπλοκος ενώ, αφετέρου, και η γνώση γύρω από τους κανόνες που διέπουν αυτή τη διαδικασία είναι περιορισμένη [6]. Ακριβώς γι' αυτούς τους λόγους, ένα από τα βασικά σημεία της έρευνας γύρω απ' τα miRNAs είναι να προσδιοριστούν με ακρίβεια οι παράγοντες εκείνοι που καθορίζουν πότε ένα miRNA θα αλληλεπιδράσει με ένα μετάγραφο [3].
Το θεμελιώδες πρώτο βήμα προς αυτή την κατεύθυνση είναι να εντοπιστούν πιθανοί στόχοι των miRNAs. Για να γίνει αυτό εργαστηριακά, απαιτείται πολύς χρόνος, σημαντική προσπάθεια ενώ ενδεχομένως είναι και οικονομικά ασύμφορο δεδομένου του μεγάλου αριθμού των miRNAs και του τεράστιου αριθμού των πιθανών στόχων [3]. Για παράδειγμα, στον άνθρωπο αυτή τη στιγμή η Ensembl αριθμεί περί τα 215,000 μετάγραφα και η miRBase κάτι λιγότερο από 2,000 miRNAs. Επομένως γίνεται αντιληπτό ότι, για τέτοιες τάξεις μεγεθών, η εξαγωγή σαφών συμπερασμάτων σε εύλογο χρονικό διάστημα με τις κλασσικές *in vitro* εργαστηριακές διαδικασίες είναι αρκετά δύσκολη.
Εδώ ακριβώς υπεισέρχεται ο ρόλος των αλγορίθμων πρόβλεψης αλληλεπιδράσεων οι οποίοι, αξιοποιώντας την ισχύ των σημερινών υπολογιστών, μπορούν να προβούν σε μία πολύ πιο εκτεταμένη αναζήτηση αλληλεπιδράσεων σε πολύ μικρό χρονικό διάστημα. Έτσι, ένας ερευνητής που αναζητά συγκεκριμένες πληροφορίες μπορεί να έχει μία σαφή αφετηρία βασιζόμενος στα αποτελέσματα των αλγορίθμων πρόβλεψης. Για παράδειγμα, μπορεί να λάβει μια πρώτη ένδειξη για πιθανούς στόχους ενός miRNA ή, αντίστοιχα, πιθανά miRNA που στοχεύουν ένα επιθυμητό γονίδιο. Συνεπώς, η έρευνά του, μπορεί να επικεντρωθεί στα πιο πιθανά ζεύγη γονιδίων–miRNAs και τα όποια επακόλουθα πειράματα να γίνουν επάνω σε αυτά.

1.1.3 Η σημασία των συγκρίσεων μεταξύ αλγορίθμων

Οι διάφοροι αλγόριθμοι πρόβλεψης τις περισσότερες φορές εμφανίζουν σημαντικές διαφορές στα αποτελέσματά τους. Αυτό συμβαίνει διότι βασίζονται σε διαφορετικούς παράγοντες για τις προβλέψεις τους ενώ αποδίδουν και διαφορετική βαρύτητα σε καθέναν από αυτούς [7, 8]. Με δεδομένη, λοιπόν, τη χρησιμότητά των αλγορίθμων και ακριβώς λόγω των διαφορών που παρουσιάζουν στις προβλέψεις τους, η

συναλήθευση αποτελεσμάτων και η επεξεργασία αντικρουόμενων προβλέψεων θα μπορούσε να προσφέρει ακόμη μεγαλύτερα οφέλη.

Η σημασία που αποδίδεται στη σύγκριση των αλγορίθμων καταδεικνύεται και από αρκετές δημοσιεύσεις που συγκρίνουν αλγορίθμους (αναφέρουμε ενδεικτικά τις [3, 6, 7, 9]), παρ' ότι η σύγκριση δεν αφορά πάντα τα αποτελέσματα των αλγορίθμων αυτά καθαυτά αλλά τις μεθοδολογίες τους. Στις μελέτες [8, 10] έχουν γίνει προσπάθειες να αναπτυχθούν μέθοδοι που αυτόματα συνδυάζουν αποτελέσματα από διαφορετικούς αλγορίθμους ώστε να εξάγονται προβλέψεις με μεγαλύτερη ακρίβεια.

Επίσης, κάποιες μελέτες όπως οι [6, 8], υπογραμμίζουν ότι σε κάποιες περιπτώσεις οι προβλέψεις ενός μεμονωμένου αλγορίθμου δεν είναι πάντοτε αξιόπιστες ή μπορεί να μην επαληθεύονται πειραματικά, ενώ οι λόγοι που συμβαίνει αυτό δεν είναι πάντοτε ξεκάθαροι. Αυτός είναι άλλος ένας λόγος για τον οποίο είναι σημαντικό να μπορεί κανείς να αντιπαραβάλλει στοιχεία από διαφορετικούς αλγορίθμους.

Σε κάθε περίπτωση, πάντως, το ζητούμενο από τις συγκρίσεις είναι να βρεθούν αλληλεπιδράσεις οι οποίες αξιολογούνται ως «πιθανές» όχι μόνο από έναν αλγόριθμο αλλά από περισσότερους. Διότι, αν μία αλληλεπίδραση αξιολογείται ως «πιθανή» από πολλούς αλγορίθμους, αυτό σημαίνει ότι πληροί τα περισσότερα από τα διαφορετικά κριτήρια αυτών των αλγορίθμων. Επομένως, η πιθανότητα αυτή η συγκεκριμένη αλληλεπίδραση να συμβαίνει όντως και στην πραγματικότητα είναι αυξημένη.

Μία ακόμη ένδειξη ότι οι ερευνητές πράγματι ενδιαφέρονται να συλλέγουν αποτελέσματα από διάφορους αλγορίθμους είναι η πληθώρα ιστοτόπων, όπως οι [23, 24, 25], που διατηρούν συγκεντρωτικές λίστες αλγορίθμων.

1.1.4 Επιστημονική στάθμη

Σε αντίθεση με τα όσα αναφέρθηκαν προηγουμένως, αυτή τη στιγμή υπάρχουν πολύ λίγα εργαλεία που υποστηρίζουν τέτοιες συγκρίσεις και κανένα που να συνοδεύει τις συγκρίσεις με δυνατότητες οπτικοποίησης πληροφορίας. Παρ' ότι βρισκόμαστε στην εποχή των μεγάλων όγκων δεδομένων (big data), παρ' ότι οι τεχνικές οπτικοποίησης πληροφοριών (data visualisation) είναι ιδιαίτερα δημοφιλείς σε άλλους τομείς της βιολογίας [26, 54], παρ' ότι η έρευνα για τα miRNAs παράγει συνεχώς νέα γνώση και παρ' ότι η γνώση αυτή είναι ήδη οργανωμένη σε σημαντικό βαθμό (έστω και σε διαφορετικές πηγές), υπάρχει σαφής έλλειψη εργαλείων γύρω από τα miRNAs που να αξιοποιούν αυτές τις δυνατότητες και να συνεισφέρουν προς αυτή την κατεύθυνση.

Ως προς τις συγκρίσεις, οι αλγόριθμοι πρόβλεψης συνήθως διαθέτουν τα αποτελέσματά τους ελεύθερα στο Διαδίκτυο αλλά ελάχιστοι συγκρίνουν τα αποτελέσματά τους με άλλους. Για να προβεί ένας ερευνητής σε συγκριτική επεξεργασία αποτελεσμάτων πρέπει να συλλέξει τα πρωτογενή δεδομένα των αλγορίθμων, να τα επεξεργαστεί κατάλληλα, να τα φέρει σε μορφή που να επιτρέπει συγκρίσεις και αφού καταβάλει πρώτα όλη αυτή τη σημαντική προσπάθεια, μόνο τότε θα είναι σε θέση να προχωρήσει στο βασικό σκέλος της μελέτης του.

Το ακόμη σημαντικότερο ως προς την οπτικοποίηση των πληροφοριών, είναι πως ο όγκος δεδομένων γύρω από τα miRNAs είναι τόσο μεγάλος που η εξαγωγή συμπερασμάτων από πίνακες και αρχεία κειμένου είναι αρκετά δύσκολη όταν αυτά περιέχουν μερικά εκατομμύρια εγγραφές. Ωστόσο, ακόμη και στα εργαλεία που

ήδη υπάρχουν, ο βασικός τρόπος προβολής των αποτελεσμάτων είναι η παρουσίαση σε μορφή πίνακα και σπάνια παρέχεται κάποιο στοιχειώδες οπτικό βοήθημα (π.χ. ο TargetScan παρουσιάζει σε πολύ βασικό επίπεδο τις διάφορες θέσεις πρόσδεσης επάνω σε ένα μετάγραφο). Η οπτικοποίηση πληροφορίας, μάλιστα, έχει αποδεδειγμένα βοηθήσει σε άλλους τομείς ως προς την καλύτερη κατανόηση μεγάλων όγκων δεδομένων (π.χ. επιχειρηματικότητα με business analytics, εργοστάσια με process control analytics κ.ά.).
Συνεπώς, με τα υπάρχοντα εργαλεία, υπάρχουν σημαντικά περιθώρια βελτίωσης τόσο ως προς την αξιοποίηση των συγκρίσεων όσο και ως προς τη χρήση τεχνικών οπτικοποίησης πληροφορίας.

1.2 Αντικείμενο & συνεισφορά της διπλωματικής

Σκοπός της παρούσης εργασίας, λοιπόν, είναι να αναπτυχθεί μία εφαρμογή η οποία θα υποστηρίζει τη συγκριτική παρουσίαση αλληλεπιδράσεων από διαφορετικούς αλγορίθμους πρόβλεψης και θα παρέχει δυνατότητες οπτικοποίησης των πληροφοριών (data visualisation).

Η συνεισφορά της εργασίας έγκειται στα εξής τρία σημεία:

1) **Η εφαρμογή παρέχει γραφήματα που οπτικοποιούν τα δεδομένα των αλληλεπιδράσεων, διευκολύνοντας την κατανόηση των πληροφοριών και την εξαγωγή συμπερασμάτων**. Ειδικά ως προς αυτό το σημείο, δεν υπάρχουν προς το παρόν διαθέσιμα εργαλεία που να υποστηρίζουν την οπτικοποιημένη σύγκριση αλληλεπιδράσεων και αλγορίθμων πρόβλεψης. Μάλιστα, τα γραφήματα που παρέχονται από την εφαρμογή, είναι διαδραστικά και επιτρέπουν στο χρήστη να αλληλεπιδρά με τα δεδομένα σε πραγματικό χρόνο. Αυτή είναι και η πρώτη προσπάθεια εφαρμογής τεχνικών οπτικοποίησης πληροφορίας στο ερευνητικό πεδίο των miRNAs.
2) **Η εφαρμογή υποστηρίζει τη συγκριτική παρουσίαση αλληλεπιδράσεων από τρεις αλγορίθμους ταυτόχρονα**. Έτσι μπορεί να αξιολογηθεί μία αλληλεπίδραση με βάση τα κριτήρια τριών αλγορίθμων και όχι μόνο ενός κάθε φορά. Αυτό βοηθά να εντοπιστούν εκείνες οι αλληλεπιδράσεις που αξιολογούνται ως πιθανές και από τις τρεις επιμέρους πηγές. Ένα ακόμη πιθανό όφελος είναι η παρατήρηση μοτίβων που ενδεχομένως υπάρχουν, π.χ. αν κάποιος αλγόριθμος βαθμολογεί συστηματικά καλύτερα ή χειρότερα από τους άλλους κάποια ομάδα αλληλεπιδράσεων. Σε τέτοιες περιπτώσεις θα μπορούσαν να αναζητηθούν οι λόγοι που συμβαίνει αυτό.
3) **Η εφαρμογή υποστηρίζει την εξερεύνηση αλληλεπιδράσεων με πολλούς τρόπους**. Π.χ. αναζήτηση αλληλεπιδράσεων για μεμονωμένα γονίδια ή miRNAs αλλά και για συνδυασμούς αυτών, αναζήτηση με βάση κάποιον μεμονωμένο αλγόριθμο ή πολλούς κτλ.

1.2.1 Συνεισφορά της εργασίας ως μεθοδολογία

Ωστόσο, πιστεύουμε πως, πέραν της εφαρμογής και η ίδια η μεθοδολογία που ακολουθήθηκε στην εργασία θα μπορούσε να αποτελέσει ενδεικτικό τρόπο του πώς να αναπτύσσονται βιολογικές βάσεις δεδομένων. Όπως θα αναλυθεί αργότερα, ο συγκερασμός δεδομένων από διαφορετικές πηγές, οι οποίες τις περισσότερες φορές είναι και ετερογενείς μεταξύ τους, είναι μια αρκετά απαιτητική διαδικασία. Επίσης, η ιδιαίτερη φύση

των βιολογικών δεδομένων, η συνέπεια ή μη των πληροφοριών από τις επιμέρους πηγές και οι αναγκαίοι συμβιβασμοί που επιβάλλονται κατά τη σχεδίαση μιας βάσης δεδομένων, εισάγουν προκλήσεις σε μία αποδοτική σχεδίαση.

Για όλους αυτούς τους λόγους, κρίθηκε σκόπιμο να καταγραφούν και να αναλυθούν λεπτομερώς όλοι εκείνοι οι παράγοντες που επηρέασαν τη σχεδίαση της εφαρμογής, με την ελπίδα ο τρόπος αντιμετώπισης των σχεδιαστικών ζητημάτων αυτών να αποδειχτεί χρήσιμος και σε κάποια παρόμοια εργασία στο μέλλον.

1.3 Διάρθρωση τόμου

Ο τόμος αποτελείται από 11 κεφάλαια συνολικά, 2 εκ των οποίων παραρτήματα. Το περιεχόμενο ανά κεφάλαιο έχει ως εξής:

- **Κεφάλαιο 2**: αναλύεται το θεωρητικό υπόβαθρο της εργασίας τόσο ως προς το βιολογικό μέρος, όπου δίνονται ορισμοί, σχετικές έννοιες κτλ., όσο και ως προς το προγραμματιστικό μέρος, όπου εξηγείται η θεωρία επάνω στην οποία βασίζονται τα εργαλεία που χρησιμοποιήθηκαν στην εργασία.
- **Κεφάλαιο 3**: δίνονται λεπτομέρειες σχετικά με εργαλεία βιοπληροφορικής που χρησιμοποιήθηκαν, όπως οι βιολογικές βάσεις δεδομένων. Επίσης γίνεται αναφορά σε συναφείς εργασίες σχετικά με τη σύγκριση αλγορίθμων πρόβλεψης και την οπτικοποίηση βιολογικής πληροφορίας.
- **Κεφάλαιο 4**: αναλύονται τα προγραμματιστικά εργαλεία και οι τεχνολογίες που αξιοποιήθηκαν στην εργασία.
- **Κεφάλαιο 5**: διατυπώνονται οι απαιτήσεις από την σκοπιά του χρήστη, δηλαδή τι λειτουργίες πρέπει να υποστηρίζονται από την εφαρμογή, καθώς και οι τεχνικές προδιαγραφές του συστήματος.
- **Κεφάλαιο 6**: αναλύεται εκτενώς η διαδικασία υλοποίησης της εφαρμογής. Το κεφάλαιο εστιάζει στο «πώς» έγινε η υλοποίηση.
- **Κεφάλαιο 7**: αναλύονται επιμέρους σχεδιαστικά ζητήματα που προέκυψαν κατά την υλοποίηση. Το κεφάλαιο εστιάζει στο «γιατί» η υλοποίηση έγινε με τον συγκεκριμένο τρόπο.
- **Κεφάλαιο 8**: η σύνοψη της εργασίας, τα σημαντικότερα συμπεράσματα και πιθανές μελλοντικές επεκτάσεις.
- **Κεφάλαιο 9**: βιβλιογραφία.
- **Παράρτημα Ι**: αποτελεί το αναλυτικό εγχειρίδιο της εφαρμογής με εξαντλητική παρουσίαση όλων των δυνατοτήτων που η εφαρμογή προσφέρει στο χρήστη.
- **Παράρτημα ΙΙ**: ευρετήριο των εργαλείων που χρησιμοποιήθηκαν στην εργασία.

2

Θεωρητικό υπόβαθρο

2.1 Βιολογικές έννοιες και ορισμοί

Σε αυτή την πρώτη ενότητα του κεφαλαίου παρατίθενται βιολογικοί ορισμοί και επεξηγούνται έννοιες οι οποίες είναι απαραίτητες για την κατανόηση της εργασίας. Επισημαίνεται πως οι ορισμοί που δίνονται εδώ ***εσκεμμένα δεν είναι εξαντλητικοί***. Μερικά παραδείγματα είναι ότι στον ορισμό του γονιδιώματος αγνοείται το μιτοχονδριακό DNA, στους ορισμούς των νουκλεϊκών οξέων δεν εξετάζονται ειδικές περιπτώσεις (π.χ. μονόκλονα μόρια DNA, δίκλωνα μόρια RNA, RNA ως γενετικό υλικό ιών κτλ.), στον ορισμό των ακολουθιών νουκλεοτιδίων αγνοείται η οργάνωση σε κωδικόνια, ο ορισμός της γονιδιακής έκφρασης είναι σαφώς απλοποιημένος κτλ. Επίσης, οι ορισμοί δίνονται με βάση τη συνήθη περίπτωση στους πολυκύτταρους οργανισμούς αφού τόσο ο άνθρωπος όσο και ο ποντικός (οργανισμοί με τους οποίους ασχοληθήκαμε στην παρούσα εργασία) είναι πολυκύτταροι.

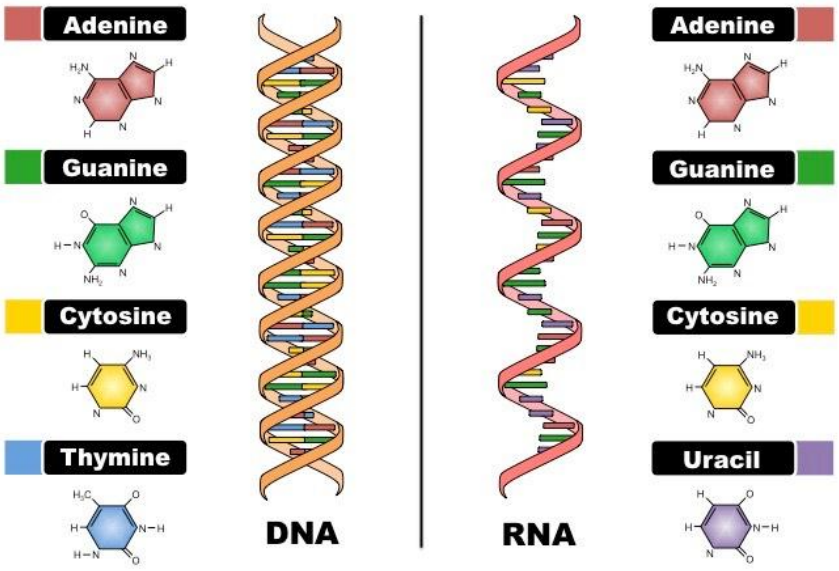
Σκοποί της ενότητας είναι:

- Να σκιαγραφηθεί το βιολογικό υπόβαθρο στο οποίο βασίζεται η εργασία.
- Να αποκτήσει ο μη εξοικειωμένος με τη βιολογία αναγνώστης τη βασική αντίληψη που απαιτείται για την κατανόηση της εργασίας στη συνέχεια.

Ακριβώς γι' αυτούς τους λόγους κρίθηκε σκόπιμο οι ορισμοί να μην υπεισέλθουν σε μεγάλο βιολογικό βάθος, κάτι που θα είχε το ακριβώς αντίθετο αποτέλεσμα από το επιθυμητό. Επομένως, οποιεσδήποτε απλοποιήσεις υπάρχουν στην ενότητα αυτή είναι εσκεμμένες.

2.1.1 Γενικοί ορισμοί και έννοιες βιολογίας

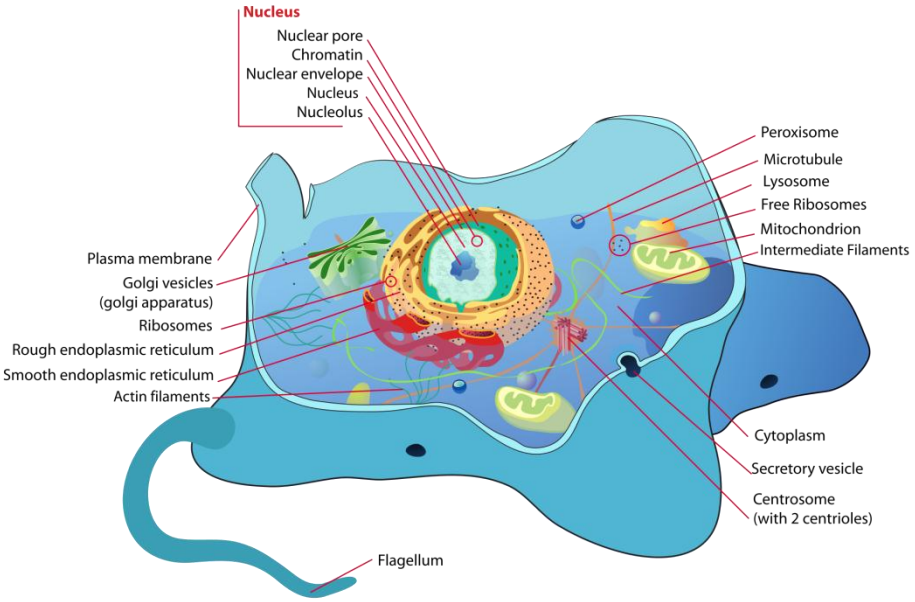
- **Νουκλεοτίδια**: οργανικά μόρια τα οποία αποτελούν τα μονομερή των νουκλεϊκών οξέων, δηλαδή του DNA και του RNA. Αποτελούνται από μία πεντόζη (πεντόζη: μονοσακχαρίτης, δηλαδή μονομερής υδατάνθρακας, με πέντε άτομα άνθρακα που αποτελεί δομικό συστατικό των πολυσακχαριτών), τουλάχιστον μία φωσφορική ομάδα και μία αζωτούχο βάση. Οι αζωτούχες βάσεις που συναντώνται είναι: η αδενίνη (A), η γουανίνη (G), η θυμίνη (T), η κυτοσίνη (C) και η ουρακίλη (U). Τα νουκλεοτίδια που περιέχουν A,G,T,C συμμετέχουν στο σχηματισμό μορίων DNA ενώ νουκλεοτίδια που περιέχουν A,G,U,C συμμετέχουν στο σχηματισμό μορίων RNA. Στο RNA, δηλαδή, η ουρακίλη αντικαθιστά τη θυμίνη.
- **Αλληλουχία νουκλεοτιδίων:** τα νουκλεοτίδια, καθώς διατάσσονται εν σειρά για να δημιουργήσουν ένα μόριο DNA/RNA, δημιουργούν ακολουθίες, π.χ. AGGCTTAGTACTATACGT. Οι γενετικές πληροφορίες κωδικοποιούνται μέσω της διαδοχής των νουκλεοτιδίων, όπως στους υπολογιστές οι πληροφορίες κωδικοποιούνται από διαδοχικά ψηφία 0 και 1.
- **DNA (DeoxyriboNucleic Acid)**: δεοξυριβονουκλεϊκό οξύ. Πρόκειται για πολυμερές, γραμμικό μόριο μεγάλου μήκους το οποίο συγκροτείται από εν σειρά τοποθετημένα νουκλεοτίδια. Αποτελείται από δύο αλυσίδες νουκλεοτιδίων οι οποίες είναι συμπληρωματικές μεταξύ τους. Οι δύο αλυσίδες «διπλώνονται» στο χώρο η μία γύρω από την άλλη σχηματίζοντας διπλή έλικα. Γι' αυτό το λόγο το DNA αποκαλείται «δίκλωνη» αλυσίδα. Αποτελεί το μέσο αποθήκευσης της γενετικής πληροφορίας και βρίσκεται μέσα στον πυρήνα των κυττάρων.
- **Συμπληρωματικότητα νουκλεοτιδίων:** τα νουκλεοτίδια της κάθε αλυσίδας του DNA συνδέονται με δεσμούς υδρογόνου με τα νουκλεοτίδια της απέναντι αλυσίδας. Αυτοί οι δεσμοί επιτυγχάνονται μέσω της συμπληρωματικότητας των αζωτούχων βάσεων των νουκλεοτιδίων. Η αδενίνη (A) συνδέεται πάντοτε με θυμίνη (T) ενώ η γουανίνη (G) συνδέεται πάντοτε με κυτοσίνη (C). Η συμπληρωματικότητα επιτρέπει στο DNA να διπλασιάζεται δημιουργώντας **ακριβές αντίγραφο** του εαυτού του. Έτσι επιτυγχάνεται η μεταβίβαση της γενετικής πληροφορίας αναλλοίωτη στις επόμενες γενεές κυττάρων και, τελικά, στους απογόνους του οργανισμού. Επίσης, η συμπληρωματικότητα είναι αυτή που εξασφαλίζει πως, όταν δημιουργείται ένα μόριο RNA από κάποιο τμήμα DNA, η πληροφορία που περιέχεται στο RNA είναι **πιστό αντίγραφο** της πληροφορίας που περιέχεται στο αρχικό τμήμα DNA. Η συμπληρωματικότητα είναι η ιδιότητα που καθιστά το DNA αξιόπιστο μέσο αποθήκευσης της γενετικής πληροφορίας.
- **Γονίδιο:** τμήμα DNA το οποίο κωδικοποιεί κάποια συγκεκριμένη λειτουργία ή χαρακτηριστικό του οργανισμού, π.χ. πληροφορία για την παραγωγή κάποιας πρωτεΐνης.
- **Μεταγραφή:** η διαδικασία δημιουργίας ενός μορίου RNA από ένα τμήμα DNA, δηλαδή από ένα γονίδιο. Το DNA χρησιμοποιείται ως εκμαγείο και με βάση τη συμπληρωματικότητα παράγεται ένα μόριο RNA. Το παραγόμενο RNA περιέχει ένα ακριβές αντίγραφο της πληροφορίας του τμήματος DNA που μεταγράφηκε. Συμβαίνει μέσα στον πυρήνα του κυττάρου.



Εικόνα 2.1: η δομή του DNA και του RNA και οι αζωτούχες βάσεις

- **RNA (RiboNucleic Acid)**: ριβονουκλεϊκό οξύ. Είναι επίσης γραμμικό μόριο μεγάλου μήκους το οποίο συγκροτείται από εν σειρά τοποθετημένα νουκλεοτίδια. Συνήθως είναι «μονόκλονο», δηλαδή αποτελείται από μία αλυσίδα νουκλεοτιδίων και όχι δύο όπως το DNA. Τα μόρια RNA προκύπτουν από το DNA με τη διαδικασία της μεταγραφής και γι' αυτό στα ελληνικά ονομάζονται «μετάγραφα». Υπάρχουν διάφοροι τύποι RNA ανάλογα με το ρόλο που επιτελούν.
- **mRNA (messenger RNA)**: μόριο RNA το οποίο περιέχει πληροφορία για τη σύνθεση κάποιας πρωτεΐνης. Προκύπτει από μεταγραφή τμήματος του DNA και αποστέλλεται στα ριβοσώματα – εξ ου και το όνομά του. Η σύνθεση των πρωτεϊνών από τα ριβοσώματα γίνεται με βάση τις «οδηγίες» που βρίσκονται κωδικοποιημένες στα μόρια mRNA.
- **non–coding RNAs**: μόρια RNA τα οποία δεν κωδικοποιούν πρωτεΐνες αλλά έχουν άλλους ρόλους. Βεβαίως και αυτά προκύπτουν από μεταγραφή τμημάτων DNA. Ένα παράδειγμα τέτοιου μορίου είναι τα micro–RNAs που αποτελούν αντικείμενο αυτής της εργασίας.
- **Γονιδίωμα**: αποτελεί το σύνολο της γενετικής πληροφορίας που είναι απαραίτητη για την ανάπτυξη και την εύρυθμη λειτουργία ενός οργανισμού. Καταγράφεται σε μόρια DNA και βρίσκεται στον πυρήνα των κυττάρων. Ανάλογα με τον οργανισμό απαιτείται διαφορετικός αριθμός μορίων DNA για την καταγραφή της γενετικής πληροφορίας του. Για παράδειγμα, τα ανθρώπινα κύτταρα περιλαμβάνουν 46 μόρια DNA.
- **Χρωμόσωμα**: ένα μόριο DNA αναδιπλωμένο κατάλληλα στο χώρο. Τα μόρια DNA λόγω του τεράστιου μήκους τους αναδιπλώνονται και οργανώνονται κατάλληλα εντός του κυτταρικού πυρήνα. Έτσι, μετά από διάφορα στάδια αναδίπλωσης, διαμορφώνονται τελικώς τα χρωμοσώματα. Ο άνθρωπος διαθέτει 46 χρωμοσώματα στα κύτταρά του, δηλαδή κάθε ένα από τα μόρια DNA δημιουργεί και ένα χρωμόσωμα. Κάθε χρωμόσωμα διαφέρει σε μήκος ενώ, για να αναγνωρίζονται μεταξύ τους, τα χρωμοσώματα αριθμούνται.

- **Κύτταρο (ευκαρυωτικό)**: η βασική δομική, λειτουργική και βιολογική μονάδα όλων των ζώντων οργανισμών. Είναι η στοιχειώδης μονάδα ζωής που μπορεί να αναπαραχθεί αυτοτελώς. Το κύτταρο οριοθετείται και διαχωρίζεται από το περιβάλλον του με την κυτταρική μεμβράνη. Μέσα στην κυτταρική μεμβράνη υπάρχει το κυτταρόπλασμα στο οποίο βρίσκονται όλα τα σωματίδια και οι ουσίες τα οποία απαιτούνται για τις διάφορες λειτουργίες του κυττάρου. Μέσα στο κυτταρόπλασμα υπάρχει και ο πυρήνας του κυττάρου μέσα στον οποίο βρίσκεται το DNA.



Εικόνα 2.2: ένα ευκαρυωτικό κύτταρο. Διακρίνεται η κυτταρική μεμβράνη, το κυτταρόπλασμα και ο πυρήνας. Το κυτταρόπλασμα καταλαμβάνει την κυρίως μάζα του κυττάρου και μέσα σε αυτό βρίσκονται τα διάφορα κυτταρικά σωματίδια

Οι πολυκύτταροι οργανισμοί, όπως ο άνθρωπος, αποτελούνται από διαφορετικούς τύπους κυττάρων, όπως τα μυϊκά, τα νευρικά, τα κύτταρα του αίματος κτλ. Ωστόσο, κάθε κύτταρο οποιουδήποτε είδους, φέρει ολόκληρο το γονιδίωμα του οργανισμού και όχι μόνο όση γενετική πληροφορία είναι απαραίτητη για τη λειτουργία του εν λόγω είδους κυττάρων. Αυτό συμβαίνει διότι όλα τα κύτταρα προκύπτουν από διαδοχικούς διπλασιασμούς ενός μοναδικού αρχικού κυττάρου: του ζυγωτού.

Εκτός από τα ευκαρυωτικά κύτταρα υπάρχουν και τα προκαρυωτικά τα οποία έχουν σημαντικές διαφορές με τα ευκαρυωτικά και συναντώνται μόνο σε μονοκύτταρους οργανισμούς. Δε θα μας απασχολήσουν εδώ.

- **Ριβοσώματα**: σωματίδια που βρίσκονται στο κυτταρόπλασμα. Αποστολή τους είναι να συνθέτουν πρωτεΐνες με βάση τις πληροφορίες που λαμβάνουν από τα mRNAs.
- **Μετάφραση**: η διαδικασία παραγωγής πρωτεΐνης από τα ριβοσώματα με βάση τις πληροφορίες ενός mRNA. Ονομάζεται έτσι διότι τα mRNAs «μεταφράζονται» σε πρωτεΐνες.
- **Κυτταρική διαφοροποίηση**: η διαδικασία δημιουργίας διαφορετικών τύπων κυττάρων στους πολυκύτταρους οργανισμούς. Επιτυγχάνεται παρά το γεγονός ότι όλα τα κύτταρα διαθέτουν επακριβώς την ίδια γενετική πληροφορία στο DNA του πυρήνα τους. Η διαφοροποίηση συμβαίνει

σε διάφορα στάδια της ζωής του οργανισμού αλλά κυρίως κατά την ανάπτυξή του από ένα απλό ζυγωτό σε ένα ολοκληρωμένο καινούριο άτομο.

- **Γονιδιακή έκφραση**: η διαδικασία αξιοποίησης της γενετικής πληροφορίας ενός γονιδίου ούτως ώστε να επιτελεστεί η λειτουργία που κωδικοποιείται από το γονίδιο αυτό. Στην απλή περίπτωση ενός γονιδίου που κωδικοποιεί πρωτεΐνη, πρόκειται για τη ροή της γενετικής πληροφορίας από το γονίδιο προς τα ριβοσώματα για την παραγωγή της αντίστοιχης πρωτεΐνης. Όταν λέγεται, λοιπόν, πως ένα γονίδιο «εκφράζεται», ***συνήθως*** εννοείται πως παράγεται στο κύτταρο η αντίστοιχη πρωτεΐνη.
- **Γονιδιακή ρύθμιση**: ο μηχανισμός που επιτρέπει στα κύτταρα να ελέγχουν ποιο υποσύνολο των γενετικών τους πληροφοριών θα αξιοποιήσουν και πότε ακριβώς. Συνεπώς, η γονιδιακή ρύθμιση είναι εκείνος ο μηχανισμός που επιτρέπει στα κύτταρα να ρυθμίζουν τη λειτουργία τους κάθε στιγμή. Επιτυγχάνεται με διάφορους τρόπους, ένας εκ των οποίων είναι τα miRNAs. Ένα πολύ χαρακτηριστικό παράδειγμα του τι μπορεί να επιτευχθεί με τη γονιδιακή ρύθμιση είναι η κυτταρική διαφοροποίηση η οποία επιτυγχάνεται παρ' ότι όλα τα κύτταρα ενός ατόμου διαθέτουν επακριβώς την ίδια γενετική πληροφορία.

2.1.2 Ορισμοί και έννοιες σχετικά με τα miRNAs και τις αλληλεπιδράσεις

- **Ώριμο microRNA (miRNA)**: μόριο RNA μικρού μήκους (~22 νουκλεοτίδια) το οποίο δεν κωδικοποιεί πρωτεΐνη. Τα miRNAs συμμετέχουν στη μετα–μεταγραφική γονιδιακή ρύθμιση (post–transcriptional regulation of gene expression). Τα miRNAs προσδένονται επάνω στα mRNAs και είτε επάγουν την καταστροφή τους είτε εμποδίζουν τη μετάφρασή τους από τα ριβοσώματα. Σε κάθε περίπτωση, δεν επιτρέπουν τη δημιουργία πρωτεΐνης κι έτσι, όπως λέγεται, «καταστέλλουν» την έκφραση του αντίστοιχου γονιδίου από το οποίο προήλθαν τα mRNAs.
- **Πρόδρομο miRNA**: τα ώριμα miRNAs δεν προκύπτουν απευθείας μετά τη μεταγραφή ενός τμήματος DNA. Το αρχικό προϊόν της μεταγραφής ενός γονιδίου που κωδικοποιεί miRNA είναι το «πρωτογενές miRNA» (primary miRNA ή pri–miRNA). Από το πρωτογενές miRNA, κατόπιν επεξεργασίας, προκύπτει το «πρόδρομο miRNA» (miRNA precursor ή pre–miRNA) το οποίο πρέπει να υποστεί περαιτέρω επεξεργασία για να προκύψει ένα ώριμο miRNA. Κάθε πρόδρομο miRNA δίνει 1 ή (συνηθέστερα) 2 ώριμα miRNAs, ένα από κάθε άκρο της αλυσίδας του.
- **Περιοχή πρόσδεσης ενός miRNA (seed region)**: εκείνο το τμήμα της ακολουθίας ενός miRNA που του επιτρέπει να αναγνωρίζει τα mRNA στόχους και να προσδένεται επάνω τους. Η πρόσδεση του miRNA επάνω σε ένα mRNA υπόκειται στον κανόνα της συμπληρωματικότητας των νουκλεοτιδίων, δηλαδή οι ακολουθίες των δύο μορίων πρέπει να είναι συμπληρωματικές σε εκείνη την περιοχή ώστε να επιτευχθεί πρόσδεση. Στη συνήθη περίπτωση η περιοχή πρόσδεσης είναι οι θέσεις 2–8 του miRNA (μήκους περίπου 5 με 7 νουκλεοτίδια στη συνήθη περίπτωση).

  Σημείωση: στα φυτά τα miRNAs παρουσιάζουν σχεδόν απόλυτη συμπληρωματικότητα με το mRNA σε όλο το μήκος τους.

- **Θέση πρόσδεσης (binding site)**: το τμήμα της ακολουθίας ενός mRNA στο οποίο προσδένεται η περιοχή πρόσδεσης ενός miRNA. Μία θέση πρόσδεσης ορίζεται με συντεταγμένες επάνω στην αλληλουχία του mRNA. Ως αρχή της θέσης πρόσδεσης ορίζεται ο αριθμός του νουκλεοτιδίου στο οποίο συμβαίνει το πρώτο «ταίριασμα» μεταξύ του mRNA και του miRNA. Το μήκος της θέσης πρόσδεσης είναι ο αριθμός συνεχόμενων συμπληρωματικών νουκλεοτιδίων μεταξύ mRNA–miRNA. Ο τύπος της θέσης πρόσδεσης ορίζεται διαφορετικά από αλγόριθμο σε αλγόριθμο. Ωστόσο, μία γενική κατεύθυνση είναι ότι ο τύπος μιας θέσης πρόσδεσης καθορίζεται κυρίως από το μήκος της και, ενδεχομένως, από άλλους παράγοντες όπως αν υπάρχουν γειτονικά μεμονωμένα ταιριάσματα γύρω από τη θέση πρόσδεσης.

  Σημειώνουμε εδώ πως, ανεξάρτητα από τη θέση πρόσδεσης, μεταγενέστερα τμήματα του miRNA μπορεί να είναι συμπληρωματικά με μεταγενέστερα τμήματα του mRNA. Σε τέτοιες περιπτώσεις επιτυγχάνεται ξανά πρόσδεση και σε εκείνα τα σημεία. Ωστόσο, ως θέση πρόσδεσης ενός mRNA συνήθως ορίζεται το σημείο στο οποίο θα προσδεθεί η περιοχή πρόσδεσης του miRNA.

Σημαντική σημείωση: Παρ' ότι οι αλληλεπιδράσεις, στην πραγματικότητα, εκδηλώνονται μεταξύ των μεταγράφων (mRNAs) και των miRNAs, στην εργασία, για λόγους που θα εξηγηθούν παρακάτω, προσεγγίσαμε το ζήτημα σε επίπεδο γονιδίων–miRNAs. Αυτή η προσέγγιση δεν είναι βιολογικά λανθασμένη ή αυθαίρετη αφού η συνέπεια μιας αλληλεπίδρασης είναι η καταστολή της έκφρασης του αντίστοιχου γονιδίου. Επομένως, στη συνέχεια της εργασίας οι αλληλεπιδράσεις θα περιγράφονται σε επίπεδο γονιδίου και όχι μεταγράφου ώστε να διευκολύνεται ο αναγνώστης να ακολουθήσει το σκεπτικό της υλοποίησης. Ωστόσο η σημείωση αυτή αποσκοπεί στο να αποφευχθεί ενδεχόμενη παρερμηνεία της φράσης «αλληλεπίδραση γονιδίου–miRNA» και να μη θεωρηθεί πως υπονοείται ότι τα miRNAs αλληλεπιδρούν άμεσα με τα ίδια τα γονίδια, δηλαδή με το DNA.
Επίσης, στη συνέχεια της εργασίας, όταν γίνεται αναφορά σε ένα "miRNA" εννοούμε πάντα τα ώριμα miRNAs.

2.2 Θεωρητικό υπόβαθρο των τεχνολογιών που χρησιμοποιήθηκαν

Ως προς το προγραμματιστικό θεωρητικό υπόβαθρο της εργασίας, θα γίνει αναφορά σε σημαντικές έννοιες του προγραμματισμού διαδικτυακών εφαρμογών, στη σημασία των μεγάλων δεδομένων καθώς και στη σημασία της οπτικοποίησης δεδομένων (data visualisation).

2.2.1 Πλαίσια ανάπτυξης διαδικτυακών εφαρμογών (web application frameworks)

Τα πλαίσια ανάπτυξης εφαρμογών αποτελούν, κατά κάποιον τρόπο, μία μετεξέλιξη της έννοιας της «βιβλιοθήκης». Πλέον, για την ανάπτυξη μιας εφαρμογής, δεν αρκεί να χρησιμοποιεί κάποιος επιμέρους βιβλιοθήκες. Λόγω των αυξημένων απαιτήσεων που έχουν τώρα πια οι εφαρμογές υπολογιστών, απαιτούνται εργαλεία πιο ισχυρά από απλές βιβλιοθήκες πάνω στα οποία να στηριχθεί η ανάπτυξη μιας νέας εφαρμογής. Υπό αυτό το πρίσμα, τα πλαίσια ανάπτυξης εφαρμογών δεν παρέχουν απλώς ομαδοποιημένες

λειτουργίες, όπως μια απλή βιβλιοθήκη, αλλά παρέχουν ένα ολόκληρο πλαίσιο, μια πλατφόρμα, πάνω στην οποία μπορεί να στηριχθεί η ανάπτυξη μιας νέας εφαρμογής.

Τα πλαίσια ανάπτυξης διαδικτυακών εφαρμογών, λοιπόν, είναι εργαλεία που διευκολύνουν την ανάπτυξη ιστοσελίδων, διαδικτυακών εφαρμογών, υπηρεσιών διαδικτύου (web services) και προγραμματιστικών διεπαφών (web APIs).

Τα τελευταία χρόνια τα πλαίσια ανάπτυξης διαδικτυακών εφαρμογών γίνονται όλο και πιο δημοφιλή λόγω της διαρκώς αυξανόμενης χρήσης του διαδικτύου [27, 55] είτε μέσω ιστοσελίδων, είτε μέσω εφαρμογών για κινητές συσκευές (mobile applications), είτε λόγω της διασύνδεσης συσκευών στον παγκόσμιο ιστό (internet of things). Πλέον, η ανάπτυξη μιας ιστοσελίδας σπανίως ξεκινά εκ του μηδενός καθώς, σχεδόν πάντα, οι προγραμματιστές επιλέγουν να βασίσουν τις νέες εφαρμογές επάνω σε ένα από τα διαθέσιμα πλαίσια ανάπτυξης.

Ο βασικός σκοπός των πλαισίων ανάπτυξης είναι να ενθαρρύνουν την επαναχρησιμοποίηση καλώς δομημένου και δοκιμασμένου κώδικα, να τυποποιήσουν επαναλαμβανόμενες εργασίες και, τελικά, να επιτρέψουν στον προγραμματιστή να επικεντρωθεί στην ανάπτυξη της εφαρμογής αυτής καθαυτής [28]. Αυτό το επιτυγχάνουν μέσα από δύο βασικά χαρακτηριστικά:

i) Παρέχουν έτοιμο και δοκιμασμένο κώδικα για ορισμένες κοινές, επαναλαμβανόμενες και τετριμμένες διεργασίες που κάθε ιστοσελίδα χρειάζεται. Ορισμένα παραδείγματα είναι η επαλήθευση αυθεντικότητας χρήστη (user authentication), η διαχείριση συνεδρίας (session handling), ζητήματα ασφαλείας (π.χ. XSS – cross site scripting, SQL injection attacks κ.ά.), συναλλαγές με βάσεις δεδομένων (δημιουργία, ανάγνωση, ενημέρωση και διαγραφή εγγραφών – CRUD operations), η δρομολόγηση της εφαρμογής (application routing), η σχεδίαση ιστοσελίδων με πρότυπα (HTML templating) κ.ά. Τα πλαίσια παρέχουν έτοιμες κλάσεις που υλοποιούν επιμέρους διεργασίες όπως οι παραπάνω έτσι ώστε ο προγραμματιστής να μπορεί κατευθείαν να χρησιμοποιήσει τις κλάσεις αυτές και να υλοποιήσει την αντίστοιχη λειτουργία με, σχεδόν, μηδενικό νέο κώδικα.

ii) Ο προγραμματιστής ακολουθεί το προγραμματιστικό μοτίβο που επιβάλλει το πλαίσιο πάνω στο οποίο βασίζεται, με αποτέλεσμα ο παραγόμενος κώδικας να είναι πιο καλά δομημένος, πιο εύκολα κατανοητός από άλλους προγραμματιστές και πιο εύκολα συντηρήσιμος. Αυτό, φυσικά, δεν αφορά μόνο όσες έτοιμες κλάσεις χρησιμοποιηθούν αλλά εφαρμόζεται και σε όσα νέα τμήματα κώδικα γράψει ο προγραμματιστής

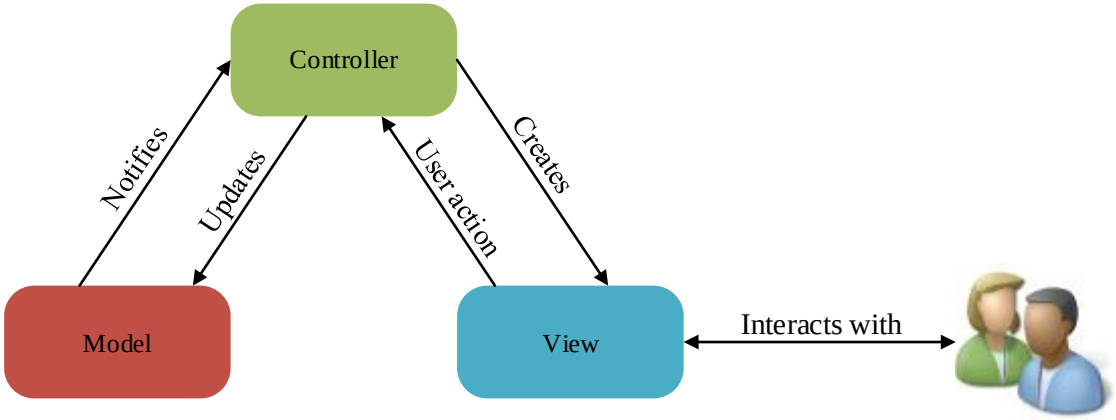
Αν έπρεπε, λοιπόν, να περιγράψει κανείς τα πλαίσια ανάπτυξης εφαρμογών με πολύ απλά λόγια, τότε θα λέγαμε (ίσως ελαφρώς καταχρηστικά) ότι πρόκειται για «μεγάλες και εκτενείς βιβλιοθήκες» που παρέχουν στον προγραμματιστή τα δύο βασικά «βοηθήματα» που προαναφέρθηκαν.

2.2.2 MVC pattern (Model – View – Controller)

Το MVC είναι μία σχεδιαστική μεθοδολογία που χρησιμοποιείται για την ανάπτυξη εφαρμογών που αλληλεπιδρούν με χρήστες, δηλαδή πρόκειται για εφαρμογές που παρέχουν διεπαφή. Προτάθηκε πρώτη φορά ήδη από το 1978 από τον Trygve Reenskaug στα εργαστήρια XEROX Parc ως γενική λύση στο

πρόβλημα του πώς να δομηθεί ένα σύστημα που επιτρέπει στους χρήστες να ελέγχουν μεγάλα και περίπλοκα δεδομένα [29]. Σήμερα, η μεθοδολογία αυτή εφαρμόζεται πρωτίστως για τη σχεδίαση διεπαφών χρήστη διαδικτυακών εφαρμογών (και εκεί, κυρίως, οφείλει τη δημοφιλία της) αλλά και για εφαρμογές υπολογιστών (desktop applications) και κινητών συσκευών (mobile applications).

Η βασική ιδέα είναι ο επιμερισμός των διεργασιών της εφαρμογής σε τρία μέρη έτσι ώστε καθένα από αυτά να λειτουργεί ανεξάρτητα, αυτοτελώς και να εκτελεί απολύτως διακριτές λειτουργίες. Έτσι προκύπτει καλύτερα δομημένος κώδικας, πιο ευανάγνωστος και πιο εύκολα συντηρήσιμος και επεκτάσιμος. Επιπλέον, ο διαχωρισμός αυτός βοηθάει στον κατακερματισμό της πολυπλοκότητας κατά την ανάπτυξη εφαρμογών ενώ καθίσταται πολύ ευκολότερη η ανεξάρτητη δοκιμή των επιμέρους μερών της εφαρμογής (unit testing).



Εικόνα 2.3: το μοντέλο MVC

Οι «αρμοδιότητες» του κάθε μέρους, λοιπόν, είναι οι εξής [30, 31, 32]:

- **Model**: αυτό το κομμάτι της εφαρμογής ορίζει τη δομή αποθήκευσης των δεδομένων και αναλαμβάνει όλη την επικοινωνία με τη βάση δεδομένων της εφαρμογής (αποθήκευση, ανάγνωση, ενημέρωση, διαγραφή). Επενεργεί όταν λάβει σχετική εντολή από το κομμάτι Controller. Στη συνέχεια επιστρέφει στον Controller τα δεδομένα που ανακτήθηκαν.
- **View**: αναλαμβάνει την αλληλεπίδραση με τον χρήστη, δηλαδή την παρουσίαση των δεδομένων σε αυτόν και τη συλλογή των ενεργειών του. Με άλλα λόγια, πρόκειται για τη διεπαφή χρήστη (user interface). Παραλαμβάνει τα δεδομένα από τον Controller ώστε να διαμορφώσει κατάλληλα την παρουσίαση προς το χρήστη. Επίσης, όταν συλλέγει είσοδο από το χρήστη, «ειδοποιεί» κατάλληλα το κομμάτι Controller.
- **Controller**: αποτελεί την υλοποίηση της κυρίως λογικής της εφαρμογής και είναι το ενδιάμεσο «στρώμα» μεταξύ του Model και του View. Ερμηνεύει την είσοδο που παραλαμβάνει από το κομμάτι View και καθορίζει, κατόπιν, τι ενέργειες πρέπει να γίνουν ώστε να προκύψει η κατάλληλη απόκριση προς το χρήστη. Όταν απαιτείται κάποια επικοινωνία με τη βάση δεδομένων, δίνει την ανάλογη εντολή στο κομμάτι Model. Τέλος, όταν έχουν ολοκληρωθεί όλες οι ενέργειες εκ μέρους της εφαρμογής, δημιουργεί ή ενημερώνει κατάλληλα το κομμάτι View ώστε να προκύψει η σωστή απεικόνιση προς τον χρήστη.

Η λογική της ανεξάρτητης και αυτοτελούς λειτουργίας των τριών μερών σημαίνει ότι, σε μία ορθά σχεδιασμένη εφαρμογή, το κάθε μέρος οφείλει να «αγνοεί» την ύπαρξη των άλλων δύο και να λειτουργεί ακόμη και μόνο του. Θα επιχειρήσουμε με ορισμένα παραδείγματα να εξηγήσουμε τι σημαίνει αυτό.

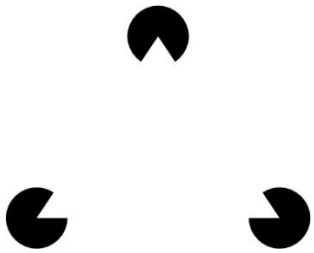
Για παράδειγμα, το κομμάτι Model οφείλει να ανακτήσει ορισμένα δεδομένα από τη βάση δεδομένων και να τα παράσχει, μέσω του Controller, στο κομμάτι View. Αν αυτό δεν συμβεί (π.χ. σφάλμα στην επικοινωνία με τη βάση δεδομένων), η προβολή προς το χρήστη πρέπει και πάλι να μπορέσει να δημιουργηθεί αφού το κομμάτι View οφείλει να είναι ανεξάρτητο και αυτοτελές. Σε μία τέτοια περίπτωση είναι αποδεκτό να λείπει π.χ. ένα γράφημα από την ιστοσελίδα. Όμως, οποιοδήποτε άλλο στοιχείο της ιστοσελίδας **δεν** εξαρτάται από αυτά τα δεδομένα, δεν επιτρέπεται να λείπει (π.χ. μία φόρμα συλλογής εισόδου από τον χρήστη). Σε αυτή την περίπτωση γίνεται σαφές πώς το MVC βοηθάει στην αποσφαλμάτωση (debugging).

Το δεύτερο παράδειγμα καταδεικνύει πώς η εν λόγω μεθοδολογία συμβάλλει τα μέγιστα στη συντήρηση του κώδικα. Έστω ότι γίνονται ορισμένες αλλαγές στη βάση δεδομένων και απαιτείται να αλλάξει η μορφή των ερωτημάτων (SQL queries). Τότε αρκεί να μεταβληθεί αναλόγως το κομμάτι Model και, μάλιστα, κανονικά δεν πρέπει να αλλάξει απολύτως τίποτα στα μέρη View και Controller.

2.2.3 Οπτικοποίηση δεδομένων (data visualisation)

2.2.3.1 Η ανθρώπινη αντίληψη

Ο ανθρώπινος εγκέφαλος λειτουργεί με τέτοιο τρόπο έτσι ώστε να παρατηρεί κανονικότητες μέσα σε τυχαία δεδομένα. Αυτό μπορεί να γίνει καλύτερα αντιληπτό μέσω παραδειγμάτων που σχετίζονται με την όραση.



Εικόνα 2.4: παράδειγμα που δείχνει πώς ο εγκέφαλος προσπαθεί να εντοπίζει κανονικότητες.

Στην παραπάνω εικόνα απεικονίζονται τρεις μαύροι κύκλοι σε καθέναν από τους οποίους λείπει ένα μικρό μέρος. Αυτό, σε συνδυασμό με τη διάταξη των κύκλων, κάνει τον εγκέφαλο του παρατηρητή να συμπεράνει την ύπαρξη ενός τριγώνου. Ωστόσο, στην πραγματικότητα, δεν υπάρχει κανένα τρίγωνο στην εικόνα! Αντίστοιχα, διάφορες έννοιες (π.χ. κατά τη διδασκαλία) γίνονται πολύ πιο εύκολα κατανοητές από τους ανθρώπους αν μία επεξήγηση συνοδεύεται από εικόνες ή σχήματα.

Είναι εύκολα αντιληπτό πως το να επεξεργαστεί κάποιος έναν πίνακα εκατομμυρίων γραμμών είναι αρκετά δύσκολο. Το να εξαχθούν συμπεράσματα, μάλιστα, από τέτοιους όγκους δεδομένων είναι πρακτικά αδύνατο. Οι πίνακες μπορούν να δώσουν ενδείξεις αλλά είναι αρκετά απαιτητικό να καταφέρει κανείς να εντοπίσει κανονικότητες μέσα σε κείμενο. Ακόμη κι όταν κάποιος εργάζεται με πίνακες είναι σχεδόν βέβαιο ότι τους ταξινομεί με κάποια κριτήρια. Έστω κι αυτό, όμως, αποτελεί μία πολύ βασική μορφή

οπτικοποίησης! Αναλογιστείτε ότι, με την ταξινόμηση και μόνο, ο χρήστης έστω και ασυναίσθητα δημιουργεί την εικόνα «επάνω = καλό, κάτω = κακό» ή το αντίστροφο.

Όλα αυτά, λοιπόν, καταδεικνύουν τη σημασία που έχει για τον ανθρώπινο εγκέφαλο η σχηματική αναπαράσταση μιας πληροφορίας. Ο άνθρωπος κατανοεί καλύτερα έννοιες και εξάγει γρηγορότερα συμπεράσματα όταν παρατηρεί εικόνες και σχήματα παρά όταν διαβάζει κείμενο.

2.2.3.2 Η σημασία της οπτικοποίησης δεδομένων

Σε αυτό το σκεπτικό ακριβώς στηρίζεται και η οπτικοποίηση δεδομένων η οποία στοχεύει στο να απλοποιεί την παρουσίαση των δεδομένων, να βοηθά στην καλύτερη κατανόησή τους, να διευκολύνει την εξαγωγή συμπερασμάτων από αυτά και να καταστήσει πιο εύληπτες τις κανονικότητες που «κρύβονται» μέσα στα δεδομένα. Στη σημερινή εποχή, μάλιστα, με τους μεγάλους όγκους δεδομένων (big data), η οπτικοποίηση πληροφοριών αποκτά ακόμη μεγαλύτερη σημασία.

Μέσω της οπτικοποίησης, τα δεδομένα καθίστανται προσβάσιμα και επεξεργάσιμα από όλους και – το κυριότερο – από ανθρώπους που δεν έχουν γνώσεις υπολογιστών. Γι' αυτό είναι ουσιώδες οι εφαρμογές οπτικοποίησης πληροφοριών να προσφέρουν διαδραστικά γραφήματα με τα οποία οι χρήστες να μπορούν να αλληλεπιδρούν. Είναι πολύ σημαντικό για κάποιον που θέλει να μελετήσει δεδομένα, να μπορεί να τα «φιλτράρει» και να δει αμέσως πώς αυτό αλλάζει την απεικόνιση. Το ακόμη σημαντικότερο είναι να μπορέσει να αντιληφθεί τι συμπέρασμα αντιπροσωπεύει στον πραγματικό κόσμο αυτή η αλλαγή της απεικόνισης.

Ένα πολύ βασικό όφελος της οπτικοποίησης δεδομένων που σπάνια επισημαίνεται είναι η ανάδειξη λανθασμένων δεδομένων (corrupt data). Ακριβώς όπως με ένα γράφημα είναι εύκολο να εντοπίσει κανείς μια κανονικότητα, εξίσου εύκολο είναι να εντοπιστεί κάτι που είναι προφανώς παράλογο και δε θα έπρεπε να συμβαίνει. Αυτό μπορεί να είναι ένα πολύ χρήσιμο συμπέρασμα ώστε να εντοπιστεί ένα εσφαλμένο σύνολο δεδομένων που, αλυσιδωτά, μπορεί να οδηγήσει σε εσφαλμένα συμπεράσματα. Αντίστοιχα, καταδεικνύει ενδεχόμενα σφάλματα στη συλλογή των δεδομένων, π.χ. διαδικασία συλλογής δεδομένων, αισθητήρες και μηχανήματα καταγραφής κτλ.

Παρ' όλα αυτά, απλά και μόνο η δημιουργία ενός γραφήματος δεν εγγυάται ότι το γράφημα είναι πράγματι χρήσιμο. Ανάλογα με τη δομή των δεδομένων και ανάλογα με τα συμπεράσματα που αναμένονται, πρέπει να επιλέγεται η κατάλληλη απεικόνιση και να προστίθενται κατάλληλες δυνατότητες αλληλεπίδρασης με τους χρήστες.

3

Σχετικές εργασίες & εργαλεία

Σε αυτό το κεφάλαιο παρουσιάζονται προηγούμενες σχετικές εργασίες και υφιστάμενα εργαλεία βιοπληροφορικής τα οποία χρησιμεύουν στην έρευνα γύρω από τις αλληλεπιδράσεις γονιδίων–miRNAs και απετέλεσαν σημείο αφετηρίας για την παρούσα εργασία.

3.1 Βιολογικές βάσεις δεδομένων

Ένα από τα πιο σημαντικά εργαλεία που έχει συνεισφέρει η βιοπληροφορική στους βιολόγους ερευνητές, είναι οι βάσεις δεδομένων στις οποίες καταγράφεται οργανωμένα η αποκτηθείσα γνώση γύρω από τη βιολογία. Η σημασία των βάσεων δεδομένων γίνεται όλο και πιο εμφανής τα τελευταία χρόνια, αρχής γενομένης από την έρευνα γύρω από το ανθρώπινο γονιδίωμα (Human Genome Project). Οι βιολογικές βάσεις δεδομένων βοηθούν στο να οργανωθούν συναφείς πληροφορίες, να αποθηκευτούν κατάλληλα και να είναι ελεύθερα διαθέσιμες μέσω διαδικτύου. Επίσης, η παροχή προγραμματιστικών διεπαφών από τις βάσεις δεδομένων είναι σημαντικός παράγοντας που βοηθά την αυτοματοποιημένη ανταλλαγή πληροφοριών μεταξύ διαφορετικών βάσεων δεδομένων ενώ διευκολύνει και την ενσωμάτωση πληροφοριών από μία βάση δεδομένων σε μία άλλη [11].

Για την παρούσα εργασία χρησιμοποιήθηκαν ως πηγή πληροφοριών ορισμένες από τις πλέον σημαντικές βιολογικές βάσεις δεδομένων, η Ensembl, η RefSeq και η miRBase, οι οποίες καταγράφουν δεδομένα γύρω από τα γονίδια (οι δύο πρώτες) και τα miRNAs (η τελευταία) αντιστοίχως.

3.1.1 Ensembl

3.1.1.1 Γενικά

Η Ensembl είναι μία βιολογική βάση δεδομένων που καταγράφει το γονιδίωμα ορισμένων σπονδυλωτών οργανισμών. Το εγχείρημα της Ensembl ξεκίνησε το 1999 με αφορμή την πρώτη προσπάθεια καταγραφής του ανθρώπινου γονιδιώματος. Κατά την προσπάθεια καταγραφής του γονιδιώματος έγινε η διαπίστωση πως, λόγω της έκτασής του, η επισημείωσή του (annotation) από ανθρώπους (δηλαδή χειροκίνητα) δε θα επέτρεπε την εύκολη και γρήγορη πρόσβαση των ερευνητών στις πληροφορίες αυτές. Έτσι η Ensembl προέκυψε με σκοπό να δημιουργηθεί ένα εργαλείο για την αυτόματη επισημείωση του ανθρώπινου γονιδιώματος, την ενσωμάτωση και άλλων σχετικών βιολογικών δεδομένων που ήταν ήδη διαθέσιμα και, τέλος, την διάθεση όλων αυτών των συγκεντρωμένων πληροφοριών ανοιχτά στο Διαδίκτυο. Έκτοτε, η Ensembl επεκτάθηκε πολύ πέρα από το ανθρώπινο γονιδίωμα και σήμερα περιέχει 87 οργανισμούς.
Η ιστοσελίδα της Ensembl, εκτός από τα δεδομένα γύρω από τα γονιδιώματα, παρέχει πληθώρα εργαλείων τα οποία αξιοποιούν την εκτενή βάση δεδομένων που έχει διαμορφωθεί. Τα εργαλεία αυτά υποστηρίζουν την έρευνα, μεταξύ άλλων, στα πεδία της συγκριτικής γενετικής (συγκρίσεις μεταξύ γονιδίων διαφορετικών οργανισμών), εξελικτικής βιολογίας και της ρύθμισης της γονιδιακής έκφρασης [33].
Η Ensembl παρέχει τη δυνατότητα να κατεβάσει κανείς τα δεδομένα της είτε με προγραμματιστικό τρόπο (υπάρχουν κατάλληλες προγραμματιστικές διεπαφές – APIs) είτε κατευθείαν ως αρχεία με το εργαλείο BioMart.

3.1.1.2 Μοναδικά αναγνωριστικά Ensembl

Κάθε βιομόριο που είναι καταγεγραμμένο στην Ensembl διαθέτει ένα μοναδικό αναγνωριστικό. Τα μοναδικά αναγνωριστικά της Ensembl ορίζονται από 4 πεδία (τα κενά έχουν προστεθεί για ευκρίνεια) [56]:

ENS xxx b yyyyyyyyyyy . vv

- ENS: εκ του Ensembl
- xxx: δύο ή τρία γράμματα που σηματοδοτούν σε ποιον οργανισμό ανήκει το εν λόγω γονίδιο. Για τα ανθρώπινα βιομόρια αυτό το πεδίο παραλείπεται.
- b: ένα γράμμα το οποίο συμβολίζει το είδος του βιομορίου που καταγράφεται. Η Ensembl καταγράφει 3 είδη βιομορίων: τα γονίδια που συμβολίζονται με “G” (εκ του gene), τα μετάγραφα που συμβολίζονται με “T” (εκ του transcript), τα εξόνια που συμβολίζονται με “E” (εκ του exon) και τις πρωτεΐνες που συμβολίζονται με “P” (εκ του peptide).
- yyy…: 12 ψηφία που αποτελούν τον αύξοντα αριθμό του αναγνωριστικού. Αυτός ο αύξον αριθμός είναι μοναδικός ανά οργανισμό και ανά είδος βιομορίου.
- vv: η έκδοση του γονιδίου. Σημειώνουμε, ωστόσο, πως δεν αναγράφεται πάντα.

Ο ίδιος αύξων αριθμός μπορεί να εμφανίζεται σε διαφορετικούς οργανισμούς και σε διαφορετικούς τύπους βιομορίων. Έτσι, αν τύχει ο ίδιος 12ψήφιος αύξων αριθμός να χρησιμοποιείται σε πολλούς οργανισμούς,

υπάρχει το πεδίο που διακρίνει τον οργανισμό. Αντίστοιχα, επειδή ο ίδιος αύξων αριθμός μπορεί να χρησιμοποιείται για διαφορετικούς τύπους μορίων (π.χ. γονίδιο και πρωτεΐνη), υπάρχει το γράμμα που διαφοροποιεί τον τύπο του βιομορίου.

Παράδειγμα: το γονίδιο TNMD στον άνθρωπο έχει αναγνωριστικό το ENSG000000000005. Παρατηρούμε πως δεν υπάρχει πεδίο που να συμβολίζουν τον οργανισμό. Το ίδιο γονίδιο στον ποντικό έχει αναγνωριστικό ENSMUSG00000031250. Τα τρία γράμματα που συμβολίζουν τον ποντικό είναι τα "MUS" εκ του "Mus Musculus". Παρατηρούμε, επίσης, πως το είδος βιομορίου συμβολίζεται από το γράμμα "G" και στις δύο περιπτώσεις αφού πρόκειται για γονίδιο.

3.1.1.3 Εκδόσεις γονιδίων

Καθώς η μελέτη του γονιδιώματος ενός οργανισμού εξελίσσεται, οι ιδιότητες των γονιδίων καθορίζονται με μεγαλύτερη ακρίβεια και τα αντίστοιχα καταγεγραμμένα δεδομένα διορθώνονται και βελτιώνονται. Για παράδειγμα, σε ένα γονίδιο μπορεί να οριστεί ορθότερη επισημείωση της ακολουθίας του, να εντοπιστεί με μεγαλύτερη ακρίβεια η θέση του επάνω στο γονιδίωμα ή να υπάρξει αλλαγή στα μετάγραφά του. Τέτοιου είδους αλλαγές οδηγούν στην αλλαγή της «έκδοσης» του γονιδίου. Είναι σημαντικό να διευκρινιστεί, όμως, πως αλλαγή της έκδοσης ***δεν*** σημαίνει αλλαγή στη λειτουργία του γονιδίου.

Για ποιο λόγο, όμως, υπάρχουν οι εκδόσεις; Ένας ερευνητής που μελετά ένα γονίδιο μια συγκεκριμένη στιγμή, βασίζεται στις πληροφορίες της Ensembl τη στιγμή αυτή. Αν μετά από κάποιο χρονικό διάστημα οι πληροφορίες αυτές αλλάξουν, ένας άλλος ερευνητής παρ' ότι θα μελετά **το ίδιο γονίδιο**, θα βλέπει διαφορετικές ιδιότητες σε σχέση με τον πρώτο. Το πρόβλημα αυτό γίνεται εντονότερο αν ο δεύτερος ερευνητής προσπαθεί να βασίσει τη μελέτη του επάνω στη μελέτη του προηγούμενου. Εδώ, λοιπόν, ανακύπτει το εξής σημαντικό πρόβλημα: παρ' ότι και οι δύο μελετούν το ίδιο γονίδιο, αναφέρονται σε διαφορετικές ιδιότητες αυτού. Επομένως είναι αναγκαίο να υπάρχει κοινά αποδεκτός τρόπος καταγραφής για το ποιο ακριβώς σύνολο ιδιοτήτων αποδίδεται σε ένα γονίδιο μία δεδομένη χρονική στιγμή.

Και αυτό ακριβώς το πρόβλημα επιλύουν οι εκδόσεις. Κάθε έκδοση του γονιδίου αφορά ένα συγκεκριμένο σύνολο των χαρακτηριστικών που αποδίδονται σε αυτό ώστε οι ερευνητές να γνωρίζουν ανά πάσα στιγμή με ποια έκδοση του γονιδίου δουλεύουν και, συνεπώς, σε ποιο σύνολο ιδιοτήτων αναφέρονται. Αν υπάρξει αλλαγή στην έκδοση ενός γονιδίου, οι ερευνητές μπορούν να γνωρίζουν αμέσως ότι έχουν εντοπιστεί αλλαγές από την προηγούμενη έκδοση και, μάλιστα, ποιες ακριβώς αλλαγές είναι αυτές.

3.1.1.4 Εκδόσεις της Ensembl

Όπως εξηγήθηκε, καθώς η γνώση γύρω από το γονιδίωμα των οργανισμών συνεχώς βελτιώνεται, προκύπτουν αλλαγές στις πληροφορίες των βιομορίων που καταγράφει η Ensembl. Καθώς τα δεδομένα αυτά ανανεώνονται, η Ensembl δημοσιεύει καινούριες εκδόσεις της συνολικής βάσης δεδομένων της, περίπου ανά 3 μήνες, με όλες τις αλλαγές που εντοπίστηκαν μέσα σε αυτό το χρονικό διάστημα. Ο λόγος που γίνεται αυτό είναι για να μπορούν να υπάρχουν κάποια σταθερά σημεία αναφοράς στην καταγραφή των γονιδιωμάτων και να μην βρίσκεται η βάση δεδομένων σε μία διαρκή κατάσταση ενημέρωσης (άρα και μία

ασυνεπή κατάσταση). Έτσι, αν δύο ερευνητές χρησιμοποιούν την ίδια έκδοση της Ensembl ξέρουν πως δουλεύουν επάνω στα ίδια δεδομένα και, αντίστοιχα, αν χρησιμοποιούν διαφορετικές εκδόσεις μπορούν να γνωρίζουν επακριβώς ποιες είναι οι διαφορές στα δεδομένα τους. Αν η βάση δεδομένων ενημερωνόταν συνεχώς και όχι με εκδόσεις, τότε δεν θα υπήρχε τρόπος να γνωρίζουν οι ερευνητές αν εργάζονται επάνω στα ίδια δεδομένα ή όχι και, αν όχι, δεν θα υπήρχε τρόπος να γνωρίζουν ποιες είναι οι όποιες διαφορές. Πρέπει να σημειωθεί, βέβαια, πως μία καινούρια έκδοση της Ensembl δεν συνεπάγεται υποχρεωτικά αλλαγές σε κάθε καταγεγραμμένο βιομόριο.

3.1.2 RefSeq

Η RefSeq είναι μία βάση δεδομένων που καταγράφει καλώς επισημειωμένες, μη επαναλαμβανόμενες ακολουθίες DNA, RNA και πρωτεϊνών για τις οποίες παρέχει εκτενείς αναφορές προς άλλες, εξωτερικές πηγές πληροφοριών. Στη RefSeq καταγράφονται ακολουθίες ευκαρυωτικών οργανισμών, ιών, βακτηρίων, αρχαιοβακτηρίων καθώς και ποικίλλων άλλων οργανισμών. Η RefSeq αποτελεί μία βάση συγκερασμού πληροφοριών γύρω από τις ακολουθίες που καταγράφει, τόσο ως προς την αλληλουχία τους όσο και ως προς τις γενετικές και λειτουργικές τους πληροφορίες.

Η βασική της διαφοροποίηση από άλλες βάσεις δεδομένων για ακολουθίες είναι ότι, προσπαθεί να αποφύγει την πολλαπλή καταγραφή της ίδιας ακολουθίας υπό διαφορετικά ονόματα ή μοναδικά αναγνωριστικά. Ωστόσο αυτό δεν αποκλείει την καταγραφή εναλλακτικών μεταγράφων του ίδιου γονιδίου (alternative spliced transcripts) ή εναλλακτικών μορφών μιας πρωτεΐνης [12].

Όπως και η Ensembl, έτσι και η RefSeq μπορεί να χρησιμοποιηθεί ως πηγή για την ακολουθία γονιδίων και μεταγράφων ενός οργανισμού.

3.1.3 Αντιστοιχίες μεταξύ Ensembl IDs και RefSeq Ids

3.1.3.1 Το πρόβλημα των αντιστοιχίσεων

Πολλές φορές σε διάφορες εργασίες απαιτείται να βρεθεί μία αντιστοιχία μεταξύ των ακολουθιών μίας βάσης δεδομένων με μια άλλη. Το πρόβλημα αυτό είναι αρκετά έντονο μεταξύ Ensembl και RefSeq καθώς και οι δύο παρέχουν δεδομένα για γονίδια και μετάγραφα ενώ χρησιμοποιούν η καθεμία τα δικά της μοναδικά αναγνωριστικά. Σε κάποιες περιπτώσεις είναι αναγκαίο να βρεθεί ένα μετάγραφο που ορίζεται στη μία σε ποιο μετάγραφο της άλλης αντιστοιχεί. Η δυσκολία προκύπτει διότι υπεισέρχονται θέματα σχετικά με την επισημείωση των ακολουθιών, ποιες συμβάσεις ακολουθεί κάθε πηγή, πώς λήφθηκαν τα δεδομένα της ακολουθίας κτλ.

Και οι δύο αυτές πηγές παρέχουν κάποιες αντιστοιχίες. Ωστόσο και πάλι προκύπτει πρόβλημα διότι:

- Η κάθε πηγή ορίζει τις αντιστοιχίες με δικό της τρόπο.
- Οι αντιστοιχίες δεν είναι ένα προς ένα. Π.χ. παρατηρήθηκε πως σε πολλές περιπτώσεις ένα μετάγραφο της RefSeq αντιστοιχίζεται σε περισσότερα από ένα μετάγραφα της Ensembl, ανεξαρτήτως του ποια από τις δύο χρησιμοποιείται ως πηγή των αντιστοιχίσεων.

- Σπάνια οι δύο πηγές συμφωνούν στις αντιστοιχίες που προτείνουν.

3.1.3.2 CCDS project

Η CCDS (https://www.ncbi.nlm.nih.gov/projects/CCDS/CcdsBrowse.cgi) είναι μία βάση δεδομένων που δημιουργείται ακριβώς με αυτό τον σκοπό: να ορισθούν αντιστοιχίες κοινά αποδεκτές και από τις δύο επιμέρους πηγές. Έταιρη βασική επιδίωξη αποτελεί, επίσης, να υπάρχει συμφωνία ως προς την επισημείωση των ακολουθιών από τις επιμέρους πηγές. Ωστόσο, ακόμη και η CCDS, δεν καταφέρνει πάντα να ορίσει μία «1 προς 1» αντιστοιχία μεταξύ μεταγράφων των δύο πηγών.

3.1.4 miRBase

3.1.4.1 Γενικά

Η miRBase είναι βιολογική βάση δεδομένων στην οποία καταγράφονται miRNAs, οι ακολουθίες τους καθώς και άλλες σχετικές πληροφορίες. Στη βάση καταγράφονται δεδομένα τόσο των πρόδρομων όσο και των ώριμων miRNAs [34].

Λόγω της έντονης ερευνητικής δραστηριότητας γύρω από τα miRNAs, καινούρια miRNAs ανακαλύπτονται συνεχώς ενώ οι μέθοδοι επισημείωσης των ακολουθιών τους ποικίλλουν. Υπό αυτό το πρίσμα θεωρείται αποφασιστικής σημασίας η ύπαρξη ενός συνόλου δεδομένων υψηλής ποιότητας, κοινά αποδεκτού ως έγκυρου, γύρω από τα miRNAs. Για παράδειγμα, ένα πολύ βασικό πρόβλημα που έπρεπε να αντιμετωπιστεί παλαιότερα και το οποίο προέκυψε λόγω των διαρκώς αυξανόμενων νέων γνώσεων γύρω από τα miRNAs και τη λειτουργία τους, ήταν να καθοριστεί ένας σταθερός και συνεπής τρόπος ονοματολογίας των καινούριων miRNAs. [13]

Δημιουργήθηκε το 2002 ενώ η τελευταία ανανέωσή της έγινε το 2014 με την έκδοση 21. Σε αυτή την έκδοση καταγράφονται 28,645 πρόδρομα miRNAs από τα οποία προκύπτουν 35,828 ώριμα miRNAs σε 223 διαφορετικούς οργανισμούς. Τα δεδομένα της είναι ελεύθερα διαθέσιμα στο Διαδίκτυο [34].

3.1.4.2 Ονοματολογία των miRNAs

Η δομή των ονομάτων των miRNAs είναι η εξής [35]:

xxx – mi[r, R] – yyy – [3, 5]p

- xxx: το πεδίο αυτό αποτελείται από τρία γράμματα και συμβολίζει τον οργανισμό. Χρησιμοποιούνται οι τριγράμματοι κωδικοί KEGG όπως “hsa” για τον άνθρωπο, “mmu” για τον ποντικό κτλ.
- Το δεύτερο πεδίο περιέχει
  i) **mir** αν πρόκειται για πρόδρομο miRNA
  ii) **miR** αν πρόκειται για ώριμο miRNA

- yyy: το πεδίο αυτό περιέχει έναν αριθμό ο οποίος αποδίδεται στα miRNAs κατά σειρά όταν ονοματίζονται. Π.χ. αν το τελευταίο miRNA που έχει ανακαλυφθεί στον ποντικό είναι το 412 το επόμενο miRNA που θα ανακαλυφθεί θα αριθμηθεί 413. Ωστόσο σε αυτό το πεδίο μπορούν να υπάρχουν διαφοροποιήσεις κατά περίπτωση που θα εξηγηθούν παρακάτω.
- Το τελευταίο πεδίο εμφανίζεται μόνο σε ορισμένα ώριμα miRNAs και σηματοδοτεί από ποιο άκρο του πρόδρομου miRNA προήλθε το ώριμο miRNA. Ωστόσο υπάρχουν και ώριμα miRNAs χωρίς αυτό το πεδίο, π.χ. hsa–miR–2110.
    i) 5p: εκ του "5 prime" και σημαίνει ότι το ώριμο miRNA προήλθε από το 5΄ άκρο.
    ii) 3p: εκ του "3 prime", αντίστοιχα.

Ωστόσο υπάρχουν δύο μεμονωμένες εξαιρέσεις στους κανόνες αυτούς. Τα miRNAs let-7 και lin-4 έχουν διατηρήσει το όνομά τους χωρίς το δεύτερο πεδίο να αναγράφει "mir" ενώ και ενδεχόμενα νέα miRNAs των οικογενειών αυτών συνεχίζουν να λαμβάνουν αυτό το διαφορετικό όνομα. Αυτό γίνεται για ιστορικούς λόγους καθότι αυτά ήταν τα πρώτα miRNAs που ανακαλύφθηκαν ανοίγοντας αυτό το νέο πεδίο. Τα miRNAs αυτά ανιχνεύθηκαν πρώτη φορά στο νηματώδες Caenorhabditis elegans [2].

Παραδείγματα: hsa–mir–32, πρόδρομο ανθρώπινο miRNA. hsa–miR–32–5p, το ώριμο miRNA που προκύπτει απ' το 5΄ άκρο του προηγούμενου πρόδρομου miRNA. mmu–mir–32, πρόδρομο miRNA ποντικού με ίδια ακολουθία με το αντίστοιχο πρόδρομο στον άνθρωπο.

3.1.4.3 Διαφοροποιήσεις στην αρίθμηση miRNAs

Ορισμένες φορές τα ονόματα των miRNAs παρουσιάζουν διαφοροποιήσεις στο τρίτο πεδίο [35].

- (με βάση το προηγούμενο παράδειγμα) Εάν το νέο miRNA του ποντικού είναι παρεμφερές με το miRNA cel-miR-250 (το οποίο ανήκει στον οργανισμό Caenorhabditis elegans) τότε το νέο miRNA του ποντικού πιθανότητα θα ονομαστεί mmu-mir-250 και όχι mmu-mir-413 όπως θα ήταν το αναμενόμενο.
- Διαφορετικά πρόδρομα miRNAs από τα οποία προέρχονται σχεδόν πανομοιότυπα ώριμα miRNAs παίρνουν ονόματα όπως hsa–mir–138–1, hsa–mir–138–2.
- Διαφορετικά πρόδρομα miRNAs των οποίων οι ώριμες ακολουθίες μοιάζουν αλλά σε μικρότερο βαθμό λαμβάνουν ένα επίθεμα με γράμματα. Π.χ. hsa–mir–130a, hsa–mir–130b.

3.1.4.4 Μοναδικά αναγνωριστικά miRBase (miRBase accession numbers)

Αρχικά πρέπει να επισημανθεί ότι ο αριθμός στο τρίο πεδίο των ονομάτων των miRNAs **δε σχετίζεται** με τον αριθμό στα μοναδικά αναγνωριστικά τους. Τα μοναδικά αναγνωριστικά της miRBase ακολουθούν την εξής δομή:

MI (MAT) xxxxxxx

- **MI**: τυπικό πρόθεμα κάθε αναγνωριστικού. Υπάρχει πάντα.
- **MAT**: εκ του "mature". Εισάγεται μόνο για τα ώριμα miRNAs και συμβολίζει αν το μοναδικό αναγνωριστικό αντιστοιχεί σε πρόδρομο ή ώριμο miRNA.
- **xxxxxxx:** επταψήφιος ακέραιος αύξων αριθμός του αναγνωριστικού. Μπορεί να συναντάται ο ίδιος αριθμός σε πρόδρομα και ώριμα miRNAs.

Εδώ παρατηρείται πως δεν υπάρχει διαφοροποίηση στα μοναδικά αναγνωριστικά ανάλογα με τον οργανισμό, όπως συμβαίνει με τα αναγνωριστικά της Ensembl.

Παραδείγματα: MI0000077 που αντιστοιχεί στο πρόδρομο miRNA hsa–mir–21. MIMAT0000077 που αντιστοιχεί στο ώριμο miRNA hsa–miR–22–3p.

3.2 Συλλογή βιολογικών δεδομένων

3.2.1 Αρχεία βιολογικών δεδομένων

Οι διάφορες βιολογικές βάσεις δεδομένων διαθέτουν διάφορους τρόπους για να κάνουν διαθέσιμα τα δεδομένα τους προς κατέβασμα. Σε πολλές περιπτώσεις διατίθενται εργαλεία που βοηθούν τους χρήστες να καθορίσουν με ακρίβεια ποιο (υπο)σύνολο των δεδομένων μιας βάσης δεδομένων θέλουν να αποκτήσουν. Ένα τέτοιο παράδειγμα είναι το BioMart το οποίο αναλύεται στη συνέχεια.

Ωστόσο, ανεξαρτήτως από το αν παρέχονται τέτοια εργαλεία, οι περισσότερες βιολογικές βάσεις δεδομένων επιλέγουν να διαθέτουν τα δεδομένα τους και ως απλά αρχεία κειμένου. Το βασικό πλεονέκτημα των αρχείων κειμένου είναι ότι είναι εύκολα αναγνώσιμα από οποιονδήποτε επεξεργαστή κειμένου και άρα ο ερευνητής μπορεί άμεσα να διαβάσει και να αξιοποιήσει τα δεδομένα. Επίσης, τα έτοιμα αυτά αρχεία είναι πολύ χρήσιμα αν ο ερευνητής χρειάζεται μεγάλο μέρος των δεδομένων οπότε δεν εξυπηρετούν τα ανωτέρω αναφερθέντα εργαλεία. Το προφανές μειονέκτημα, ωστόσο, είναι ότι η απλή καταγραφή των πληροφοριών σε «ελεύθερο κείμενο» σαφώς δε διευκολύνει τις αναλύσεις.

Για αυτόν ακριβώς το λόγο έχουν δημιουργηθεί ειδικές «διαμορφώσεις» αρχείων (file formats) οι οποίες καθορίζουν μία συγκεκριμένη δομή έτσι ώστε οι πληροφορίες να καταγράφονται μεν σε ένα απλό αρχείο κειμένου αλλά αυτό να γίνεται συστηματικά και οργανωμένα. Η συστηματική καταγραφή των πληροφοριών αυξάνει την αναγνωσιμότητα του αρχείου ενώ το επιπλέον κέρδος είναι ότι καθιστά δυνατή την επεξεργασία των αρχείων αυτών από προγράμματα υπολογιστών. Αυτή ακριβώς η δυνατότητα θα αξιοποιηθεί αργότερα και σε αυτή την εργασία.

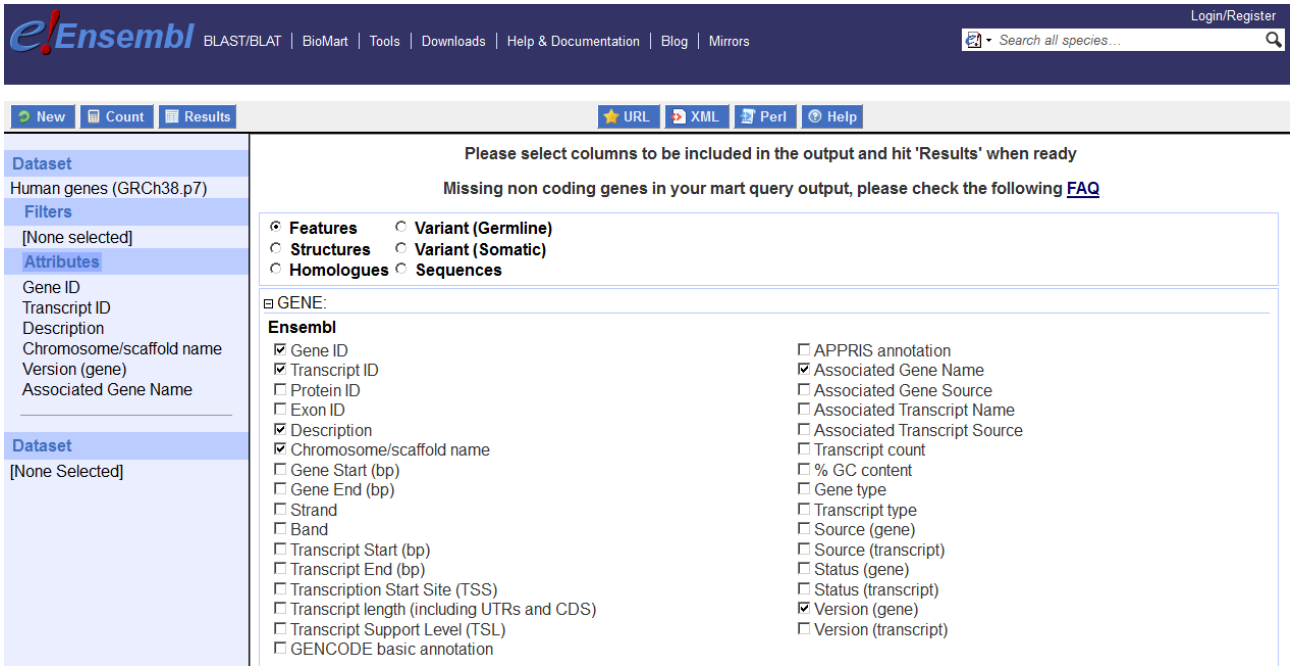
Τέτοια ειδικά διαμορφωμένα αρχεία κειμένου χρησιμοποιούνται εκτενώς για την καταγραφή πληροφοριών επισημειωμένων ακολουθιών. Μερικές πολύ δημοφιλείς διαμορφώσεις αρχείων για ακολουθίες είναι τα: EMBL flat–file format, FASTA, FASTQ, GenBank καθώς και άλλες.

3.2.2 BioMart

Το BioMart είναι ένα εργαλείο που παρέχει μία πολύ φιλική διεπαφή χρήστη με σκοπό να διευκολύνει την εξαγωγή πληροφοριών από βάσεις δεδομένων χωρίς να απαιτείται ο χρήστης να γνωρίζει προγραμματισμό ή τη δομή της βάσης δεδομένων. Πρόκειται για ένα πολύ δημοφιλές και διαδεδομένο εργαλείο που χρησιμοποιείται εκτενώς σε πολλές βιολογικές βάσεις δεδομένων [36, 37].

Κάθε φορά που ο χρήστης πραγματοποιεί μια αναζήτηση στο BioMart, δημιουργείται ένας πίνακας με τα αποτελέσματα της αναζήτησης. Ο χρήστης επιλέγει ποια πεδία ιδιοτήτων επιθυμεί να συμπεριλάβει στον πίνακα (δηλαδή τις στήλες του πίνακα) ενώ μπορεί να ορίσει φίλτρα που περιορίζουν ποια αποτελέσματα θα συμπεριληφθούν. Τέλος, ο πίνακας που προκύπτει εισάγεται σε ένα αρχείο HTML, CSV, TSV ή XLS, ανάλογα με την επιλογή του χρήστη.

Η Ensembl παρέχει αυτό το εργαλείο στην ιστοσελίδα της για να μπορούν οι χρήστες να αντλούν γονιδιακά δεδομένα από τη βάση δεδομένων της. Το εργαλείο αυτό αξιοποιήθηκε και από εμάς, όπως θα περιγραφεί στο κεφάλαιο 6.



Εικόνα 3.1: η οθόνη του BioMart στην ιστοσελίδα της Ensembl

3.2.3 EMBL flat–file format

[38] Καθώς η miRBase διαθέτει τα δεδομένα των miRNAs σε αρχεία "EMBL flat–file format" επιλέχθηκε να παρουσιαστεί αυτή τη διαμόρφωση με κάποια λεπτομέρεια. Τα αρχεία αυτά χρησιμοποιούνται για την καταγραφή δεδομένων για ακολουθίες μορίων DNA και RNA. Για την καταγραφή των ακολουθιών καθώς

και των επιμέρους ιδιοτήτων της καθεμίας, έχει οριστεί μία συγκεκριμένη δομή την οποία πρέπει να ακολουθεί ένα τέτοιο αρχείο. Τα αρχεία "EMBL flat–file" περιέχουν κείμενο σε απλή γλώσσα, άρα είναι κατανοητά από τους ανθρώπους, ενώ η δομή τους είναι συστηματοποιημένη και διαρθρωμένη με τέτοιο τρόπο ώστε να είναι εφικτή η επεξεργασία τους από υπολογιστή.

Ένα αρχείο οργανώνεται σε εγγραφές όπου κάθε εγγραφή περιλαμβάνει όλες τις πληροφορίες που σχετίζονται με μια συγκεκριμένη ακολουθία. Για καλύτερη κατανόηση, παρατίθεται στην επόμενη σελίδα ένα παράδειγμα εγγραφής από το αρχείο της miRBase. Καθώς κάθε αρχείο μπορεί να περιλαμβάνει δεδομένα για πολλές ακολουθίες, αποτελείται από πολλές τέτοιες εγγραφές, τη μία κάτω από την άλλη.

Όπως φαίνεται στο παράδειγμα, κάθε εγγραφή οργανώνεται σε γραμμές ενώ κάθε γραμμή ξεκινάει με έναν κωδικό. Ο κωδικός αποτελείται από δύο γράμματα και σηματοδοτεί τον «τύπο» της γραμμής. Ανάλογα με τον τύπο της, μια γραμμή μπορεί να περιέχει μόνο ένα είδος πληροφορίας ενώ ο τύπος ορίζει και τη σύνταξή της γραμμής. Η σειρά με την οποία εμφανίζονται οι γραμμές είναι συγκεκριμένη έτσι ώστε οι πληροφορίες να καταγράφονται με την ίδια σειρά σε όλες τις εγγραφές.

Ανάλογα με τον κωδικό, λοιπόν, ένα πρόγραμμα που διαβάζει τέτοια αρχεία μπορεί να γνωρίζει επακριβώς τι σύνταξη έχει αυτή η γραμμή. Αυτό ακριβώς το χαρακτηριστικό των αρχείων "EMBL flat–file" είναι που τα καθιστά εύκολα επεξεργάσιμα από υπολογιστή.

Υπάρχουν πολλοί τύποι γραμμών κάποιοι εκ των οποίων υπάρχουν πάντοτε σε μία εγγραφή ενώ, άλλοι, υπάρχουν μόνο αν η αντίστοιχη πληροφορία υπάρχει για μια εγγραφή. Οι σημαντικότεροι τύποι γραμμών είναι οι εξής (και υπάρχουν σε κάθε εγγραφή):

- **ID (identification)**: η εναρκτήρια γραμμή κάθε εγγραφής. Περιέχει ορισμένα βασικά στοιχεία της ακολουθίας όπως το όνομα, το είδος βιομορίου (DNA/RNA), το μήκος της, τον οργανισμό κτλ.
- **AC (accession number)**: το μοναδικό αναγνωριστικό της ακολουθίας σύμφωνα με την πηγή που διαθέτει το αρχείο. Στην περίπτωση της miRBase αυτό είναι το miRBase accession number. Ωστόσο, κατά περίπτωση, μπορεί να αναγράφονται πολλαπλά αναγνωριστικά της ακολουθίας και σύμφωνα με άλλες πηγές.
- **DE (description)**: η αναλυτική περιγραφή της ακολουθίας.
- **R**[A, T, L, N, C, P, X, G] **(reference)**: όσοι κωδικοί ξεκινούν από "R" συμβολίζουν γραμμές οι οποίες αφορούν εξωτερικές αναφορές, π.χ. δημοσιεύσεις. Οι αναφορές αυτές αποτελούν την πηγή των πληροφοριών της εγγραφής. Κάθε κωδικός αφορά ένα συγκεκριμένο χαρακτηριστικό της εξωτερικής αναφοράς, όπως συγγραφέας, τίτλος, τοποθεσία κτλ.
- **CC (comment)**: γραμμή που περιέχει σχόλιο. Το σχόλιο μπορεί να αφορά την ίδια την ακολουθία είτε να αποτελεί διευκρίνιση κάποιου δεδομένου άλλης γραμμής. Περιέχει ελεύθερο κείμενο και δεν ακολουθεί κάποια συγκεκριμένη σύνταξη.
- **FH (feature table header)**: χρησιμεύει στο να διευκολύνει τον άνθρωπο αναγνώστη να αντιληφθεί τι σημαίνουν τα πεδία που ακολουθούν στις επόμενες γραμμές. Από τον υπολογιστή μπορεί να αγνοηθεί εντελώς. Καθορίζουν τις στήλες ενός «πίνακα χαρακτηριστικών» που θα ακολουθήσει.

```
ID   hsa-let-7a-1      standard; RNA; HSA; 80 BP.
XX
AC   MI0000060;
XX
DE   Homo sapiens let-7a-1 stem-loop
XX
RN   [1]
RX   PUBMED; 11679670.
RA   Lagos-Quintana M, Rauhut R, Lendeckel W, Tuschl T;
RT   "Identification of novel genes coding for small expressed RNAs";
RL   Science. 294:853-858(2001).
XX
CC   let-7a-3p cloned in [6] has a 1 nt 3' extension (U), which
CC   is incompatible with the genome sequence.
XX
FH   Key          Location/Qualifiers
FH
FT   miRNA        6..27
FT                /accession="MIMAT0000062"
FT                /product="hsa-let-7a-5p"
FT                /evidence=experimental
FT                /experiment="cloned [1-3,5-8], Northern [1], Illumina [9]"
XX
SQ   Sequence 80 BP; 21 A; 15 C; 19 G; 0 T; 25 other;
ugggaugagg uaguagguug uauaguuuua gggucacacc caccacuggg agauaacuau       60
acaaucuacu gucuuuccua                                                   80
//
```

Εικόνα 3.2: ένα τμήμα αρχείου με EMBL flat–file διαμόρφωση

- **FT (feature table data)**: το κυρίως μέρος του «πίνακα χαρακτηριστικών» το οποίο περιέχει συγκεκριμένες ιδιότητες της ακολουθίας. Ανάλογα με το βιομόριο, περιλαμβάνονται ιδιότητες που θεωρούνται σημαντικές για τον προσδιορισμό του ή την περιγραφή της λειτουργίας του. Για παράδειγμα, περιοχές ιδιαίτερου ενδιαφέροντος επάνω στο βιομόριο αναγράφονται εδώ.
- **SQ**: η αρχική γραμμή της ακολουθίας με τα γενικά χαρακτηριστικά της όπως το μήκος της και την καταμέτρηση των βάσεων. Ωστόσο δεν περιέχει καθόλου από την αλληλουχία βάσεων.
- [κενό]: με κενό ξεκινούν οι γραμμές που περιέχουν την αλληλουχία βάσεων της ακολουθίας και υποχρεωτικά ακολουθούν αμέσως μετά τη γραμμή με κωδικό SQ. Οι βάσεις αναγράφονται σε ομάδες των 10 ενώ κάθε γραμμή περιέχει το πολύ 6 πλήρεις ομάδες. Στο τέλος της γραμμής αναγράφεται η θέση (ο αριθμός) της τελευταίας βάσης του τμήματος της αλληλουχίας που περιέχεται σε αυτή τη γραμμή.

- **XX**: κενή γραμμή που τοποθετείται σε κατάλληλα σημεία μιας εγγραφής ώστε να διαχωρίζει τις υπόλοιπες γραμμές σε ομάδες με συναφείς πληροφορίες. Το XX χρησιμοποιείται για να διαχωρίζει τις κενές γραμμές ομαδοποίησης από τον «κενό κωδικό» που συμβολίζει γραμμές ακολουθίας.
- / /: τερματική γραμμή κάθε εγγραφής. Χρησιμεύει μόνο για να διαχωρίζει τις εγγραφές και, ταυτόχρονα, να εισάγει μία κενή γραμμή ανάμεσά τους (διαφορετικά θα μπορούσε απλώς να σηματοδοτείται η έναρξη μιας νέας εγγραφής με μια γραμμή ID).

3.2.4 Το αρχείο της miRBase

Όπως φαίνεται και στο παράδειγμα, στο αρχείο της miRBase κάθε εγγραφή αφορά ένα πρόδρομο miRNA. Τα ώριμα miRNAs δεν αποτελούν ξεχωριστές εγγραφές αλλά απαριθμούνται μέσα στην εγγραφή του πρόδρομου miRNA από το οποίο προκύπτουν. Η καταγραφή των ώριμων miRNAs γίνεται στις FT γραμμές μιας εγγραφής. Για κάθε ώριμο miRNA που καταγράφεται σημειώνονται, κατά σειρά, οι εξής οι ιδιότητές του:

- Μία γραμμή που σηματοδοτεί ένα νέο ώριμο miRNA. Η γραμμή πρώτα αναγράφει τη λέξη "mirna" και μετά τις θέσεις επάνω στο πρόδρομο miRNA όπου ξεκινά και σταματάει το ώριμο miRNA.
- Το μοναδικό αναγνωριστικό του **ώριμου miRNA.**
- Το όνομα του ώριμου miRNA.
- Αν η ύπαρξη του εν λόγω ώριμου miRNA έχει επαληθευτεί πειραματικώς και, αν ναι, από ποια πειράματα.
- Άλλες επιπλέον πληροφορίες που έχουν, πιθανώς, προσδιοριστεί.

Ιδιαίτερη προσοχή χρειάζεται στη διαφοροποίηση των μοναδικών αναγνωριστικών μεταξύ των πρόδρομων και των ώριμων miRNAs. Όπως εξηγήθηκε, κάθε εγγραφή αποτελεί καταγραφή ενός πρόδρομου miRNA. Άρα η **γραμμή AC** μιας εγγραφής θα αναφέρει το αναγνωριστικό του αντίστοιχου **πρόδρομου miRNA**. Τα μοναδικά αναγνωριστικά των **ώριμων miRNAs** αναφέρονται στη δεύτερη απ' τις **FT γραμμές** που περιγράφουν το εν λόγω ώριμο miRNA.
Σημειώνουμε, ακόμη, πως το αρχείο περιλαμβάνει όλα τα miRNAs, όλων των οργανισμών.

3.3 Αλγόριθμοι πρόβλεψης αλληλεπιδράσεων

3.3.1 Τι είναι οι αλγόριθμοι πρόβλεψης αλληλεπιδράσεων και πώς λειτουργούν

Όπως εξηγήθηκε στη εισαγωγή, η μελέτη αλληλεπιδράσεων με τους κλασσικούς εργαστηριακούς τρόπους είναι αρκετά χρονοβόρα και δύσκολη. Αυτό συμβαίνει επειδή τόσο τα μετάγραφα που πρέπει να εξεταστούν όσο και τα miRNAs είναι πάρα πολλά και επομένως το πλήθος των πιθανών αλληλεπιδράσεων προς εξέταση είναι ιδιαίτερα μεγάλο. Υπό αυτό το πρίσμα αναπτύχθηκαν οι αλγόριθμοι πρόβλεψης αλληλεπιδράσεων με σκοπό να διευκολύνουν τον εντοπισμό πιθανών ζευγών μεταγράφων–miRNAs.

Οι αλγόριθμοι πρόβλεψης είναι αλγόριθμοι που εκτελούνται από υπολογιστές. Ως είσοδο λαμβάνουν την αλληλουχία των μεταγράφων και των miRNA που θα μελετηθούν καθώς και το μοντέλο πρόβλεψης που έχει αναπτυχθεί. Κατόπιν εφαρμόζουν το μοντέλο πρόβλεψης επάνω σε αυτά τα μετάγραφα και τα miRNAs έτσι ώστε:

i) να εντοπίσουν πιθανές αλληλεπιδράσεις,
ii) να αποδώσουν μία βαθμολογία σε κάθε πιθανή αλληλεπίδραση,

και να δώσουν τα προκύπτοντα αποτελέσματα ως έξοδο.

Επιχειρώντας μια πολύ βασική εξήγηση του πώς λειτουργούν οι αλγόριθμοι πρόβλεψης, αναφέρουμε ότι αρχικά εξετάζουν τη συμπληρωματικότητα μεταξύ των miRNAs και των μεταγράφων–στόχων. Εάν η περιοχή πρόσδεσης του miRNA (seed region) παρουσιάζει συμπληρωματικότητα με κάποιες περιοχές ενός μεταγράφου τότε υπάρχει πιθανότητα αλληλεπίδρασης σε αυτές τις συμπληρωματικές περιοχές. Πέραν της συμπληρωματικότητας, βέβαια, ο κάθε αλγόριθμος εξετάζει και διάφορους άλλους παράγοντες για να αποφανθεί αν μία περιοχή του μεταγράφου αποτελεί όντως πιθανή περιοχή πρόσδεσης (binding site) ή όχι. Μερικοί τέτοιοι παράγοντες είναι [3] κατά πόσον το συγκεκριμένο τμήμα της αλληλουχίας του μεταγράφου διατηρείται και σε άλλους οργανισμούς (site conservation), το είδος των βάσεων στην περιοχή πρόσδεσης, τα χαρακτηριστικά των γειτονικών προς την περιοχή πρόσδεσης περιοχών, αν η περιοχή πρόσδεσης βρίσκεται στο 5΄ ή στο 3΄ άκρο του μεταγράφου κ.ά. Έτσι, τελικά, κάθε αλγόριθμος αποδίδει μία βαθμολογία στις πιθανές περιοχές πρόσδεσης που εντοπίζει.

3.3.2 Οι αλγόριθμοι που χρησιμοποιήθηκαν

Οι τρεις αλγόριθμοι πρόβλεψης αλληλεπιδράσεων που χρησιμοποιήθηκαν σε αυτή την εργασία είναι ο DIANA–microT–CDS [14], ο TargetScan [15] και ο MirTarget [16]. Οι αλγόριθμοι αυτοί επελέγησαν καθότι αποτελούν τρεις απ' τους πιο ευρέως χρησιμοποιούμενους ενώ, ειδικά οι δύο πρώτοι, θεωρούνται από τους πλέον αξιόπιστους αλγορίθμους πρόβλεψης με αρκετές αναφορές σε δημοσιεύσεις, όπως στις [3, 6, 7]. Ο δεύτερος λόγος για τον οποίο τους επιλέξαμε είναι ότι πρόκειται για αλγορίθμους που έχουν ανανεώσει τα αποτελέσματά τους πολύ πρόσφατα (λίγο παλαιότερα μόνο ο MirTarget), κάτι που ήταν σημαντικό για την καλύτερη διαχείριση των αναφορών τους προς Ensembl και miRBase.

3.3.2.1 DIANA–microT–CDS

Ο αλγόριθμος αυτός αναπτύχθηκε πρώτη φορά το 2004 [17], είναι από τους πρώτους αλγορίθμους που αναπτύχθηκαν και αυτή τη στιγμή βρίσκεται στην 5η έκδοσή του. Αποτελεί μία ελληνική προσπάθεια στην έρευνα για τα miRNAs ενώ η συντήρηση και η βελτίωσή του γίνεται από το DIANA Lab του Ινστιτούτου Πληροφοριακών Συστημάτων «Αθηνά». Στην ιστοσελίδα του εργαστηρίου μπορεί να βρει κανείς πληθώρα άλλων εργαλείων σχετικών με την έρευνα γύρω από τα miRNAs.

3.3.2.2 TargetScan

Ο TargetScan είναι ο πρώτος αλγόριθμος που αναπτύχθηκε για την πρόβλεψη αλληλεπιδράσεων και εμφανίστηκε το 2003 [18]. Έκτοτε, το μοντέλο των προβλέψεών του έχει επεκταθεί και αναθεωρηθεί αρκετές φορές. Αυτή τη στιγμή βρίσκεται στην έκδοση 7.1 για τον άνθρωπο και τον ποντικό και στην έκδοση 6.2 για την κοινή μύγα (Drosophila melanogaster) και τους οργανισμούς Caenorabditis elegans και Danio rerio.

3.3.2.3 MirTarget

Ο MirTarget αναπτύχθηκε πρώτη φορά το 2008 [19] και βρίσκεται αυτή τη στιγμή στην 5η έκδοσή του. Τελευταία ανανέωση των αποτελεσμάτων του έγινε τον Αύγουστο του 2014 και είναι ο πιο παλιά ανανεωμένος από τους αλγορίθμους που χρησιμοποιήθηκαν σε αυτή την εργασία. Στην ιστοσελίδα του, πέραν των αποτελεσμάτων, παρέχονται επίσης και διάφορα εργαλεία βασισμένα στον αλγόριθμο. Το πιο ενδιαφέρον εξ αυτών είναι η δυνατότητα να καταχωρίσει ο χρήστης μια αλληλουχία βάσεων (η οποία να αντιπροσωπεύει είτε γονίδιο είτε miRNA) και ο αλγόριθμος να εκτελεστεί επάνω σε αυτή την ακολουθία παρέχοντας άμεσα προβλέψεις πιθανών αλληλεπιδράσεων.

3.4 Εργαλεία σύγκρισης αλγορίθμων πρόβλεψης

Κατά την έρευνα που πραγματοποιήθηκε σε αυτή την εργασία βρέθηκαν μόνο δύο εργαλεία τα οποία να υποστηρίζουν τη συγκριτική παρουσίαση αποτελεσμάτων από πολλούς αλγορίθμους. Αυτά είναι τα miRWalk 2.0 (http://zmf.umm.uni-heidelberg.de/apps/zmf/mirwalk2/index.html) και Tools4MiRs (https://tools4mirs.com/software/target_prediction/). Και στα δύο αυτά εργαλεία μπορεί κανείς να αναζητήσει αλληλεπιδράσεις από τους επιμέρους επιλεγμένους αλγορίθμους και να συγκρίνει τα αποτελέσματά τους. Στην αρχική σελίδα του Tools4MiRs, συγκεκριμένα, είναι συγκεντρωμένα πολλά εργαλεία σχετικά με τα miRNAs οπότε μπορεί να χρησιμοποιηθεί και ως ευρετήριο άλλων συναφών εργαλείων.

3.4.1 miRWalk 2.0

Εκτός από τις συγκρίσεις των αλγορίθμων, το εργαλείο αυτό συνδυάζει τις προβλέψεις των αλγορίθμων ώστε να εξάγει και συνδυαστικά αποτελέσματα. Τα αποτελέσματα που παρέχονται είναι ομαδοποιημένα σε δύο κατηγορίες ως «προβλέψεις» και «επαληθευμένα». Κάτω από αυτές τις δύο κατηγορίες παρέχει πληθώρα επιλογών και επιμέρους εργαλείων. Κάτι ιδιαίτερα ενδιαφέρον που παρέχεται είναι η μελέτη ορισμένων αλληλεπιδράσεων σε συνάρτηση με τις ασθένειες με τις οποίες έχουν συσχετισθεί τα αντίστοιχα miRNAs. Ωστόσο ένα μικρό μειονέκτημα είναι ότι λαμβάνει υπ' όψη παλαιότερες ανανεώσεις των δεδομένων των αλγορίθμων που χρησιμοποιεί (π.χ. την έκδοση 4 του DIANA–microT) ενώ το περιβάλλον του δεν είναι ιδιαίτερα ξεκάθαρο και εύχρηστο.

3.4.2 Tools4MiRs

Παρέχει τον "Target Prediction Server" όπου ο χρήστης μπορεί να επιλέξει ποιους αλγορίθμους θέλει να συγκρίνει. Το μεγάλο πλεονέκτημα είναι ότι μπορεί να εκτελέσει τους αλγορίθμους σε πραγματικό χρόνο επάνω σε ακολουθίες που θα παρέχει ο χρήστης εκείνη τη στιγμή, κάτι που προσδίδει ευελιξία. Ωστόσο, αυτό είναι ταυτόχρονα και μειονέκτημα καθώς δεν παρέχει τη δυνατότητα επιλογής γονιδίων και miRNAs με εύκολο τρόπο. Αντίθετα, ακόμη κι αν ο χρήστης επιθυμεί να εξετάσει ένα γνωστό γονίδιο ή miRNA, πρέπει να καταγράψει την ακολουθία του σε αρχείο και κατόπιν να το «ανεβάσει» (upload) στο εργαλείο.

3.5 Εργαλεία οπτικοποίησης βιολογικής πληροφορίας

Και τα δύο προαναφερθέντα εργαλεία αποτελούν σημαντικές και χρήσιμες προσπάθειες ως προς τη σύγκριση προβλέψεων διαφορετικών αλγορίθμων. Ωστόσο κανένα από τα δύο δεν παρέχει οπτικοποιημένη παρουσίαση των συγκρίσεων ενώ, κατά την παρούσα μελέτη, δεν υπέπεσε στην αντίληψη μας κάποιο άλλο εργαλείο που να προσφέρει αυτή τη δυνατότητα. Έτσι θεωρούμε ότι ως προς αυτό το σημείο υπάρχει μεγάλο περιθώριο συνεισφοράς στη μελέτη των miRNAs.

Ανεξαρτήτως αυτού, ωστόσο, θα παρουσιάσουμε στην ενότητα αυτή ορισμένα εργαλεία οπτικοποίησης βιολογικής πληροφορίας που εφαρμόζονται σε άλλα πεδία της βιολογίας καθώς και συναφείς πρωτοβουλίες. Επίσης, η [20] είναι μία πολύ ενδιαφέρουσα δημοσίευση που εξετάζει βιβλιοθήκες ανοιχτού κώδικα οι οποίες θα μπορούσαν να χρησιμοποιηθούν για τη δημιουργία απεικονίσεων βιολογικών δεδομένων.

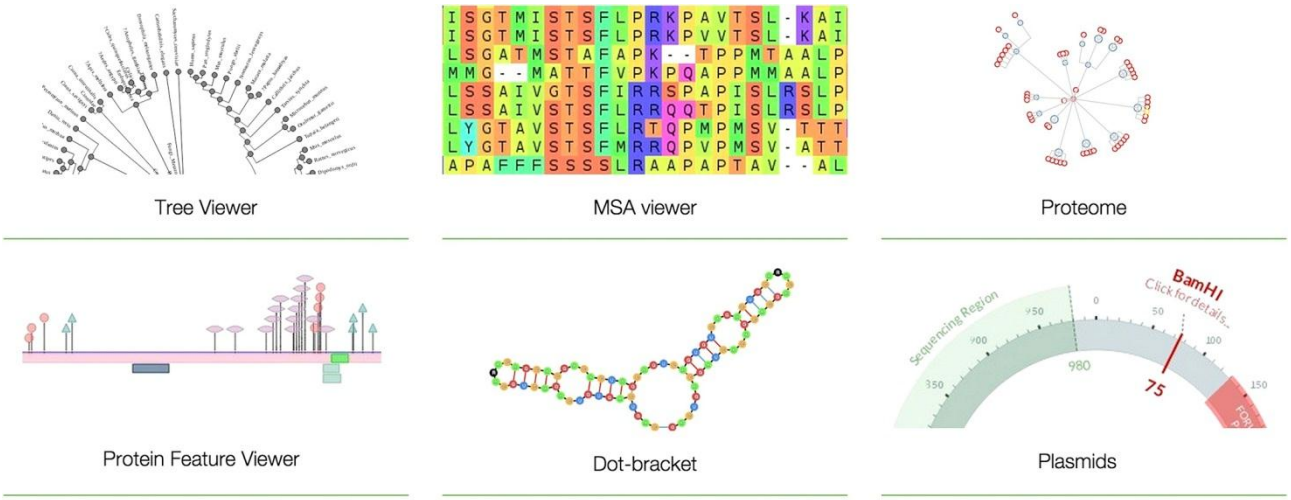
3.5.1 VIZBI (Visualizing Biological Data)

Πρόκειται για ένα ετήσιο διεθνές συνέδριο που λαμβάνει χώρα κάθε Μάρτιο και φιλοξενείται εναλλάξ σε Ευρώπη και ΗΠΑ. Στο συνέδριο αυτό συμμετέχουν ερευνητές που αναπτύσσουν και χρησιμοποιούν υπολογιστικά εργαλεία οπτικοποίησης σε ένα ευρύ φάσμα ερευνητικών περιοχών. Το συνέδριο προσελκύει επίσης και επιστήμονες της ιατρικής απεικόνισης καθώς και γραφίστες [26]. Στην ιστοσελίδα του συνεδρίου μπορεί να βρει κανείς βιντεοσκοπήσεις και διαφάνειες από το συνέδριο.

Ιδιαίτερα χρήσιμα είναι επίσης τα posters, που είναι επίσης διαθέσιμα στην ιστοσελίδα, στα οποία παρουσιάζονται εργαλεία που έχουν παρουσιαστεί στο συνέδριο. Αποτελεί, έτσι, μία πολύ σημαντική πηγή πληροφοριών σχετικά με εργαλεία που είτε υπάρχουν είτε βρίσκονται υπό ανάπτυξη από διάφορες ερευνητικές ομάδες ανά τον κόσμο ενώ μπορεί να αποτελέσει και σημαντική πηγή έμπνευσης για νέες προσπάθειες. Αποτελεί, τέλος, μία εξαιρετική προσπάθεια κοινοποίησης των διάφορων προσπαθειών αυτών ώστε να γίνονται ευρύτερα γνωστές στην ερευνητική κοινότητα της βιολογίας. Έτσι μπορεί να μεγιστοποιηθεί το όφελος των εργαλείων που αναπτύσσονται αν προωθηθεί μέσω του συνεδρίου η επαναχρησιμοποίησή τους από περισσότερες ερευνητικές ομάδες.

3.5.2 BioJS

[21, 39] Αποτελεί ένα έργο ανοιχτού κώδικα (open–source project) διαθέσιμο στην ιστοσελίδα https://biojs.net/ που στοχεύει στην ανάπτυξη εργαλείων οπτικοποίησης βιολογικής πληροφορίας. Σκοπός του έργου είναι να δημιουργηθεί μία βιβλιοθήκη της JavaScript «από χρήστες για χρήστες» η οποία να υποστηρίζει τη δημιουργία γραφημάτων που σχετίζονται με βιολογικά δεδομένα. Η προσπάθεια ξεκίνησε το 2012 ως συνεργασία του Ευρωπαϊκού Ινστιτούτου Βιοπληροφορικής (EMBL–EBI) και του Ινστιτούτου Earlham.



Εικόνα 3.3: παραδείγματα γραφημάτων δημιουργημένων με BioJS (πηγή: [21])

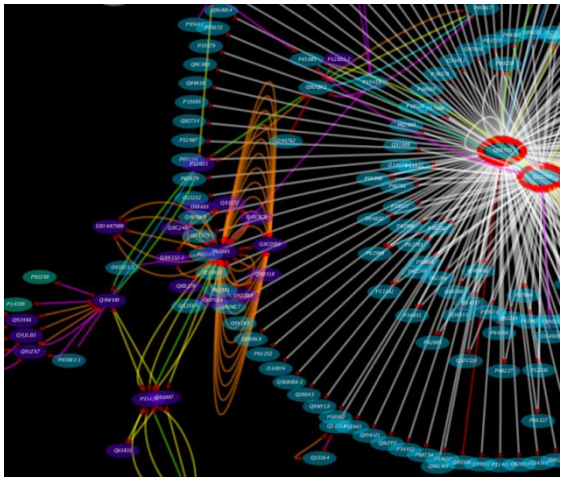
Για να επιτευχθεί η δημιουργία της βιβλιοθήκης οι δημιουργοί του ενθαρρύνουν τη δημιουργία πακέτων (packages) από τους χρήστες τα οποία, στη συνέχεια, να μπορούν να διανέμονται ώστε να χρησιμοποιούνται και από άλλους χρήστες της βιβλιοθήκης. Τα πακέτα είναι ολοκληρωμένα κομμάτια λειτουργικότητας, όπως π.χ. ένα γράφημα ή ένας επεξεργαστής αρχείων TSV (TSV file parser). Έτσι άλλοι χρήστες που επιθυμούν να χρησιμοποιούν αυτό τον τύπο γραφήματος θα μπορούν να χρησιμοποιήσουν ένα έτοιμο πακέτο αντί να δημιουργήσουν το γράφημα από την αρχή.

Με άλλα λόγια, στόχος του συγκεκριμένου έργου είναι να προωθήσει τη συνεργασία στο πεδίο της ανάπτυξης γραφημάτων για βιολογικά δεδομένα και, κατ' επέκταση, να δημιουργηθεί μία πλατφόρμα έτοιμων γραφημάτων και εργαλείων.

3.5.3 Cytoscape

Το Cytoscape (http://www.cytoscape.org/) είναι λογισμικό ανοιχτού κώδικα που δημιουργεί απεικονίσεις δικτύων μοριακών αλληλεπιδράσεων και βιολογικών μονοπατιών. Οι απεικονίσεις αυτές ενισχύονται με την προσθήκη επισημειώσεων, προφίλ γονιδιακής έκφρασης και άλλα συναφή δεδομένα. Ο πυρήνας του λογισμικού παρέχει ένα βασικό σύνολο λειτουργιών για την επεξεργασία δεδομένων, ανάλυσή τους και τη δημιουργία απεικονίσεων που ανταποκρίνονται σε αυτά τα δεδομένα. Επιπρόσθετη λειτουργικότητα

παρέχεται μέσω επιπλέον πακέτων που ονομάζονται “Apps”. Το λογισμικό αυτό χρησιμοποιείται εκτενώς σε διάφορες βιολογικές εφαρμογές (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC403769/citedby/).



Εικόνα 3.4: παράδειγμα απεικόνισης βιολογικού δικτύου με Cytoscape (πηγή: cytocsape.org)

4

Εργαλεία και τεχνολογίες που χρησιμοποιήθηκαν

Η υλοποίηση της παρούσης εργασίας μπορεί να διαχωριστεί σε τέσσερα διακριτά μέρη:

1. Το πρώτο κομμάτι αφορά όλες τις εργασίες που σχετίζονταν με τα δεδομένα της εφαρμογής, δηλαδή το κατέβασμα των δεδομένων, την προετοιμασία τους (data preparation and data clean-up) και τη διαμόρφωση της βάσης δεδομένων.
2. Το δεύτερο μέρος αφορά τον προγραμματισμό της εφαρμογής από την πλευρά του εξυπηρετητή (server side programming).
3. Το τρίτο μέρος αφορά τη διαμόρφωση της ιστοσελίδας και της διεπαφής με το χρήστη (web pages and user interface).
4. Το τέταρτο μέρος αφορά τη δημιουργία των γραφημάτων (data visualisations) με τις δυνατότητες αλληλεπίδρασης με το χρήστη που αυτά πρέπει να έχουν.

Στο κεφάλαιο αυτό παρουσιάζονται οι διάφορες τεχνολογίες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν στα διάφορα στάδια υλοποίησης της εργασίας.

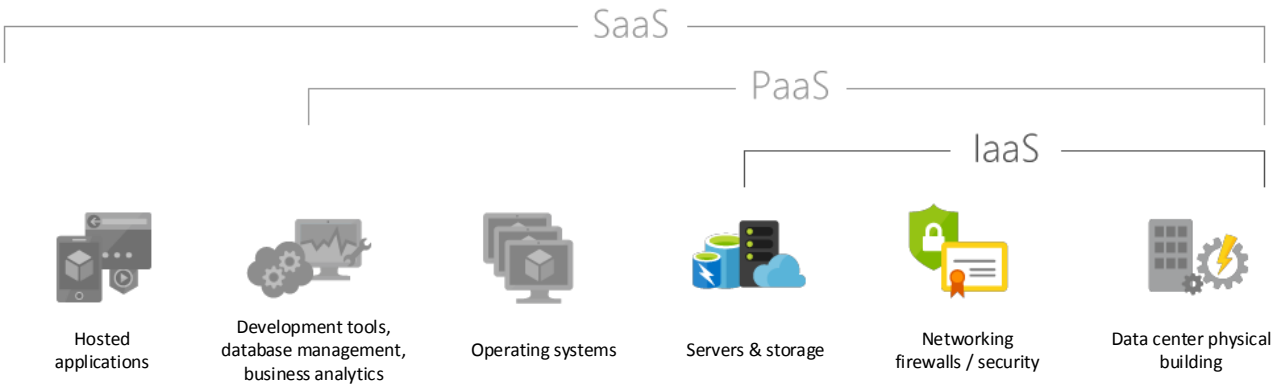
4.1 ~Okeanos

4.1.1 Γενικά

Ο Ωκεανός (https://okeanos.grnet.gr/) είναι η υπηρεσία νέφους (cloud computing service) του «Εθνικού Δικτύου Έρευνας & Τεχνολογίας» (https://grnet.gr/) και είναι ελεύθερα διαθέσιμη για όλα τα μέλη της ερευνητικής και ακαδημαϊκής κοινότητας. Παρέχει εικονικούς πόρους υλικού (virtualised hardware) και διατίθεται με τη μορφή της «υποδομής–ως–υπηρεσία» (IaaS – Infrastructure as a Service) [40].

Ο Ωκεανός παρέχει δύο κύρια εργαλεία: ένα χώρο αποθήκευσης (cloud storage), τον «Πίθο» (Pithos), και μία υπηρεσία παροχής εικονικών μηχανών και εικονικών δικτύων, τις «Κυκλάδες» (Cyclades). Με τις Κυκλάδες μπορεί κανείς να δημιουργήσει εικονικές μηχανές σε πολύ μικρό χρόνο, με τα επιθυμητά χαρακτηριστικά υλικού και το επιθυμητό λειτουργικό σύστημα [41]. Η υπηρεσία παρέχει έτοιμες εικόνες (images) από τα πιο δημοφιλή λειτουργικά συστήματα αλλά ένας έμπειρος χρήστης μπορεί, αν επιθυμεί, να δημιουργήσει και να χρησιμοποιήσει στις εικονικές μηχανές του μία δική του, ειδικά προσαρμοσμένη εικόνα (custom image).

Για την εργασία αυτή θα αξιοποιήσουμε τις Κυκλάδες όπου θα δημιουργηθεί ένας εικονικός εξυπηρετητής ο οποίος θα φιλοξενήσει την εφαρμογή που θα αναπτυχθεί.



Εικόνα 4.1: τα τρία μοντέλα παροχής υπηρεσιών νέφους (cloud services)

(πηγή: azure.microsoft.com/en-us/overview/what-is-iaas/)

4.1.2 Πλεονεκτήματα υπηρεσιών IaaS

Τα βασικά πλεονεκτήματα των υπηρεσιών IaaS είναι [42, 43, 57]:

- καθότι οι υπηρεσίες αυτές βασίζονται σε συστάδες υπολογιστών (clusters) και η φυσική υποδομή που τα υποστηρίζει διαθέτει εφεδρείες (redundancies), δεν υπάρχουν μεμονωμένα φυσικά εξαρτήματα που θα μπορούσαν να προκαλέσουν την κατάρρευση της υπηρεσίας (single point of failure). Σε έναν φυσικό υπολογιστή κάθε εξάρτημα είναι μοναδικό.
- ένα εικονικό μηχάνημα καταστρέφεται και δημιουργείται εκ νέου σε πολύ μικρό χρόνο και με μηδαμινό κόπο. Η επαναδιαμόρφωση (format) ενός φυσικού υπολογιστή απαιτεί αρκετές ώρες.
- ενδεχόμενες επιπλοκές κατά την ανάπτυξη μιας εφαρμογής δεν επιδρούν στη φυσική υπολογιστή υποδομή που διαθέτουμε.
- ικανότητα αναβάθμισης ή υποβάθμισης (scale up/down) των χαρακτηριστικών της εικονικής υποδομής που διαθέτουμε σε πολύ μικρό χρόνο. Για να γίνει αυτό σε μία φυσική υποδομή (υπολογιστής ή, ακόμη χειρότερα, ένα δίκτυο) απαιτεί επίπονη διαδικασία αναβάθμισης και, το κυριότερο, σημαντικό χρόνο κατά τον οποίο η εξυπηρετούμενη εφαρμογή θα μείνει ανενεργή (downtime).

- η ευελιξία μιας εικονικής υποδομής επιτρέπει στο χρήστη να προσαρμόζει το κόστος χρήσης της υπηρεσίας ανάλογα με το πόση υπολογιστική ισχύ πράγματι χρησιμοποιεί. Αντίθετα, μία φυσική υποδομή, έχει σταθερά κόστη αγοράς και συντήρησης.
- ο χρήστης απαλλάσσεται από την ανάγκη να επενδύσει σε φυσική υποδομή.

Τα παραπάνω πλεονεκτήματα καθιστούν τις υπηρεσίες IaaS ιδανικές για περιβάλλοντα ανάπτυξης εφαρμογών, για εικονικούς εξυπηρετητές που φιλοξενούν ιστοσελίδες ή διαδικτυακές εφαρμογές καθώς και για διαμοιρασμό αρχείων (file sharing) ή γενικότερα αποθήκευση στο Νέφος (Cloud storage).

4.2 Προεπεξεργασία δεδομένων με AWK

Τα δεδομένα που θα συλλεχθούν από τις διάφορες πηγές δε βρίσκονται σε μορφή κατάλληλη ώστε να φορτωθούν αυτούσια στη βάση δεδομένων που θα δημιουργηθεί. Έτσι, είναι βέβαιο ότι θα πρέπει να γίνει κάποια προεπεξεργασία των δεδομένων. Για αυτό το σκοπό επιλέχθηκε η γλώσσα προγραμματισμού AWK, η οποία χρησιμοποιείται ως εργαλείο εξαγωγής δεδομένων και παραγωγής αναφορών και στατιστικών στοιχείων. Δημιουργήθηκε το 1977 στα "AT&T Bell Labs" από τους Alfred Aho, Peter Weinberger και Brian Kernighan ενώ το όνομά της αποτελεί απλώς ακρωνύμιο των επωνύμων τους [44].

Ένας κώδικας σε AWK εκτελείται από διερμηνέα (interpreter), δηλαδή δεν παράγεται κάποιο εκτελέσιμο. Ο προγραμματιστής γράφει τον κώδικα σε ένα αρχείο (script) το οποίο εκτελείται από το τερματικό του υπολογιστή (terminal / command line) με είσοδο το επιθυμητό αρχείο κειμένου προς επεξεργασία. Η έξοδος μπορεί να είναι από πολύ απλή (π.χ. ένα πολύ απλό στατιστικό που εξάγεται από την επεξεργασία όλων των γραμμών του αρχείου εισόδου) έως πολύ σύνθετη (π.χ. ένα ολόκληρο νέο αρχείο κειμένου το οποίο παράγεται κατόπιν επεξεργασίας του αρχείου εισόδου).

Η λογική ενός προγράμματος σε AWK είναι πολύ απλή και βασίζεται σε ζεύγη μοτίβο – δέσμη ενεργειών ως εξής [45]:

```
pattern { actions }
```

Ο προγραμματιστής ορίζει μοτίβα (patterns) ως κανονικές εκφράσεις (regular expressions) και η AWK διασχίζει το αρχείο εισόδου γραμμή προς γραμμή. Αν η τρέχουσα γραμμή ταιριάζει με κάποιο από τα μοτίβα που έχουν οριστεί, τότε εκτελείται η δέσμη ενεργειών (actions) που αντιστοιχεί σε αυτό το μοτίβο. Έτσι η AWK επεξεργάζεται το αρχείο εισόδου γραμμή προς γραμμή και εκτελεί για κάθε γραμμή τις ενέργειες που πρέπει. Βεβαίως μπορεί μία γραμμή να ταιριάξει με περισσότερα από ένα μοτίβα ή και με κανένα. Τότε θα εκτελεστούν, αντίστοιχα, όλες οι δέσμες ενεργειών που «ταίριαξαν» με τη γραμμή ή δεν θα γίνει καμία απολύτως ενέργεια.

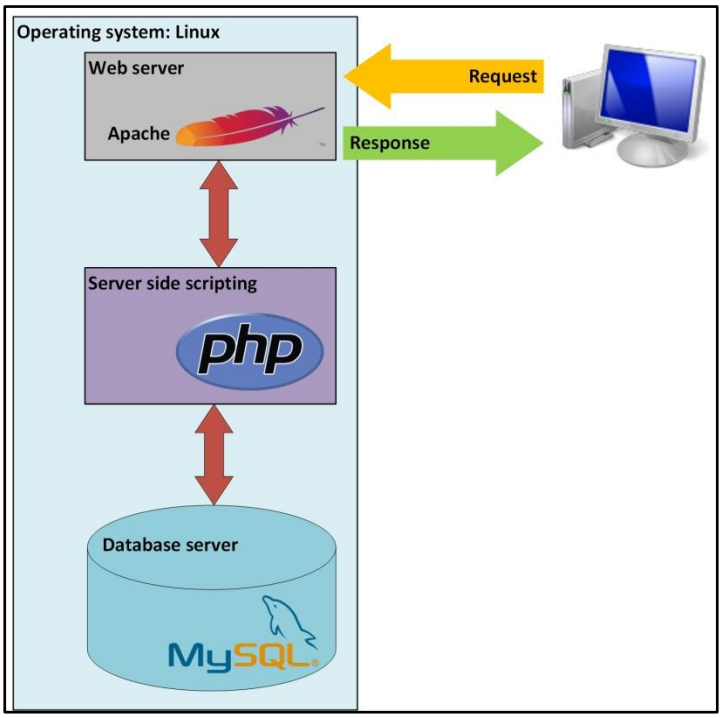
Όλα τα παραπάνω, με άλλα λόγια, σημαίνουν ότι ένας κώδικας σε AWK δεν αποτελεί απλώς μία «λίστα εντολών» όπως θα συνέβαινε σε μια γλώσσα προστακτικού προγραμματισμού. Αντίθετα ο κώδικας **περιγράφει τη δομή των δεδομένων** εισόδου, τα μοτίβα με τα οποία πρέπει να «ταιριάξουν» τα δεδομένα αυτά και το «πώς» πρέπει να γίνει η επεξεργασία των δεδομένων. Ακριβώς λόγω αυτού του τρόπου λειτουργίας της, η AWK αποκαλείται συχνά ως «γλώσσα οδηγούμενη από δεδομένα» (data-driven language) [58].

Το κυριότερο χαρακτηριστικό που ξεχωρίζει την AWK από τα υπόλοιπα εργαλεία επεξεργασίας κειμένου και την καθιστά ιδανικό εργαλείο για την επεξεργασία αρχείων όπως τα CSV και TSV, είναι ότι διαχωρίζει τις γραμμές σε πεδία ανάλογα με το διαχωριστικό χαρακτήρα (delimiter). Με άλλα λόγια, η επεξεργασία κάθε γραμμής ξεκινά με το διαχωρισμό της στα επιμέρους πεδία που την αποτελούν. Για αρχεία κειμένου που αναπαριστούν πίνακες, όπως ήταν όλα τα πρωτογενή αρχεία δεδομένων που χρησιμοποιήσαμε, αυτό το ιδιαίτερο χαρακτηριστικό της AWK κάνει πάρα πολύ εύκολη την επεξεργασία των γραμμών. Ο προγραμματιστής, σε αυτή την περίπτωση, δε χρειάζεται να γράψει κώδικα για να διαχωρίσει (parse) την κάθε γραμμή σε λέξεις ή πεδία ανάλογα με κενά, κόμματα ή στηλοθέτες οπότε ο κώδικας αφορά εξ ολοκλήρου και αποκλειστικά το χειρισμό αυτών καθαυτών των δεδομένων.

Χρησιμοποιώντας, λοιπόν, AWK επεξεργαστήκαμε τα πρωτογενή αρχεία δεδομένων που κατεβάσαμε από τις πηγές δεδομένων που χρησιμοποιήθηκαν και δημιουργήσαμε τους πίνακες με τους οποίους φορτώσαμε, τελικά, τη βάση δεδομένων.

4.3 Πακέτο εργαλείων LAMP (LAMP stack)

Επόμενο βήμα είναι η επιλογή λογισμικού για τη διαμόρφωση του εξυπηρετητή (server). Για να λειτουργήσει ένας υπολογιστής ως εξυπηρετητής χρειάζεται τέσσερα βασικά μέρη λογισμικού: λειτουργικό σύστημα, ένα πρόγραμμα εξυπηρετητή (server), μία γλώσσα προγραμματισμού για την ανάπτυξη της εφαρμογής και μία βάση δεδομένων. Ως λύση για τη διαμόρφωση του συστήματος προτιμήθηκε να χρησιμοποιηθεί το πακέτο εργαλείων LAMP (Linux, Apache, MySQL & PHP) καθώς αποτελεί μία ευρέως δοκιμασμένη και πολύ δημοφιλή λύση για την ανάπτυξη ιστοσελίδων και διαδικτυακών εφαρμογών [46].



Εικόνα 4.2: η αρχιτεκτονική ενός εξυπηρετητή που βασίζεται στο πακέτο LAMP

Παρ' ότι τα εν λόγω εργαλεία δεν σχεδιάστηκαν ώστε κατ' ανάγκην να λειτουργούν όλα μαζί, ωστόσο συνεργάζονται ιδανικά μεταξύ τους και, ως πακέτο, συναποτελούν μία πολύ ισχυρή πλατφόρμα ανάπτυξης διαδικτυακών εφαρμογών [47]. Αυτό επιτρέπει στον προγραμματιστή να επικεντρώσει τις προσπάθειές του σε αυτή καθαυτή την ανάπτυξη της εφαρμογής παρά στη διαμόρφωση του εξυπηρετητή και σε προσπάθειες να

καταστήσει δυνατή τη συνεργασία των επιμέρους μερών του συστήματος (integration). Επίσης, η εγκατάσταση και συντήρηση όλων των εφαρμογών του πακέτου είναι πολύ εύκολη και γρήγορη.
Το πακέτο LAMP είναι αρκετά δημοφιλές και έχει πολύ μεγάλη κοινότητα χρηστών. Έτσι είναι πολύ εύκολο να βρει κανείς λύσεις σε προβλήματα που ενδεχομένως προκύψουν κατά την ανάπτυξη της εφαρμογής . Πέραν της εξαιρετικής επίσημης τεκμηρίωσης (documentation) που παρέχεται από όλα τα επιμέρους εργαλεία, υπάρχουν αναρίθμητες άλλες συμπληρωματικές πηγές στο Διαδίκτυο για διαμόρφωση, αποσφαλμάτωση (debugging), βελτιστοποίηση κτλ.

4.3.1 MySQL

Επόμενο βήμα, σχετικά με τα δεδομένα, ήταν η επιλογή ενός συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS – Relational DataBase Management System). Μία βάση δεδομένων είναι απαραίτητη καθώς εκεί θα αποθηκεύονται όλες οι σχετικές πληροφορίες της εφαρμογής και κυρίως τα αποτελέσματα των αλγορίθμων και επιλεγμένα στοιχεία των γονιδίων και των miRNAs. Η βάση δεδομένων θα αποτελεί το βασικό στοιχείο αποθήκευσης πληροφορίας της εφαρμογής και έτσι απαιτείται ένα αξιόπιστο, δοκιμασμένο εργαλείο που να προσφέρει υψηλή απόδοση για βάσεις δεδομένων με μεγάλο αριθμό εγγραφών.
Η MySQL είναι βασικό συστατικό του πακέτου λογισμικού LAMP. Οι βασικοί λόγοι για τους οποίους επιλέξαμε τη συγκεκριμένη βάση δεδομένων είναι ότι είναι αξιόπιστη, η εγκατάσταση και διαμόρφωσή της είναι πολύ απλή και γρήγορη, λειτουργεί ιδανικά σε συνδυασμό με τα υπόλοιπα μέρη του πακέτου LAMP ενώ, φυσικά, είναι ελεύθερα διαθέσιμη. Ορισμένα ακόμη πλεονεκτήματά της είναι ότι πρόκειται για μια βάση δεδομένων αρκετά ισχυρή στο να πραγματοποιεί τάχιστα αναγνώσεις και να εξυπηρετεί μεγάλο αριθμό χρηστών ενώ είναι ικανή να χειρίζεται ταχύτατα και αποδοτικά μεγάλους όγκους δεδομένων [48]. Μάλιστα είναι μία βάση δεδομένων που χρησιμοποιείται κατά κόρον σε διαδικτυακές εφαρμογές και μάλιστα σε αρκετές περιπτώσεις πολύ υψηλών απαιτήσεων [49].

4.4 Laravel

Το επόμενο βήμα είναι η επιλογή των εργαλείων για την ανάπτυξη της εφαρμογής. Στο κεφάλαιο 2 παρουσιάσαμε τα πλαίσια ανάπτυξης διαδικτυακών εφαρμογών καθώς και τη μεθοδολογία MVC. Για τους λόγους που αναλύθηκαν εκεί, επιλέχθηκε και σε αυτή την εργασία ένα πλαίσιο ανάπτυξης στο οποίο βασίστηκε η δημιουργία της εφαρμογής.

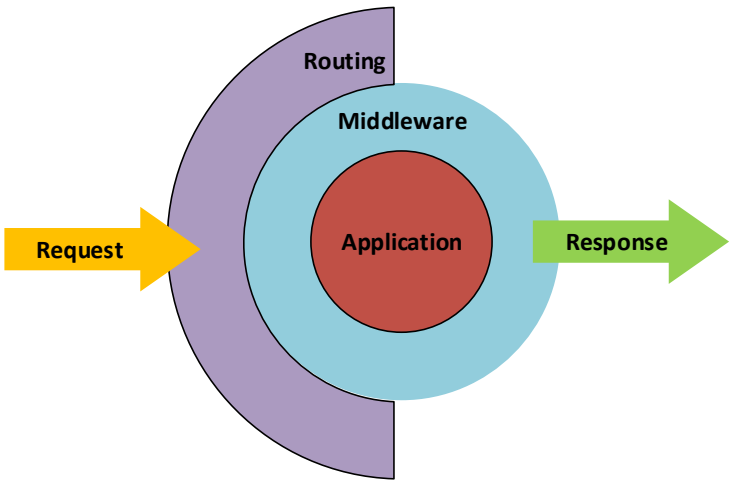
4.4.1 Γενικά

Χρησιμοποιήσαμε το Laravel, ένα πλαίσιο ανάπτυξης εφαρμογών για PHP που βασίζεται στη μεθοδολογία MVC. Εμφανίστηκε το 2011 και έκτοτε έχει αναδειχθεί σε ένα από τα πιο δημοφιλή πλαίσια ανάπτυξης διαδικτυακών εφαρμογών, ειδικά σε σύγκριση με τα υπόλοιπα πλαίσια ανάπτυξης για PHP [59, 60]. Γύρω από αυτό έχει αναπτυχθεί ένα ολόκληρο «οικοσύστημα» (όπως έχει χαρακτηριστεί) από εφαρμογές και υπηρεσίες που υποστηρίζουν την ανάπτυξη εφαρμογών με Laravel (π.χ. Envoyer, Forge, Spark, Lumen).

Επίσης, με βάση το Laravel, έχουν δημιουργηθεί και πολλά συστήματα διαχείρισης περιεχομένου (CMS) (ενδεικτικά αναφέρουμε τα October, PyroCMS, Asgard κ.ά.).
Η κοινότητα χρηστών του είναι ιδιαίτερα ενεργή με πάρα πολλές ιστοσελίδες, κυριότερες εκ των οποίων την επίσημη τεκμηρίωση (documentation), το Laracasts και το αποθετήριο (repository) του Laravel στο Github. Ωστόσο, πέραν των επίσημων πηγών, υπάρχουν στο Διαδίκτυο αναρίθμητες ιστοσελίδες με οδηγούς (tutorials), συμβουλές (common & best practices) και ομάδες συζητήσεων (forums) γύρω από αυτό, κάτι που του προσδίδει ιδιαίτερη ισχύ.

4.4.2 Πώς λειτουργεί

Δεδομένου ότι το Laravel αποτελεί τη βάση και της δικής μας εφαρμογής θα αφιερώσουμε λίγο χώρο για να το παρουσιάσουμε κάπως πιο αναλυτικά. Ο πιο εύκολος τρόπος για να αναλύσουμε τα πιο σημαντικά μέρη του καθώς και να δείξουμε πώς λειτουργεί, είναι να εξηγήσουμε τι συμβαίνει όταν ένα αίτημα (HTTP request) φτάνει στον εξυπηρετητή.
Η πρώτη ενέργεια που γίνεται για την εξυπηρέτηση ενός αιτήματος είναι να δημιουργηθεί το πρώτο αντικείμενο (instance), η ίδια η εφαρμογή. Στη συνέχεια αρχικοποιείται η εφαρμογή και φορτώνονται τα βασικά της μέρη (application bootstrapping). Μόλις η εφαρμογή είναι έτοιμη αρχίζει η επεξεργασία του αιτήματος η οποία περιλαμβάνει τα ακόλουθα στάδια:



Εικόνα 4.3: σχηματική αναπαράσταση της εξυπηρέτησης ενός αιτήματος HTTP από μία εφαρμογή ανεπτυγμένη με Laravel

1. **Δρομολόγηση (routing)**. Ανάλογα με τη διεύθυνση URL που ζήτησε ο χρήστης η εφαρμογή δρομολογεί το αίτημα προς τον αντίστοιχο ελεγκτή (controller) που θα το εξυπηρετήσει. Το ίδιο ισχύει για όλα τα αιτήματα (get, put, post, delete) καθώς επίσης και για ασύγχρονα αιτήματα (AJAX).
2. **Ενδιάμεσες μονάδες (middleware)**. Σε πολλές περιπτώσεις, προτού εξυπηρετηθεί ένα αίτημα, πρέπει να γίνουν πρώτα κάποιοι έλεγχοι ή κάποια βήματα προεπεξεργασίας του αιτήματος. Μερικά παραδείγματα είναι ο έλεγχος αν ο χρήστης είναι πιστοποιημένος (authenticated user), αν έχει υποβληθεί μία φόρμα να ελεγχθεί ότι οι επιλογές είναι έγκυρες (form validation), να γίνουν κάποιοι

έλεγχοι ασφαλείας όπως το να επιβεβαιωθεί ότι δεν προσπαθεί κακόβουλα κάποιος να «μιμηθεί» έναν πιστοποιημένο χρήστη (Cross Site Request Forgery protection) κ.ά. Αν όλες οι ενέργειες αυτού του βήματος ολοκληρωθούν επιτυχώς, τότε η επεξεργασία θα προχωρήσει κανονικά και το αίτημα θα ανατεθεί στον αντίστοιχο ελεγκτή. Ωστόσο, μπορεί σε αυτό το στάδιο να προκύψει η διαπίστωση ότι το αίτημα δεν πρέπει να εξυπηρετηθεί για κάποιον λόγο. Τότε ακυρώνεται η περαιτέρω επεξεργασία του αιτήματος και επιστρέφεται στο χρήστη κατάλληλη απόκριση, όπως π.χ. ανακατεύθυνση στην αρχική σελίδα για είσοδο (login).

Εκτός από τις ενδιάμεσες μονάδες που δρουν πριν τους ελεγκτές, υπάρχουν και ενδιάμεσες μονάδες οι οποίες δρουν μετά την κυρίως επεξεργασία, αν και αυτή η χρήση τους είναι κάπως πιο σπάνια. Ένα παράδειγμα θα ήταν αν πρέπει να καταγράφονται σε κάποιο αρχείο (log) οι λειτουργίες που εκτελέστηκαν απ' την εφαρμογή ή να γίνει «καθάρισμα» προσωρινών δεδομένων μετά την αποσύνδεση (logout) ενός χρήστη. Τέτοιου είδους εργασίες δεν μπορούν να γίνουν αν δεν έχει ολοκληρωθεί η κυρίως επεξεργασία.

3. **Κυρίως επεξεργασία**. Η κυρίως επεξεργασία των αιτημάτων γίνεται από τους ελεγκτές (controllers). Η λειτουργικότητα της εφαρμογής οργανώνεται με βάση τους ελεγκτές ούτως ώστε κάθε ελεγκτής να υλοποιεί μία συγκεκριμένη λειτουργία. Έτσι ο κάθε ελεγκτής αναλαμβάνει να επεξεργαστεί ένα συγκεκριμένο αίτημα ή ομάδα συναφών αιτημάτων προς την εφαρμογή. Γι' αυτόν ακριβώς το λόγο έχει προηγηθεί το στάδιο της δρομολόγησης έτσι ώστε η εφαρμογή να αποφασίσει ποιος από τους διαθέσιμους ελεγκτές είναι ο αρμόδιος να επεξεργαστεί το τρέχον αίτημα. Η επεξεργασία ενός αιτήματος περιλαμβάνει, επίσης, και ενδεχόμενη επικοινωνία με τη βάση δεδομένων (κομμάτι Model).
4. **Επιστροφή απόκρισης στο χρήστη**. Η τελευταία αρμοδιότητα των ελεγκτών είναι να δημιουργήσουν μία απόκριση (response) και να την επιστρέψουν στο χρήστη (κομμάτι View). Έτσι,. Η εξυπηρέτηση αυτού του αιτήματος έχει ολοκληρωθεί και η ιστοσελίδα προβάλλεται στον φυλλομετρητή (browser) του χρήστη.

Αυτός είναι ο σκελετός της λειτουργίας του Laravel. Πάνω σε αυτή τη λογική ο προγραμματιστής μπορεί να δημιουργήσει ενδιάμεσες μονάδες (middleware) και ελεγκτές που υλοποιούν τις επιθυμητές λειτουργίες της εφαρμογής (κομμάτι Controller). Επίσης, δημιουργεί κλάσεις που ονομάζονται «μοντέλα» (κομμάτι Model), οι οποίες αναλαμβάνουν την επικοινωνία με τη βάση δεδομένων. Τα μοντέλα καλούνται από τους ελεγκτές όταν χρειάζεται. Τέλος, η κατασκευή της απόκρισης προς το χρήστη (κομμάτι View) γίνεται με HTML και ένα πολύ ισχυρό εργαλείο του Laravel, το Blade.

4.4.3 Η σημασία της επικύρωσης δεδομένων από τον εξυπηρετητή (server side validation) και ο ρόλος του middleware

Τα αιτήματα HTTP από το χρήστη προς έναν ιστότοπο συνοδεύονται σχεδόν πάντα από κάποια δεδομένα (εκτός από την περίπτωση της απλής προβολής μιας στατικής ιστοσελίδας). Ο ιστότοπος αναμένει αυτά τα δεδομένα έτσι ώστε να εξυπηρετήσει κατάλληλα το αίτημα. Για παράδειγμα, σε μία οθόνη εισόδου μιας

υπηρεσίας, όλοι οι χρήστες θα επισκέπτονται τη σελίδα "homepage/user/login". Όμως για να γνωρίζει ο ιστότοπος ποιος ακριβώς χρήστης επιχειρεί να εισέλθει στο σύστημα, το αίτημα προς αυτή τη διεύθυνση θα πρέπει να συνοδεύεται και από μερικά δεδομένα.

Ωστόσο, δεν μπορεί μια εφαρμογή να υποθέτει ότι πάντοτε τα δεδομένα που θα λαμβάνει θα είναι πλήρη και έγκυρα. Υπάρχουν πολλών ειδών σφάλματα που μπορεί να προκαλέσουν αποστολή ενός αιτήματος με λανθασμένα δεδομένα. Το πιο απλό, μία ενδεχόμενη λανθασμένη κατασκευή (rendering) μιας HTML σελίδας μπορεί να έχει ως αποτέλεσμα να λείπουν στοιχεία <input> του HTML κώδικα από μία φόρμα και, άρα, να λείπουν δεδομένα όταν υποβληθεί η φόρμα. Για τον ίδιο λόγο, μπορεί τα στοιχεία <input> να υπάρχουν αλλά να μην έχουν δημιουργηθεί σωστά από την PHP/HTML, δηλαδή να έχουν λάθος τιμές (π.χ. λάθος ιδιότητα value). Άλλη πιθανή αιτία είναι σφάλματα που συμβαίνουν κατά την αποστολή του αιτήματος από το φυλλομετρητή του χρήστη προς τον εξυπηρετητή, όπως η μη αποστολή κάποιων δεδομένων ή διάβρωσή τους κατά την αποστολή. Επίσης, δεν πρέπει να αμελείται η πιθανότητα αποστολής κακόβουλων αιτημάτων προς την ιστοσελίδα, π.χ. με κακόβουλο κώδικα JavaScript ή με ψευδή ερωτήματα από άλλες ιστοσελίδες (CSRF – cross site request forgery).

Για αυτούς και πολλούς ακόμα λόγους, είναι απαραίτητο κάθε αίτημα που λαμβάνεται από μία εφαρμογή να ελέγχεται έτσι ώστε να επιβεβαιωθεί ότι έχει τη μορφή που πρέπει, συνοδεύεται από τα κατάλληλα δεδομένα, οι τιμές των δεδομένων αυτών ανήκουν στις αναμενόμενες, οι διευθύνσεις που ζητώνται είναι έγκυρες κτλ.

Στο Laravel αυτό το ρόλο ακριβώς εξυπηρετούν οι ενδιάμεσες μονάδες (middleware), οι οποίες ελέγχουν τέτοιου είδους ζητήματα σε κάθε ερώτημα προτού το αίτημα προωθηθεί προς τους αντίστοιχους ελεγκτές για επεξεργασία. Η ύπαρξη των ενδιάμεσων μονάδων είναι ένα πλεονέκτημα που παρουσιάζει το Laravel διότι, κατ' αυτόν τον τρόπο, διαχωρίζεται απόλυτα αυτό το κομμάτι «προεπεξεργασίας» ενός αιτήματος από την κυρίως λογική της εφαρμογής. Οι πάσης φύσεως έλεγχοι εκτελούνται από τις ενδιάμεσες μονάδες ενώ η λογική της εφαρμογής υλοποιείται αποκλειστικά από τους ελεγκτές. Έτσι, τα δύο αυτά είδη κλάσεων, έχουν απολύτως διακριτούς ρόλους.

4.5 Σχεδιασμός σελίδων

Το τρίτο προγραμματιστικό κομμάτι ανάπτυξης της εφαρμογής (μετά τα δεδομένα και τη δημιουργία της εφαρμογής) ήταν ο σχεδιασμός των ιστοσελίδων. Για αυτό το σκοπό αξιοποιήσαμε, βεβαίως, HTML, CSS και JavaScript, το θεμελιώδες τρίπτυχο γλωσσών προγραμματισμού για σχεδίαση ιστοσελίδων. Η HTML αναλαμβάνει να καθορίσει το περιεχόμενο μιας ιστοσελίδας και τη διάρθρωσή της, η CSS καθορίζει τον τρόπο εμφάνισης του ορισθέντος περιεχομένου και η JavaScript εισάγει χαρακτηριστικά δυναμικής συμπεριφοράς ώστε οι ιστοσελίδες να μην είναι στατικές και ο χρήστης να μπορεί να αλληλεπιδρά μαζί τους.

Στην παρούσα ενότητα θα αναφερθούμε σε τρία εργαλεία που διευκολύνουν τη σχεδίαση ιστοσελίδων.

4.5.1 Laravel Blade

Το Blade είναι η μηχανή σχεδίασης ιστοσελίδων με πρότυπα (templating engine) που παρέχει το Laravel και πρόκειται για ένα εργαλείο που βοηθά στο σχεδιασμό των ιστοσελίδων και στη δόμηση των HTML αρχείων. Ο λόγος που κάνουμε ειδική μνεία σε αυτό είναι ότι παρουσιάζει ορισμένες εξαιρετικές δυνατότητες που διευκολύνουν σημαντικά τον προγραμματιστή ενώ αξιοποιήθηκε σε πολύ μεγάλο βαθμό κατά τη σχεδίαση της εφαρμογής.

Όσοι ιστότοποι δεν διαθέτουν μόνο μία σελίδα, συνήθως ακολουθούν μία κοινή βασική διάρθρωση (layout) σε όλες τις σελίδες τους και αλλάζουν μόνο το περιεχόμενο στα διάφορα πεδία (containers, panels) της εκάστοτε σελίδας. Ένα τυπικό παράδειγμα κοινής διάρθρωσης είναι οι σελίδες να διαθέτουν επικεφαλίδα, υποσέλιδο, μία κεντρική μπάρα πλοήγησης στην κορυφή και μία πλευρική μπάρα εργαλείων. Ωστόσο, για να είναι όλα αυτά τα στοιχεία κοινά σε όλες τις σελίδες ενός ιστοτόπου, θα πρέπει η HTML που τα παράγει να επαναλαμβάνεται σε κάθε σελίδα. Έτσι, για να γίνει μια αλλαγή σε αυτή τη βασική δομή, θα πρέπει να αλλάξουν όλα τα αρχεία που περιέχουν αυτόν τον βασικό κώδικα HTML. Αυτή η επαναληψιμότητα, όμως, έχει πολλές, βασικές και γνωστές προγραμματιστικές δυσκολίες.

Για την αντιμετώπιση αυτού ακριβώς του προβλήματος, το Blade κάνει κάτι πάρα πολύ κομψό, μεταφέροντας ένα από τα βασικά γνωρίσματα του αντικειμενοστραφούς προγραμματισμού στη σχεδίαση ιστοσελίδων: συνδυάζει την HTML με την κληρονομικότητα. Με το Blade, είναι δυνατόν να ορίσουμε τη βασική διάρθρωση των σελίδων μία φορά ως πρότυπο (template) σε ένα αρχείο HTML. Στη συνέχεια, κάθε σελίδα που πρέπει να ακολουθεί αυτή τη δομή, αρκεί απλώς να κληρονομήσει αυτό το βασικό πρότυπο και, αν απαιτείται, να κάνει οποιεσδήποτε επιπλέον προσθήκες. Με αυτόν τον τρόπο έχει υλοποιηθεί πολύ εύκολα για τον προγραμματιστή η κοινή διάρθρωση των σελίδων ενώ, επιπλέον, υπάρχει η δυνατότητα διαφορετικού περιεχομένου για κάθε σελίδα.

Η δεύτερη βασική δυνατότητα που παρέχει το Blade, είναι το να εισάγουμε στην ιστοσελίδα ολόκληρα, αυτοτελή κομμάτια κώδικα HTML που βρίσκονται σε ξεχωριστό αρχείο. Αυτά τα αυτοτελή κομμάτια το Blade τα ονομάζει «υποενότητες» (sections). Για παράδειγμα, έστω ότι κατασκευάζουμε ένα πλαίσιο εμφάνισης μηνυμάτων ως μία υποενότητα. Ο κώδικας HTML γράφεται μία φορά μόνο σε ένα αρχείο και στη συνέχεια, σε όσες σελίδες είναι επιθυμητό να παρέχεται αυτό το πλαίσιο μηνυμάτων, μπορούμε πολύ απλά να συμπεριλάβουμε αυτή την υποενότητα με μία μόνο εντολή. Το σημαντικό εδώ (που είναι και η βασική διαφορά με τα πρότυπα) είναι ότι οι υποενότητες μπορούν να εισαχθούν σε οποιοδήποτε σημείο μιας σελίδας – δεν είναι, δηλαδή, αναγκαίο όσες σελίδες ενσωματώσουν την υποενότητα, να την εισάγουν στο ίδιο σημείο.

Η τρίτη βασική δυνατότητα του Blade (και τελευταία που θα αναφέρουμε εδώ) είναι η δυνατότητα χρήσης δομών ελέγχου σε ένα HTML αρχείο σαν να ήταν αρχείο μιας οποιασδήποτε άλλης γλώσσας προγραμματισμού. Είναι εφικτό, δηλαδή, να συντάσσεται HTML μέσα σε βρόχους (for, foreach, while) και συνθήκες (if). Στο παράδειγμα της εικόνας αριστερά φαίνεται ένα αρχείο PHP στο οποίο ένα τμήμα κώδικα HTML τυπώνεται μόνο αν ικανοποιείται κάποια συνθήκη. Στην εικόνα δεξιά το ίδιο τμήμα κώδικα HTML είναι γραμμένο με τη βοήθεια του Blade.

Αν χρησιμοποιείται απλώς PHP, για να μπορούν να δημιουργούνται δυναμικά τμήματα HTML, πρέπει η HTML να γράφεται μέσα σε εντολές "echo". Όπως φαίνεται στην εικόνα αριστερά, κατ' αυτόν τον τρόπο η πραγματική δομή της HTML «χάνεται» ενώ, επιπλέον, κομμάτια κώδικα PHP και HTML είναι γραμμένα ανάμεικτα. Έτσι ο κώδικας είναι γενικότερα δυσανάγνωστος.

```
if ($enabled) {
    echo "<div class='form-group row'>
          <div class='col-md-4'>
            <input class='form-control' type='text' id='search'
                placeholder='Type to search...'>
          </div>
          <div class='col-md-1 col-md-offset-5'>
          <h4>Show:</h4>
          </div>
          <div class='col-md-1'>
            <select class='form-control' id='exampleSelect1'
                onchange='tablepagition()'>
              <option value='10'>10</option>
              <option value='20' selected>20</option>
              <option value='50'>50</option>
              <option value='100'>100</option>
            </select>
          </div>
        </div>";
}
```

```
@if ($enabled)
    <div class='form-group row'>
        <div class='col-md-4'>
        <input class='form-control' type='text' id='search'
            placeholder='Type to search...'>
        </div>
        <div class='col-md-1 col-md-offset-5'>
        <h4>Show:</h4>
        </div>
        <div class='col-md-1'>
        <select class='form-control' id='exampleSelect1'
                onchange='tablepagination()'>
            <option value='10'>10</option>
            <option value='20' selected>20</option>
            <option value='50'>50</option>
            <option value='100'>100</option>
        </select>
        </div>
    </div>
@endif
```

Εικόνα 4.4: σύγκριση απλού κώδικα (αριστερά)και κώδικα γραμμένο με τη βοήθεια του Blade (δεξιά)

Στην εικόνα δεξιά φαίνεται πώς συντάσσεται ο ίδιος κώδικας HTML με τη βοήθεια του Blade. Βλέπουμε ότι υπάρχει διαθέσιμη μία δομή ελέγχου "@if" η οποία επιτρέπει να γράφεται κώδικας HTML σα να γραφόταν μια αντίστοιχη δομή ελέγχου της PHP. Ωστόσο το πλεονέκτημα είναι σαφές αφού ο κώδικας είναι καθαρή HTML ενώ και η διάρθρωση είναι εντελώς ξεκάθαρη. Κατά τον ίδιο τρόπο λειτουργούν και οι υπόλοιπες δομές ελέγχου που υλοποιεί το Blade.

Όπως γίνεται αντιληπτό, λοιπόν, με τις δυνατότητες που προσφέρει το Blade απλοποιείται σημαντικά η σύνταξη HTML κώδικα και επιπλέον τα αρχεία που προκύπτουν είναι πολύ πιο ευανάγνωστα. Εκτός από τις τρεις πιο σημαντικές δυνατότητές του που αναφέρθηκαν εδώ, το Blade παρέχει αρκετά ακόμη χαρακτηριστικά.

4.5.2 jQuery

Η jQuery είναι μια βιβλιοθήκη της JavaScript η οποία παρέχει δυνατότητες στους προγραμματιστές για πιο αποτελεσματική και ευκολότερη χρήση της γλώσσας. Μάλιστα το σύνθημα "write less, do more" στην ιστοσελίδα της βιβλιοθήκης προσπαθεί να καταδείξει ακριβώς αυτό. Το βασικό της πλεονέκτημα είναι ότι είναι μία βιβλιοθήκη συμπαγής, μικρή σε μέγεθος, γρήγορη και πλούσια σε δυνατότητες [50]. Ορισμένα από τα βασικά σημεία της βιβλιοθήκης είναι [50, 61]:

- να γίνεται ευκολότερα ο χειρισμός των στοιχείων μιας ιστοσελίδας (δηλαδή στοιχείων του DOM – DOM elements),
- να παρέχονται έτοιμες συναρτήσεις που υλοποιούν συνηθισμένες λειτουργίες όπως η απόκρυψη και εμφάνιση ενός στοιχείου, η απόδοση ιδιοτήτων σε ένα στοιχείο HTML (π.χ. κλάσεις), χειρισμός γεγονότων (clicks, mouse overs κτλ.), κλήσεις AJAX κτλ.

- καλύτερη διαχείριση των στυλ (CSS manipulation) και της κίνησης (animation) στα διάφορα στοιχεία των ιστοσελίδων,
- δημιουργία ιστοσελίδων με περισσότερες δυνατότητες για καλύτερο σχεδιασμό διεπαφών χρήστη (user interfaces).

Επίσης, η jQuery, δίνει τη δυνατότητα στους χρήστες της να συμπεριλάβουν τις λειτουργικότητες που αναπτύσσουν σε «πακέτα λειτουργιών» (plugins) τα οποία μετά μπορούν να δημοσιευτούν και να είναι διαθέσιμα σε όλη την κοινότητα χρηστών της βιβλιοθήκης.

4.5.3 Bootstrap

Αντίστοιχα, η Bootstrap είναι ένα πλαίσιο ανάπτυξης ιστοσελίδων (web design framework)[1] η οποία διευκολύνει τη σχεδίαση ιστοσελίδων με έμφαση στη CSS σχεδίαση της ιστοσελίδας και τις δυνατότητες προσαρμοστικότητας των ιστοσελίδων (responsive design) για συσκευές με διαφορετικά μεγέθη οθονών. Έτσι, η βιβλιοθήκη βοηθά το σχεδιαστή να εξασφαλίσει πως η ιστοσελίδα του θα προβάλλεται ευκρινώς και ορθά σε όλους τους χρήστες, ανεξάρτητα από τη συσκευή που χρησιμοποιούν για την περιήγησή τους [51].
Η Bootstrap βασίζεται στην CSS και την JavaScript και οι λειτουργικότητες που παρέχονται αποτελούν συνδυασμό και των δύο. Τα τρία βασικά μέρη λειτουργικότητας που παρέχονται είναι οργανωμένα ως εξής [51, 52]:

- **CSS**: προκαθορισμένες ρυθμίσεις στυλ και προκατασκευασμένα στοιχεία HTML για σχεδιασμό ιστοσελίδων με καλύτερη αισθητική. Η σχεδίαση μόνο με CSS είναι στατική.
- **JavaScript**: στοιχεία διεπαφής χρήστη με ενισχυμένη λειτουργικότητα βασισμένα σε πακέτα λειτουργιών της jQuery. Αποτελούν συνδυασμό CSS και JavaScript. Με αυτά τα στοιχεία, οι σελίδες που σχεδιάζονται είναι πιο διαδραστικές ενώ κάνουν την πλοήγηση του χρήστη πιο «φυσική» και πιο ευχάριστη.
- **Components**: ολοκληρωμένα «κομμάτια» διεπαφής χρήστη (π.χ. λίστες, μενού, επιλογείς, πεδία κειμένου κ.ά.) που είναι έτοιμα προς χρήση. Έχουν κατασκευαστεί ως συνδυασμός CSS και JavaScript, αξιοποιώντας τα δύο μέρη της Bootstrap που αναφέρθηκαν ήδη.

4.6 Γραφήματα & D3.js

4.6.1 Γενικά

Η κατασκευή των γραφημάτων απετέλεσε το τελευταίο προγραμματιστικό κομμάτι της εργασίας. Εδώ χρησιμοποιήθηκε η D3.js (Data Driven Documents – D3), μια βιβλιοθήκη της JavaScript σχεδιασμένη ειδικά για τη δημιουργία γραφημάτων. Η βιβλιοθήκη αυτή διευκολύνει πάρα πολύ την παραγωγή γραφημάτων από δεδομένα καθότι διαθέτει πολύ ισχυρές συναρτήσεις για το χειρισμό των δεδομένων και τη σύνδεσή τους με το παραγόμενο γράφημα. Το θεμελιώδες χαρακτηριστικό της είναι ότι τα γραφήματα επιτρέπουν σημαντική

αλληλεπίδραση με το χρήστη (δηλαδή, δεν είναι στατικές εικόνες που ο χρήστης απλώς τις παρατηρεί) ενώ μπορούν να ενημερώνονται δυναμικά μέσω AJAX και να αλλάζει το γράφημα αμέσως μόλις αλλάξουν τα δεδομένα.

Η ιδέα για μια τέτοια βιβλιοθήκη πρωτοεμφανίστηκε το 2009 με το Protovis, το οποίο ήταν μία προσπάθεια των Michael Bostock και Jeffrey Heer με τη σημαντική συνεισφορά του Vadim Ogievetsky. Η ομάδα ανήκε στο Stanford Visualisation Group του ομώνυμου πανεπιστημίου. Η ανάπτυξη του Protovis σταμάτησε στα μέσα του 2011 όταν και αντικαταστάθηκε από την D3. Ωστόσο στην πραγματικότητα η D3 αποτελεί συνέχεια και επέκταση του Protovis αφού βασίστηκε σε πολύ μεγάλο βαθμό πάνω σε αυτό.

4.6.2 Πλεονεκτήματα & μειονεκτήματα

Η βιβλιοθήκη αυτή είναι ιδιαίτερα ισχυρή για τρεις βασικούς λόγους. Πρώτον, παρέχει απεριόριστη ελευθερία ως προς το τι γράφημα επιθυμεί να δημιουργήσει κάποιος (τα ενδεικτικά παραδείγματα γραφημάτων που μπορεί να βρει κανείς στο Διαδίκτυο με μια απλή αναζήτηση είναι, κυριολεκτικά, αναρίθμητα). Αυτό συμβαίνει επειδή τα γραφήματα δημιουργούνται ως εικόνες SVG οπότε, προφανώς, μπορεί να δημιουργηθεί οποιαδήποτε απεικόνιση επιθυμεί κάποιος αρκεί να ορίσει τα κατάλληλα σχήματα μέσα σε ένα SVG στοιχείο της HTML. Δεύτερον, ο πολύ κομψός τρόπος με τον οποίο η D3 χειρίζεται τα δεδομένα προσδίδει τόσο μεγάλη προγραμματιστική ευελιξία που ο χειρισμός των δεδομένων είναι πάντοτε εύκολος ανεξάρτητα από τον όγκο τους. Τρίτον, τα γραφήματα που προκύπτουν μπορούν να έχουν πολλές δυνατότητες αλληλεπίδρασης με το χρήστη.

Ωστόσο, πιστεύουμε πως η εν λόγω βιβλιοθήκη παρουσιάζει και ένα σημαντικό μειονέκτημα: είναι ένα εργαλείο αρκετά δυσνόητο ως προς τον τρόπο που λειτουργεί, τον τρόπο που χειρίζεται τα δεδομένα για να παραχθούν τα γραφήματα αλλά και ως προς τον τρόπο γραφής του κώδικα. Ακριβώς λόγω όλων αυτών, είναι απολύτως αναγκαίο να κατανοήσει κανείς πλήρως τον τρόπο με τον οποίο λειτουργεί η D3 προτού μπορέσει να τη χρησιμοποιήσει.

Όταν κάποιος χρησιμοποιεί ένα νέο προγραμματιστικό εργαλείο ή μια νέα γλώσσα προγραμματισμού, συνήθως είναι εφικτό να το χρησιμοποιεί και παράλληλα να εξοικειώνεται με τη χρήση του. Ωστόσο στην περίπτωση της D3 αυτό δεν ισχύει κι έτσι η εκμάθησή της καθίσταται αρκετά δύσκολη και χρονοβόρα εν σχέσει με άλλα προγραμματιστικά εργαλεία (π.χ. το Laravel). Το μειονέκτημα έγκειται ακριβώς στο γεγονός ότι δεν πρόκειται για ένα εργαλείο που όσο αφιερώνει χρόνο κάποιος τόσο πιο πολύπλοκα γραφήματα μπορεί να υλοποιήσει. Αντίθετα, χρειάζεται να αφιερώσει ένα ιδιαίτερα σημαντικό χρονικό διάστημα εξοικείωσης με τη βιβλιοθήκη, πριν καν μπορέσει να τη χρησιμοποιήσει για να παράγει έστω ένα πολύ απλό γράφημα.

[1] Η Bootstrap αφορά τη σχεδίαση της ιστοσελίδας γι' αυτό και αναφέρεται ως "web design framework". Η λέξη "framework" δεν πρέπει να ερμηνευθεί με την έννοια του "web application framework", όπως το Laravel.

4.6.3 Πώς λειτουργεί

Όπως υποδεικνύει και το όνομα της βιβλιοθήκης, για να δημιουργήσουμε ένα γράφημα πρέπει να ξεκινήσουμε από το σύνολο δεδομένων (dataset) που θέλουμε να απεικονίσουμε. Τα δεδομένα πρέπει να είναι διαθέσιμα σε κάποια μορφή αρχείου. Για τις κοινές μορφές (π.χ. CSV, TSV) υπάρχουν έτοιμες συναρτήσεις της βιβλιοθήκης που επεξεργάζονται (parse) τα δεδομένα. Κατόπιν η D3 δημιουργεί ένα αντικείμενο (JavaScript object) για κάθε δεδομένο (data point) και το αντικείμενο διαθέτει ως ιδιότητες όλες τις ιδιότητες του δεδομένου με τις αντίστοιχες τιμές τους. Στη συνέχεια κάθε αντικείμενο συνδέεται με ένα στοιχείο που θα προβληθεί στο γράφημα (π.χ. μία στήλη). Έτσι το κάθε στοιχείο του γραφήματος (κάθε στήλη) δεν «απεικονίζει» απλώς το αντίστοιχο δεδομένο αλλά είναι συνδεδεμένο με αυτό· το δεδομένο έγινε αντικείμενο και το αντικείμενο είναι συνδεδεμένο με το γράφημα. Κατόπιν, οι ιδιότητες του γραφήματος (π.χ. το ύψος και το χρώμα της κάθε στήλης) δεν ορίζονται στατικά αλλά εξαρτώνται δυναμικά από τα δεδομένα (π.χ. το ύψος εξαρτάται από την ιδιότητα Α και το χρώμα από την ιδιότητα Β των δεδομένων).

Η εντυπωσιακή δυνατότητα της βιβλιοθήκης είναι ότι επιτρέπει να αλλάζουμε δυναμικά ποια ιδιότητα του γραφήματος συνδέεται με ποια ιδιότητα των δεδομένων. Έστω, για παράδειγμα, ότι θέλουμε να παράγουμε για τα ίδια δεδομένα ένα γράφημα που να απεικονίζει με το ύψος των στηλών την ιδιότητα Γ αντί για την Α. Προφανώς δεν απαιτείται να δημιουργήσουμε άλλο γράφημα ή να φορτώσουμε ξανά τα δεδομένα. Απλώς, με κατάλληλη εντολή, συνδέουμε την ιδιότητα «ύψος» των στηλών με την ιδιότητα Γ των δεδομένων και το γράφημα ανανεώνεται αμέσως. Φυσικά μπορούμε να ανανεώσουμε ολόκληρο το σύνολο δεδομένων μέσω AJAX οπότε κάθε στοιχείο του γραφήματος (κάθε στήλη) θα λάβει το ανανεωμένο δεδομένο που του αντιστοιχεί και το γράφημα θα ενημερωθεί αναλόγως.

Το κυριότερο χαρακτηριστικό της βιβλιοθήκης και το μεγάλο της πλεονέκτημα, λοιπόν, είναι ακριβώς αυτό: ο πολύ κομψός τρόπος με τον οποίο χειρίζεται τα δεδομένα και συνδέει τις ιδιότητες των δεδομένων με τις ιδιότητες του γραφήματος.

Υπενθυμίζουμε πως όλα όσα αναφέραμε εδώ γίνονται από το φυλλομετρητή του χρήστη με JavaScript. Η μόνη αλληλεπίδραση με τον εξυπηρετητή είναι όταν θέλουμε να ανανεώσουμε τα δεδομένα – όχι όταν θέλουμε να ανανεώσουμε το γράφημα. Από τη στιγμή που τα δεδομένα είναι διαθέσιμα, η βιβλιοθήκη μπορεί να τα χειριστεί όπως επιθυμούμε και να αλλάζει στη στιγμή όλη την απεικόνιση.

5

Ανάλυση απαιτήσεων συστήματος

Στο παρόν κεφάλαιο θα καταγραφούν οι απαιτήσεις του συστήματος από την πλευρά του χρήστη, δηλαδή ποιες είναι οι λειτουργίες και οι δυνατότητες που αναμένεται να παρέχει η εφαρμογή. Επίσης, θα διατυπωθούν και ορισμένες προδιαγραφές σε τεχνικό επίπεδο οι οποίες πρέπει να τηρηθούν κατά την ανάπτυξη της εφαρμογής.

5.1 Λειτουργικές απαιτήσεις

Εδώ διατυπώνονται οι απαιτήσεις που έχει από την εφαρμογή ένας χρήστης. Αυτές εξασφαλίζουν ότι η εφαρμογή που θα προκύψει θα παρέχει στους χρήστες τις κατάλληλες λειτουργίες ώστε να είναι πράγματι χρήσιμη. Οι απαιτήσεις ως προς τον τρόπο εξερεύνησης των αλληλεπιδράσεων, λοιπόν, είναι οι εξής:

- Η εφαρμογή πρέπει να υποστηρίζει δύο οργανισμούς, τον άνθρωπο (Homo sapiens) και τον ποντικό (Mus musculus).
- Η εφαρμογή πρέπει να υποστηρίζει την παρουσίαση αλληλεπιδράσεων από τους τρεις επιλεγμένους αλγορίθμους, είτε μεμονωμένα είτε ταυτόχρονα.
- Ο χρήστης πρέπει να μπορεί να αναζητήσει γονίδια και miRNAs είτε με το όνομά τους, είτε με το μοναδικό αναγνωριστικό τους είτε με μέρος αυτών. Κατόπιν πρέπει να υπάρχει τρόπος να επιλεγούν τα γονίδια και τα miRNAs για τα οποία ο χρήστης επιθυμεί να αναζητήσει τις αλληλεπιδράσεις τους.
- Ως προς τις ιδιότητες των αλληλεπιδράσεων, πρέπει να μπορούν να συγκρίνονται οι βαθμολογίες τους ανά αλγόριθμο.

- Τα αποτελέσματα που προκύπτουν πρέπει να μπορούν να περιοριστούν με κατάλληλες επιλογές (φίλτρα). Επιθυμητά φίλτρα είναι τουλάχιστον τα εξής: επιλογή αλγορίθμων και φίλτρο με τη βαθμολογία ανά αλγόριθμο.
- Τα γραφήματα πρέπει να προσφέρουν δυνατότητες προσαρμογής της εμφάνισης έτσι ώστε ο χρήστης να μπορεί να αλληλεπιδρά με τα γραφήματα και να μεταβάλει τα δεδομένα που προβάλλονται. Οι παρεχόμενες δυνατότητες αυτές πρέπει να μπορούν να καταδεικνύουν τις διαφορές στις βαθμολογίες μεταξύ των αλγορίθμων.
- Η εφαρμογή πρέπει για κάθε αντικείμενο (γονίδιο/miRNA) να παρέχει σελίδα που εμφανίζει αναλυτικά τις λεπτομέρειές του.
- Η εφαρμογή πρέπει να παρέχει υπερσυνδέσμους προς τις πηγές των δεδομένων, όπως για παράδειγμα προς την Ensembl ή τους αλγορίθμους πρόβλεψης. Απαραίτητη προϋπόθεση φυσικά είναι να υποστηρίζεται αυτό από τις αντίστοιχες ιστοσελίδες.

5.2 Τεχνικές απαιτήσεις

Πέραν της χρηστικότητας, είναι αναγκαίο να διατυπωθούν ορισμένες απαιτήσεις και σε τεχνικό επίπεδο. Αυτές θα διασφαλίζουν χαρακτηριστικά όπως η εύκολη συντήρηση της εφαρμογής, η επεκτασιμότητα της κτλ. Επομένως, απαιτείται:

- Σε κάθε σημείο της, η υλοποίηση οφείλει να «γενικεύει» τις όποιες απαιτήσεις έχουν τεθεί.
- Πιο συγκεκριμένα, η εφαρμογή πρέπει να μπορεί να υποστηρίξει οποιονδήποτε αριθμό οργανισμών και όχι μόνο δύο. Με άλλα λόγια, η πληροφορία των δύο οργανισμών που θα υποστηρίζονται δεν πρέπει να είναι πουθενά καταγεγραμμένη (hard–coded) στην υλοποίηση της εφαρμογής.
- Αντίστοιχα, η εφαρμογή πρέπει να μπορεί να υποστηρίξει οποιονδήποτε αριθμό αλγορίθμων πρόβλεψης και όχι μόνο τους τρεις που έχουν επιλεχθεί.
- Η εφαρμογή πρέπει να μπορεί να υποστηρίξει επιπλέον μοντέλα που συνδυάζουν τις προβλέψεις των αλγορίθμων (όπως έγινε με το Combo score). Αυτά τα μοντέλα, αν δε διαθέτουν προϋπολογισμένα δεδομένα τα οποία θα προέρχονται από τη βάση δεδομένων, πρέπει να μπορούν να υπολογίζονται από την εφαρμογή τη στιγμή που το ζητά ο χρήστης (on the fly).
- Τα δεδομένα πρέπει να παρέχονται σε όλα τα γραφήματα με τον ίδιο τρόπο. Δηλαδή, η συνάρτηση η οποία θα παράγει και θα ανανεώνει τα δεδομένα των γραφημάτων, πρέπει να είναι η ίδια για όλα τα γραφήματα. Αυτή η απαίτηση καθιστά πολύ εύκολη την προσθήκη επιπλέον γραφημάτων στο μέλλον.

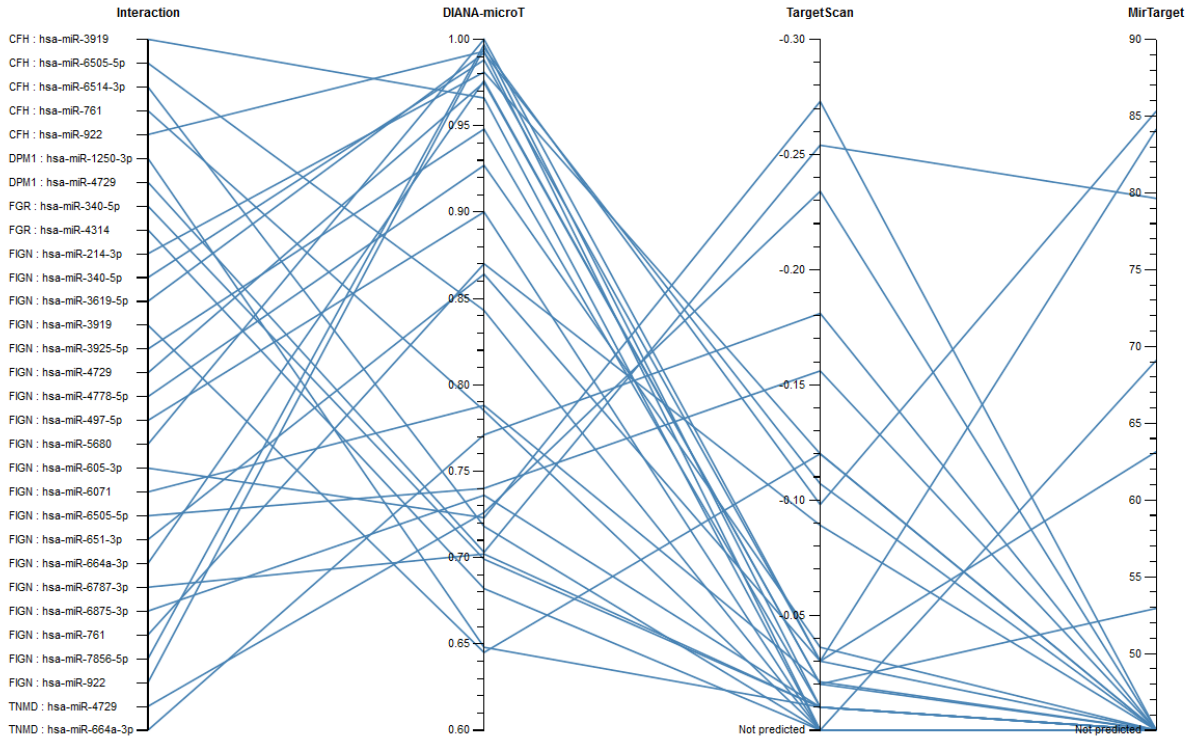
  Παράδειγμα: αν κάποιο γράφημα, για κάποιο λόγο, έχει ιδιαίτερες απαιτήσεις ως προς τη δομή των αποτελεσμάτων, είναι «ευθύνη» του γραφήματος, δηλαδή της JavaScript (client–side), να μετατρέψει τα δεδομένα από τη μορφή που τα έλαβε προς τη μορφή που τα χρειάζεται.
- Από τη στιγμή που η εφαρμογή θα επιτρέπει την είσοδο δεδομένων από το χρήστη, πρέπει να ληφθούν τα κατάλληλα μέτρα για προστασία του συστήματος από κακόβουλη έγχυση κώδικα SQL (SQL injection).

5.3 *Γραφήματα*

Στην ενότητα αυτή εξηγείται για ποιο λόγο κρίνεται χρήσιμο να υλοποιηθούν τα συγκεκριμένα γραφήματα καθώς και τι προσφέρουν στο χρήστη.

5.3.1 *Parallel coordinates*

Στο γράφημα παράλληλων συντεταγμένων (parallel coordinates) κάθε κάθετος άξονας του γραφήματος είναι βαθμονομημένος ξεχωριστά και αντιπροσωπεύει έναν ξεχωριστό αλγόριθμο (από τη διάταξη αυτή προκύπτει και το όνομα του γραφήματος). Το γράφημα αυτό παρέχει τη δυνατότητα στο χρήστη να συγκρίνει εύκολα και γρήγορα πώς κυμαίνεται η βαθμολογία μιας αλληλεπίδρασης σε όλους τους (επιλεγμένους) αλγορίθμους ταυτόχρονα. Επίσης, διευκολύνει την εποπτική παρατήρηση στις διακυμάνσεις βαθμολογιών μεταξύ των αλγορίθμων και μάλιστα για πολλές αλληλεπιδράσεις ταυτοχρόνως.



Εικόνα 5.1: παράδειγμα διαγράμματος παράλληλων συντεταγμένων από την εφαρμογή

Ορισμένα ενδεικτικά παραδείγματα παρατηρήσεων που μπορούν να γίνουν με αυτό το γράφημα είναι:

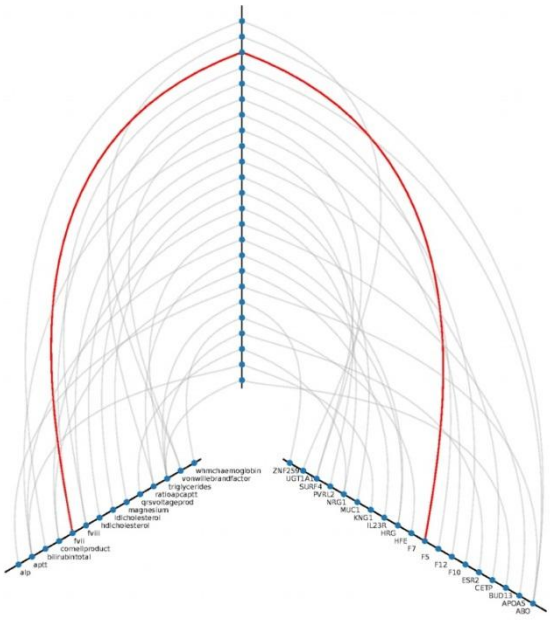
- Να διαπιστωθεί αν οι βαθμολογίες κάποιας συγκεκριμένης ομάδας αλληλεπιδράσεων ακολουθούν κάποια κανονικότητα, π.χ. συγκλίνουν προς κάποια τιμή.
- Να ελεγχθεί κατά πόσον μια υψηλή/χαμηλή βαθμολογία μίας αλληλεπίδρασης από έναν αλγόριθμο διατηρείται και στους υπόλοιπους αλγορίθμους.
- Να παρατηρηθεί αν κάποιος αλγόριθμος βαθμολογεί μια ομάδα αλληλεπιδράσεων συστηματικά χαμηλότερα/υψηλότερα από κάποιον άλλον.

5.3.2 Hive plot

5.3.2.1 Περιγραφή του διαγράμματος

Το διάγραμμα αυτό αποτελεί ένα είδος γράφου και χρησιμοποιείται για την απεικόνιση δικτύων και σχέσεων μεταξύ αντικειμένων. Τα διάφορα αντικείμενα απεικονίζονται ως κόμβοι και οι σχέσεις μεταξύ τους απεικονίζονται ως ακμές. Μία ακμή υφίσταται μεταξύ δύο κόμβων αν τα αντίστοιχα δύο αντικείμενα συνδέονται με κάποια σχέση.

Το στοιχείο που διαφοροποιεί το εν λόγω γράφημα από έναν απλό γράφο είναι ότι χρησιμοποιείται σε περιπτώσεις που οι κόμβοι εντάσσονται σε κατηγορίες. Έτσι, το γράφημα απεικονίζει τους κόμβους επάνω σε άξονες ενώ κάθε άξονας αντιστοιχεί σε συγκεκριμένη κατηγορία. Επιπλέον, εάν οι κατηγορίες μπορούν να ταξινομηθούν με κάποιο τρόπο, τότε η διάταξη των κόμβων επάνω στους άξονες είναι συγκεκριμένη. Βέβαια, το γράφημα αυτό έχει νόημα αν υπάρχουν τουλάχιστον 2 επιμέρους κατηγορίες κατάταξης των κόμβων. Αν όλα τα αντικείμενα ανήκαν σε μία κατηγορία, το εν λόγω γράφημα δε θα είχε να προσφέρει κάτι περισσότερο από έναν απλό γράφο. Ωστόσο, είναι σημαντικό να σημειωθεί ότι δεν αποκλείεται να συνδέονται με ακμή κόμβοι που ανήκουν στην ίδια κατηγορία – κάτι που, βέβαια, στη δική μας περίπτωση δε συμβαίνει.



Εικόνα 5.2: παράδειγμα Hive plot τριών αξόνων

5.3.2.2 Πλεονεκτήματα & αναμενόμενα συμπεράσματα

Το βασικό πλεονέκτημα αυτού του γραφήματος σε σχέση με τους απλούς γράφους είναι ότι πολύ γρήγορα επιτρέπει σε κάποιον να εντοπίσει περιοχές έντονης σύνδεσης μεταξύ των επιμέρους κατηγοριών. Μερικά παραδείγματα στα οποία μπορεί να συνεισφέρει ένα hive plot είναι τα εξής:

- Να εντοπιστεί κατά πόσον οι σχέσεις μιας κατηγορίας με τις υπόλοιπες εκφράζονται κυρίως μέσω συγκεκριμένων κόμβων.

- Η διάταξη των κόμβων επάνω στους άξονες μπορεί να αποκαλύψει κάποιο μοτίβο συσχετίσεων της κατηγορίας με τις υπόλοιπες
- Να εντοπιστούν κόμβοι που ίσως είναι εντελώς ανενεργοί.
- Να εντοπιστούν κόμβοι που έχουν κοινές σχέσεις με άλλους κόμβους.

Έτσι, για την απεικόνιση αλληλεπιδράσεων, το εν λόγω γράφημα εξυπηρετούσε καλύτερα από έναν απλό γράφο, διότι:

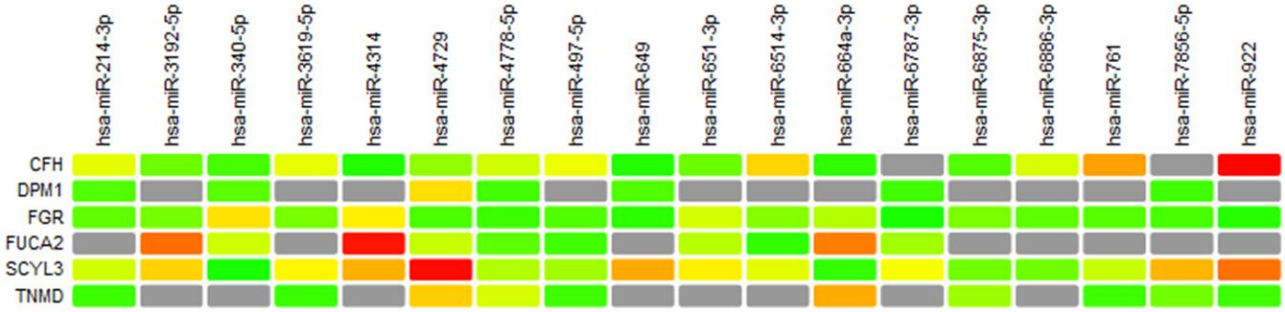
1) Τα αντικείμενα που ενδιαφέρουν εντάσσονται σε διαφορετικές κατηγορίες (γονίδια και miRNAs)
2) Η εν λόγω κατηγοριοποίηση των αντικειμένων σε γονίδια και miRNAs είναι σημαντικό να φαίνεται στο γράφο.
3) Υπάρχει μία σχέση μεταξύ αυτών των κατηγοριών (αλληλεπιδράσεις) η οποία πρέπει να απεικονιστεί.

5.3.3 Heat map

5.3.3.1 Περιγραφή του γραφήματος

Ένα γράφημα heat map είναι ένας χάρτης που αποτελείται από γραμμές και στήλες που σχηματίζουν κελιά. Κάθε κελί αντιστοιχεί σε μία γραμμή και μία στήλη υπονοώντας ότι το εν λόγω κελί απεικονίζει πληροφορία που αφορά τη σχέση της γραμμής και της στήλης.

Από μία άλλη σκοπιά, αντιλαμβάνεται κανείς ότι ένα heat map ουσιαστικά αποτελεί έναν πίνακα γειτνίασης κόμβων επάνω σε ένα γράφο. Η διαφορά με έναν πίνακα γειτνίασης είναι ότι αντί στα κελιά να αναγράφονται αριθμοί (π.χ. 1/0 ή το κόστος της ακμής) αναπαρίσταται η ανάλογη πληροφορία χρωματικά.



Εικόνα 5.3: παράδειγμα Heat map από την εφαρμογή

Στην περίπτωσή μας, κάθε κελί αναπαριστά μία αλληλεπίδραση. Αν η αλληλεπίδραση υφίσταται τότε το κελί χρωματίζεται ανάλογα με τη βαθμολογία της αλληλεπίδρασης ενώ αν η αλληλεπίδραση δεν υφίσταται το κελί χρωματίζεται γκρι. Προφανώς, για να αποδίδεται σωστά η ανωτέρω διαφοροποίηση το γκρι δεν περιλαμβάνεται στην κλίμακα χρωμάτων ανάλογα με τη βαθμολογία.

Ένα επιπλέον βήμα που έγινε στην εργασία ώστε να ενισχύεται η χρησιμότητα των heat maps είναι να παρουσιάζονται ταυτόχρονα και να είναι συνδεδεμένα μεταξύ τους περισσότερα από ένα heat maps. Κάθε

heat map μπορεί να χρωματίζεται με βάση διαφορετικό αλγόριθμο έτσι ώστε σε μία οθόνη ο χρήστης να συγκρίνει την ίδια στιγμή τους «χάρτες» που προκύπτουν από όλους τους αλγορίθμους.

5.3.3.2 Πλεονεκτήματα & αναμενόμενα συμπεράσματα

Μερικές ενδεικτικές παρατηρήσεις και τα αντίστοιχα συμπεράσματα που διευκολύνομαι από τα heat maps είναι τα εξής:

- Να εντοπιστούν περιοχές ή μεμονωμένα κελιά που παρουσιάζουν έντονη διαφοροποίηση. Π.χ. ιδιαίτερα καλές ή ιδιαίτερα κακές βαθμολογίες.
- Να αλλάζει η ταξινόμηση των γραμμών και των στηλών έτσι ώστε να εντοπιστούν κανονικότητες. Π.χ. να εντοπιστεί αν κατά μήκος μιας γραμμής/στήλης μεταβάλλονται προς το καλύτερο ή χειρότερο οι χρωματισμοί.
- Το γεγονός ότι η εφαρμογή υποστηρίζει πολλαπλούς χάρτες ταυτόχρονα βοηθά στο εξής: αν εντοπίζεται κάτι από τα παραπάνω, να συγκρίνεται άμεσα με έναν χάρτη διαφορετικού αλγορίθμου. Έτσι μπορεί να διαπιστωθεί αν η εν λόγω κανονικότητα εντοπίζεται μόνο σε έναν αλγόριθμο ή σε περισσότερους.

6

Σχεδιασμός και υλοποίηση της εφαρμογής

Στόχος του παρόντος κεφαλαίου είναι να παρουσιάσει τη διαδικασία υλοποίησης της εφαρμογής. Οι ενότητες ακολουθούν κατά σειρά τη χρονολογική πορεία της υλοποίησης ώστε να διευκολυνθεί ο αναγνώστης να παρακολουθήσει τη διαδικασία που ακολουθήθηκε. Στο παρόν κεφάλαιο περιγράφεται το «πώς» έγινε η υλοποίηση και όχι το «γιατί» έγινε κατ' αυτόν τον τρόπο. Η αιτιολόγηση των διάφορων σχεδιαστικών επιλογών, όπου υπήρξε ζήτημα, δίνεται αναλυτικά στο κεφάλαιο 7.

Παρακάτω παραθέτουμε έναν πίνακα με τη διαμόρφωση του εξυπηρετητή και τα πακέτα λογισμικού που χρησιμοποιήθηκαν.

Είδος	Λογισμικό	Έκδοση
Λειτουργικό σύστημα	Ubuntu	14.04 LTS
Πρόγραμμα εξυπηρετητή (server)	Apache server	2.4
Βάση δεδομένων	MySQL	5.5
Γλώσσα προγραμματισμού (server side scripting)	PHP	7.0
Πλαίσιο ανάπτυξης εφαρμογής (PHP framework)	Laravel	5.3
Βιβλιοθήκη κατασκευής γραφημάτων	D3	3

Πίνακας 6.1: η διαμόρφωση του εξυπηρετητή (software backbone)

6.1 Συλλογή δεδομένων

Η πρώτη εργασία που έπρεπε να γίνει ήταν να συλλεχθούν τα δεδομένα στα οποία θα βασιστεί η εφαρμογή. Πέραν του προφανούς, αφού χωρίς δεδομένα δεν μπορεί να υφίσταται εφαρμογή, αυτό το βήμα ήταν προαπαιτούμενο και για το σωστό σχεδιασμό της βάσης δεδομένων. Χωρίς να είναι γνωστό τι δεδομένα είναι διαθέσιμα από τις βιολογικές βάσεις δεδομένων και τους αλγορίθμους πρόβλεψης, σε τι μορφή είναι διαθέσιμα αυτά τα δεδομένα, τι πεδία περιλαμβάνουν κτλ. δεν θα ήταν εφικτό να γίνει σωστός σχεδιασμός της βάσης δεδομένων. Άρα, το κατέβασμα των δεδομένων, έπρεπε να είναι η πρώτη εργασία η οποία θα γίνει.

Τα δεδομένα πάνω στα οποία στηρίζεται η εφαρμογή διακρίνονται σε δύο μεγάλα σύνολα:

- δεδομένα για τα γονίδια και τα miRNAs από τις αντίστοιχες βιολογικές βάσεις δεδομένων,
- δεδομένα για τις αλληλεπιδράσεις γονιδίων – miRNAs από τους αλγορίθμους πρόβλεψης.

Για όσες πηγές δεν αναφέρεται κάτι ιδιαίτερο σχετικά με το είδος του οργανισμού σημαίνει πως η διαδικασία συλλογής δεδομένων και για τους δύο οργανισμούς είναι ακριβώς η ίδια. Όπου υπάρχουν διαφοροποιήσεις, επισημαίνονται.

6.1.1 Ensembl

6.1.1.1 Δεδομένα γονιδίων

Από την Ensembl προήλθαν όλα τα δεδομένα σχετικά με τα γονίδια. Για το κατέβασμα των δεδομένων αξιοποιήθηκε το BioMart που είναι διαθέσιμο στην ιστοσελίδα της Ensembl. Τα στοιχεία που αντλήθηκαν για το κάθε γονίδιο ήταν τα εξής:

Πεδίο	Ιδιότητα γονιδίου
Gene ID	μοναδικό αναγνωριστικό
Description	περιγραφή
Chromosome	χρωμόσωμα στο οποίο βρίσκεται το γονίδιο
Associated gene name	όνομα
Version (gene)	έκδοση

Πίνακας 6.2: τα περιεχόμενα του αρχείου δεδομένων της Ensembl

Αποκτήθηκε η πλήρης λίστα γονιδίων για κάθε έκδοση της Ensembl από την έκδοση 86 μέχρι την 75.

6.1.1.2 Δεδομένα αντιστοιχίσεων Ensembl προς RefSeq

Όπως θα εξηγηθεί παρακάτω, λόγω της μορφής των αποτελεσμάτων του MirTarget έπρεπε να βρεθούν αντιστοιχίες στα μετάγραφα μεταξύ των μοναδικών αναγνωριστικών της Ensembl και της RefSeq. Τα

δεδομένα που δίνουν τις αντιστοιχίες αυτές αντλήθηκαν επίσης με χρήση του BioMart. Τα αρχεία που δημιουργήθηκαν για αυτό το σκοπό περιείχαν τα εξής πεδία:

Πεδίο	Επεξήγηση
Gene ID	γονίδιο
Transcript ID	μετάγραφο
RefSeq mRNA	Πιθανό αντίστοιχο μετάγραφο RefSeq 1
RefSeq ncRNA	Πιθανό αντίστοιχο μετάγραφο RefSeq 2

Πίνακας 6.3: τα περιεχόμενα του αρχείου αντιστοιχιών όπως προήλθε από την Ensembl

Αντιστοιχίες αντλήθηκαν από τις εκδόσεις 75 έως 86.

Κάθε μετάγραφο που είναι καταγεγραμμένο στην Ensembl μπορεί να έχει ή να μην έχει αντίστοιχο μετάγραφο στην RefSeq. Αν υπάρχει αντίστοιχο μετάγραφο, τότε μπορεί να είναι:

- **mRNA**: messenger RNA, δηλαδή RNA που κωδικοποιεί πρωτεΐνη.
- **ncRNA**: non coding RNA, δηλαδή RNA που δεν κωδικοποιεί πρωτεΐνη.

Αν υπάρχει αντιστοιχία, μόνο ένα από τα δύο πεδία της RefSeq μπορούν να έχουν τιμή και όχι και τα δύο (άλλωστε ένα RNA είτε κωδικοποιεί πρωτεΐνη είτε όχι). Αν δεν υπάρχει αντιστοιχία, και τα δύο πεδία της RefSeq θα είναι κενά. Για όσες αντιστοιχίες βρεθούν, σκοπός είναι τα μετάγραφα της RefSeq να αντιστοιχηθούν με ένα γονίδιο της Ensembl.

Σημειώνεται πως οι αντιστοιχίες που καταγράφονται σε αυτό το αρχείο ορίζονται από την Ensembl.

6.1.2 RefSeq

Για να αντληθούν οι αντιστοιχίες μεταξύ RefSeq και Ensembl ήταν απαραίτητο και ένα ακόμη αρχείο από τη RefSeq το οποίο βρίσκεται στη διεύθυνση: ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2ensembl.gz

Η μορφή του αρχείου ήταν προκαθορισμένη και περιέχονταν τα εξής πεδία:

Πεδίο	Επεξήγηση
Taxonomy ID	Κωδικός οργανισμού
RefSeq gene ID	Αναγνωριστικό γονιδίου κατά RefSeq
Ensembl gene ID	Αναγνωριστικό γονιδίου κατά Ensembl
RefSeq RNA ID	Αναγνωριστικό RNA κατά RefSeq
Ensembl transcript ID	Αναγνωριστικό RNA κατά Ensembl
RefSeq Protein ID	Αναγνωριστικό πρωτεΐνης κατά RefSeq
Ensembl peptide ID	Αναγνωριστικό πρωτεΐνης κατά Ensembl

Πίνακας 6.4: τα περιεχόμενα του αρχείου αντιστοιχιών όπως προήλθε από την RefSeq

Τα πεδία σχετικά με τις πρωτεΐνες δεν χρειάζονταν οπότε αγνοήθηκαν. Με τα δεδομένα που υπάρχουν εδώ, για κάθε μετάγραφο της RefSeq μπορούμε να δούμε σε ποιο γονίδιο της Ensembl αντιστοιχεί, το ίδιο

ακριβώς, δηλαδή, που επιθυμούσαμε να διαπιστώσουμε και με το αντίστοιχο αρχείο της Ensembl. Ο λόγος που λήφθηκαν δεδομένα και από τα δύο αρχεία θα εξηγηθεί κατά την προεπεξεργασία των δεδομένων.

6.1.3 miRBase

Το αρχείο της miRBase με τα δεδομένα των miRNAs ελήφθη από την τοποθεσία ftp://mirbase.org/pub/mirbase/CURRENT/miRNA.dat.gz.
Το αρχείο βρίσκεται σε EMBL flat–file format, όπως περιγράφηκε ήδη αναλυτικά στην παράγραφο 3.2.4.

6.1.4 DIANA–microT

6.1.4.1 Απόκτηση αρχείου

Τα δεδομένα του αλγορίθμου αυτού, όπως και των περισσότερων αλγορίθμων γενικά στο Διαδίκτυο, είναι ελεύθερα διαθέσιμα. Ωστόσο για να μπορεί ο χρήστης να τα κατεβάσει πρέπει να κάνει εγγραφή στην ιστοσελίδα του αλγορίθμου. Η εγγραφή είναι ελεύθερη. Αρχικά, λοιπόν, επισκεπτόμαστε την ιστοσελίδα του αλγορίθμου στη διεύθυνση:
http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=microT_CDS/index
και κατόπιν πρέπει να πραγματοποιηθεί είσοδος του χρήστη (login) στο σύστημα. Στη συνέχεια επιλέγεται το πλήκτρο “Download data area” και από τα διαθέσιμα αρχεία επιλέγεται το “microT-CDS data”.

6.1.4.2 Μορφή αρχείου

Αρχικά σημειώνεται πως όλες οι αλληλεπιδράσεις για όλους τους οργανισμούς περιέχονται μέσα σε ένα αρχείο. Παρατίθεται ένα τμήμα του αρχείου για καλύτερη κατανόηση.
Το αρχείο του DIANA–microT είναι ένα CSV αρχείο κειμένου (comma separated values). Οι πληροφορίες που περιέχονται σε κάθε γραμμή του αρχείου είναι οργανωμένες σε πεδία όπου τα πεδία μεταξύ τους διαχωρίζονται με κόμμα (εξ ου και ο τύπος του αρχείου). Το αρχείο είναι οργανωμένο σε «εγγραφές» όπου κάθε εγγραφή αντιπροσωπεύει μία αλληλεπίδραση. Όπως φαίνεται στο παράδειγμα, κάθε εγγραφή ξεκινάει με μία γραμμή που έχει 4 πεδία τα οποία κατά σειρά είναι:

- Το μετάγραφο της αλληλεπίδρασης. Δίνεται το Ensembl ID του.
- Το γονίδιο της αλληλεπίδρασης, δηλαδή το γονίδιο απ’ το οποίο προκύπτει το μετάγραφο. Δίνεται το Ensembl ID του και μέσα σε παρένθεση το σύμβολο (όνομα) του γονιδίου.
- Το όνομα του miRNA της αλληλεπίδρασης.
- Η βαθμολογία της αλληλεπίδρασης.

Στη συνέχεια ακολουθούν ορισμένες γραμμές οι οποίες αντιστοιχούν στις θέσεις πρόσδεσης της αλληλεπίδρασης αυτής. Κάθε αλληλεπίδραση θα περιέχει τουλάχιστον μία θέση πρόσδεσης ενώ, οι περισσότερες αλληλεπιδράσεις, εμφανίζουν περισσότερες. Κάθε θέση πρόσδεσης αντιπροσωπεύεται από μία γραμμή με τρία πεδία τα οποία κατά σειρά είναι τα εξής:

- Η περιοχή του μεταγράφου στην οποία βρίσκεται η θέση πρόσδεσης (UTR 3΄, UTR 5΄, CDS).
- Στο δεύτερο πεδίο αναγράφεται πρώτα το χρωμόσωμα στο οποίο ανήκει το γονίδιο. Ακολουθεί άνω κάτω τελεία « : ». Μετά αναγράφεται η αρχή και το πέρας της θέσης πρόσδεσης διαχωρισμένες με παύλα «–». Οι συντεταγμένες είναι εκπεφρασμένες επάνω στο γονίδιο και όχι επάνω στο μετάγραφο.
- Η επιμέρους βαθμολογία αυτής της θέσης πρόσδεσης.

6.1.5 TargetScan

6.1.5.1 Απόκτηση αρχείων

Και για τους δύο οργανισμούς χρησιμοποιήθηκαν τόσο οι προβλέψεις του αλγορίθμου για θέσεις πρόσδεσης που διατηρούνται μεταξύ των οργανισμών (conserved sites) όσο και για θέσεις πρόσδεσης που δεν διατηρούνται (non conserved sites). Παρακάτω καταγράφεται ποια αρχεία ελήφθησαν για τον άνθρωπο και τον ποντικό.

- **Άνθρωπος**
  Διεύθυνση: http://www.targetscan.org/cgi-bin/targetscan/data_download.vert71.cgi
  Αρχεία: Predicted Targets context++ scores (default predictions), Conserved site context++ scores, Nonconserved site context++ scores
- Ποντικός
  Διεύθυνση: http://www.targetscan.org/cgi-bin/targetscan/data_download.cgi?db=mmu_71
  Αρχεία: Conserved site context++ scores, Nonconserved site context++ scores

6.1.5.2 Μορφή αρχείων

Τα αρχεία του TargetScan είναι αρχεία κειμένου TSV (tab separated values) και επί της ουσίας είναι ένας πίνακας του οποίου οι στήλες είναι διαχωρισμένες με το χαρακτήρα στηλοθέτη (tab). Η κάθε γραμμή του πίνακα αντιπροσωπεύει μία αλληλεπίδραση και περιέχει αρκετά πεδία. Αυτά που αξιοποιήθηκαν από εμάς είναι τα εξής:

- **Gene ID**: το γονίδιο της αλληλεπίδρασης. Δίνεται με το Ensembl ID του.
- **miRNA**: το miRNA της αλληλεπίδρασης. Δίνεται με το όνομά του.
- **context++ score**: η βαθμολογία της αλληλεπίδρασης.

Σημειώνεται εδώ πως, ο TargetScan, παρέχει προβλέψεις αλληλεπιδράσεων των ανθρώπινων γονιδίων τόσο με ανθρώπινα miRNAs όσο και με miRNAs άλλων οργανισμών. Το δεύτερο σκέλος είναι χρήσιμο για συγκριτικές μελέτες μεταξύ οργανισμών. Για παράδειγμα, μπορεί κάποιος να μελετήσει τις αλληλεπιδράσεις του ανθρώπινου γονιδίου TNMD με ένα συγκεκριμένο ανθρώπινο miRNA αλλά και με το αντίστοιχο miRNA του ποντικού και στη συνέχεια να μελετήσει τις αλληλεπιδράσεις του γονιδίου TNMD από ποντικό με αυτά τα miRNAs. Έτσι μπορούν να εξαχθούν κατάλληλα συμπεράσματα για τη διατήρηση (conservation) των διάφορων θέσεων πρόσδεσης μεταξύ των οργανισμών, για ομοιότητες και διαφορές των αντίστοιχων

γονιδίων και miRNAs μεταξύ των οργανισμών κτλ. Μάλιστα, στη σελίδα του TargetScan με τις συχνές ερωτήσεις (FAQ - http://www.targetscan.org/faqs.Release_7.html) υπάρχει αντίστοιχη ερώτηση σχετικά με τις διαφορές των προβλέψεων για τον ποντικό αν αυτές ληφθούν από τα αποτελέσματα για τον άνθρωπο (TargetScanHuman) ή από τα αποτελέσματα για τον ποντικό (TargetScanMouse). Σε αυτό ακριβώς το σημείο αξιοποιούνται προβλέψεις μεταξύ ανθρώπινων γονιδίων με miRNAs από άλλους οργανισμούς.
Επειδή σε αυτή την εργασία δεν έγιναν συγκρίσεις μεταξύ οργανισμών, λήφθηκαν υπ' όψη μόνο οι αλληλεπιδράσεις μεταξύ γονιδίων και miRNAs από τον ίδιο οργανισμό.

6.1.6 MirTarget

Τα αποτελέσματα του MirTarget υπάρχουν διαθέσιμα για κατέβασμα στη διεύθυνση: http://mirdb.org/miRDB/download.html. Το αρχείο είναι ένα TSV αρχείο με τρεις στήλες:

- Όνομα miRNA.
- Μετάγραφο, δοσμένο με το RefSeq ID του.
- Βαθμολογία της αλληλεπίδρασης.

Τα αποτελέσματα όλων των οργανισμών βρίσκονται στο ίδιο αρχείο.

6.1.7 Εκδόσεις των πηγών δεδομένων που χρησιμοποιήθηκαν στην εργασία

Όλα τα δεδομένα αποκτήθηκαν την 1/12/2016. Στον παρακάτω πίνακα αναγράφεται συγκεντρωτικά, για όσες πηγές διαθέτουν εκδόσεις, ποια έκδοση χρησιμοποιήθηκε. Για την RefSeq και τον DIANA–microT σημείο αναφοράς είναι η ημερομηνία ανάκτησης των δεδομένων.

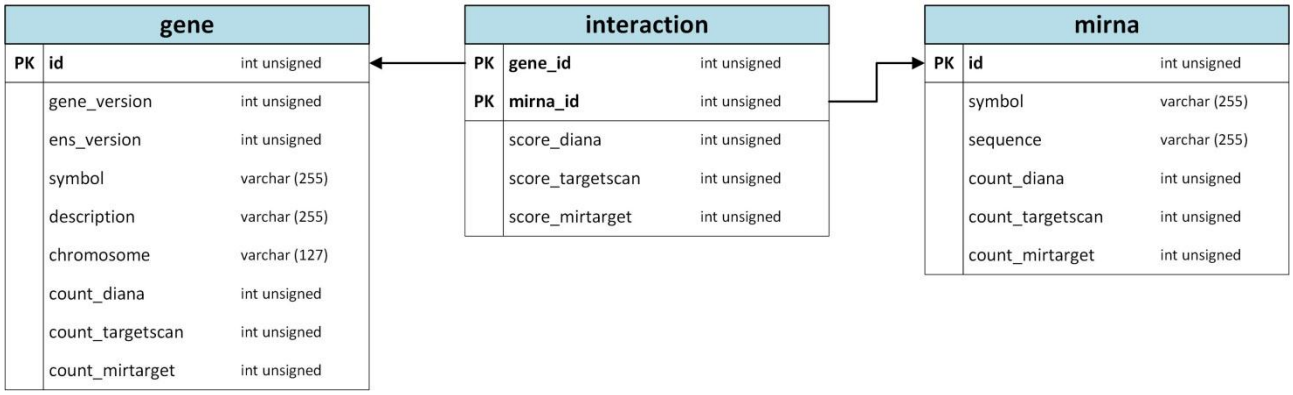
Πηγή	Έκδοση
Ensembl	75 έως 86
miRBase	21
TargetScan	7.1
MirTarget	5

Πίνακας 6.5: οι εκδόσεις των πηγών που χρησιμοποιήθηκαν

6.2 *Σχεδιασμός βάσης δεδομένων*

6.2.1 *Σχήμα της βάσης δεδομένων*

Στην εικόνα φαίνεται το σχήμα της βάσης δεδομένων, με τους τρεις πίνακες της βάσης, τα πρωτεύοντα κλειδιά (primary keys) κάθε πίνακα καθώς και τα ξένα κλειδιά (foreign keys). Με βάση το σχήμα αυτό, δημιουργείται μία βάση δεδομένων για κάθε οργανισμό. Όλες οι βάσεις δεδομένων φιλοξενούνται στην ίδια εγκατάσταση της MySQL.



Εικόνα 6.1: το σχήμα της βάσης δεδομένων

6.2.2 *Πίνακας gene*

Το σχήμα του πίνακα gene είναι το εξής:

- **id**: το μοναδικό αναγνωριστικό του κάθε γονιδίου.
- **gene_version**: η έκδοση στην οποία βρίσκεται το γονίδιο για την έκδοση της Ensembl από την οποία ελήφθησαν τα δεδομένα του.
- **ens_version**: η πιο πρόσφατη έκδοση της Ensembl η οποία περιέχει το γονίδιο και από την οποία ελήφθησαν τα δεδομένα για το γονίδιο αυτό.
- **symbol**: το όνομα του γονιδίου. Ως όνομα του πεδίου χρησιμοποιήθηκε εσκεμμένα το "symbol" αντί για "name", καθώς το "name" είναι δεσμευμένη λέξη της MySQL.
- **description**: η περιγραφή του γονιδίου.
- **chromosome**: το χρωμόσωμα στο οποίο βρίσκεται το γονίδιο. Η πληροφορία αυτή ήταν αναγκαία για να διακρίνονται περιπτώσεις αλληλόμορφων γονιδίων.
- **count_diana**: το πλήθος αλληλεπιδράσεων που ο DIANA-microT προβλέπει για αυτό το γονίδιο.
- **count_targetscan**: ομοίως για τον TargetScan.
- **count_mirtarget**: ομοίως για τον MirTarget.

6.2.2.1 Επιλογή πρωτεύοντος κλειδιού

Για τον πίνακα gene έπρεπε να επιλεγεί ένα πρωτεύον κλειδί που θα προσδιόριζε μοναδικά κάθε γονίδιο της βάσης δεδομένων. Η πρώτη και πιο απλή σκέψη ήταν να αποδοθεί ένας αύξων αριθμός κατά σειρά σε όλες τις εγγραφές του πίνακα, να δημιουργηθεί δηλαδή ένα «εσωτερικό αναγνωριστικό» (internal ID). Αυτή η επιλογή όμως παρουσιάζει μία βασική περιπλοκή. Οι αναφορές από τους αλγορίθμους πρόβλεψης προς τα γονίδια γίνονται με βάση το Ensembl ID. Αν επιλέγαμε να αποδώσουμε ένα καινούριο αύξοντα αριθμό στα γονίδια, θα έπρεπε κατά την προετοιμασία των δεδομένων να δημιουργηθεί ένα ενδιάμεσο βήμα «μετάφρασης» από το Ensembl ID προς το νέο εσωτερικό αναγνωριστικό. Αυτό το επιπλέον βήμα μετάφρασης, όμως, εισάγει σημαντική πιθανότητα σφάλματος διότι θα έπρεπε να μεταφραστούν από το Ensembl ID προς το νέο αναγνωριστικό όλα τα αρχεία δεδομένων που έχουν συλλεχθεί.
Εκτός αυτού, μοιάζει περιττό να ορίσουμε καινούριο μοναδικό αύξοντα αριθμό αφού, στην πραγματικότητα, τα γονίδια ήδη διαθέτουν μοναδικό αναγνωριστικό από την Ensembl. Επομένως επιλέχθηκε να αξιοποιηθούν τα Ensembl IDs ως πρωτεύοντα κλειδιά για τον πίνακα gene.

6.2.2.2 Παραγωγή πρωτεύοντος κλειδιού από τα Ensembl IDs

Ωστόσο και αυτή η λύση έχει μία δυσκολία: τα Ensembl IDs είναι συμβολοσειρές (strings) και μάλιστα μεγάλου μήκους (16 χαρακτήρες για τον άνθρωπο, 19 για τον ποντικό) ενώ είναι γνωστό πως, για λόγους ταχύτητας της βάσης δεδομένων, δεν προτιμάται να υπάρχουν συμβολοσειρές ως πρωτεύοντα κλειδιά. Όμως, όπως εξηγήθηκε στο κεφάλαιο 3, τα Ensembl IDs έχουν ένα σταθερό μέρος με γράμματα (ENSG για τον άνθρωπο, ENSMUSG για τον ποντικό) και ένα μέρος που αποτελεί καθαρό αριθμό ο οποίος, μάλιστα, είναι μοναδικός ανά οργανισμό και είδος βιομορίου. Άρα, για κάθε γονίδιο, θα απομονωθεί το αριθμητικό μέρος του Ensembl ID, θα αγνοηθούν τα αρχικά μηδενικά και ο ακέραιος που προκύπτει θα αποτελέσει το πεδίο "id" και πρωτεύον κλειδί για τον πίνακα gene.
Παράδειγμα: το μοναδικό αναγνωριστικό του γονιδίου FIGN είναι το ENSG00000182263. Άρα ως "id" για το γονίδιο αυτό θα χρησιμοποιηθεί ο αριθμός 182263.

Σημειώνουμε πως το σκεπτικό αυτό είναι εφικτό να εφαρμοστεί ανεξαρτήτως της επιλογής για ξεχωριστές βάσεις δεδομένων ανά οργανισμό ή όχι. Όπως είπαμε, ο 12ψήφιος αριθμός ενός Ensembl ID είναι μοναδικός ανά οργανισμό και είδος βιομορίου. Ακόμη και σε μία ενιαία βάση δεδομένων, όμως, οι πίνακες human_gene και mouse_gene και πάλι θα ήταν διαφορετικοί. Οπότε η πιθανότητα να υπάρχει ίδιος αριθμός για ένα γονίδιο ανθρώπου και ένα γονίδιο ποντικού δεν ενοχλεί σε καμία περίπτωση.

6.2.2.3 Πλήθος αλληλεπιδράσεων

Η εφαρμογή θα παρέχει μία σελίδα λεπτομερειών για κάθε γονίδιο στην οποία θα παρουσιάζονται τα λεπτομερή στοιχεία του γονιδίου. Μία τέτοια πληροφορία θέλουμε να είναι και το πλήθος των αλληλεπιδράσεων που ο εκάστοτε αλγόριθμος προβλέπει για το γονίδιο αυτό. Ο τρόπος να βρεθεί αυτό από

τον πίνακα interaction είναι να εκτελεστεί κατάλληλο ερώτημα SQL που μετρά πόσες εγγραφές του πίνακα interaction που περιέχουν αυτό το γονίδιο προβλέπονται από τον κάθε αλγόριθμο.
Όμως παρατηρήθηκε πως, αυτού του είδους τα ερωτήματα, αναλόγως με το πλήθος των αλληλεπιδράσεων ενός γονιδίου, στη μέση περίπτωση χρειάζονταν 10-12 δευτερόλεπτα να ολοκληρωθούν. Αυτό το χρονικό διάστημα είναι πολύ μεγάλο δεδομένης της συχνότητας με την οποία οι χρήστες θα επισκέπτονται τη σελίδα λεπτομερειών των γονιδίων. Επίσης, πρόκειται για μία πληροφορία η οποία δεν αλλάζει παρά μόνο αν ανανεωθούν τα δεδομένα ολόκληρης της βάσης δεδομένων – άρα εκτελούνται τα ίδια ερωτήματα με ίδιο ακριβώς αποτέλεσμα κάθε φορά. Ακόμη, αυτά τα ερωτήματα αποτελούν σημαντικό φόρτο για τη βάση δεδομένων – ο πίνακας interaction περιέχει 23,6 εκατομμύρια εγγραφές για τον άνθρωπο και 17,8 εκατομμύρια εγγραφές για τον ποντικό. Άρα αυτά τα «ερωτήματα καταμέτρησης» είναι αρκετά επίπονα για τη βάση δεδομένων τη στιγμή που η βάση καλείται να εξυπηρετήσει πιο «σημαντικά» ερωτήματα που σχετίζονται με την εξερεύνηση αλληλεπιδράσεων.
Για αυτό το λόγο επιλέχθηκε αυτή η πληροφορία να προϋπολογιστεί κατά την προετοιμασία των δεδομένων για κάθε γονίδιο και να καταγραφεί στη βάση δεδομένων (hardcoded) παρά το γεγονός πως μπορεί να εξαχθεί με κατάλληλο SQL ερώτημα. Το πλήθος αλληλεπιδράσεων ανά αλγόριθμο καταγράφεται στα πεδία count_diana, count_targetscan και count_mirtarget.

6.2.3 Πίνακας mirna

Το σχήμα του πίνακα mirna έχει ως εξής:

- **id**: το μοναδικό αναγνωριστικό κάθε miRNA.
- **symbol**: το όνομα του miRNA.
- **sequence**: η ακολουθία του miRNA.
- **count_diana**: το πλήθος αλληλεπιδράσεων που ο DIANA-microT προβλέπει για αυτό το miRNA.
- **count_targetscan**: ομοίως για τον TargetScan.
- **count_mirtarget**: ομοίως για τον MirTarget.

Παρατηρήστε πως υπάρχουν και εδώ προϋπολογισμένα τα πλήθη αλληλεπιδράσεων που ο εκάστοτε αλγόριθμος προβλέπει για το κάθε miRNA. Η λογική αυτής της επιλογής είναι επακριβώς η ίδια όπως αναλύθηκε και για τα γονίδια προηγουμένως.

6.2.3.1 Επιλογή πρωτεύοντος κλειδιού και παραγωγή του από τα miRBase accession numbers

Το σκεπτικό για το πρωτεύον κλειδί αυτού του πίνακα είναι επακριβώς το ίδιο όπως και για το πρωτεύον κλειδί του πίνακα gene. Εδώ επιλέχθηκε τα μοναδικά αναγνωριστικά της miRBase (miRBase accession numbers) να λειτουργήσουν ως πρωτεύοντα κλειδιά για τα miRNAs.
Τα αναγνωριστικά της miRBase περιέχουν ένα σταθερό μέρος με γράμματα (MIMAT) και ένα μέρος που αποτελεί καθαρό 7ψήφιο αριθμό. Για κάθε miRNA θα απομονωθεί ο 7ψήφιος αριθμός χωρίς τα αρχικά μηδενικά και ο ακέραιος που θα προκύψει θα αποτελέσει το πεδίο “id” και πρωτεύον κλειδί για τον πίνακα mirna.

Παράδειγμα: το μοναδικό αναγνωριστικό του miRNA has-miR-21-3p είναι το MIMAT0004494. Άρα ως "id" για το miRNA αυτό θα χρησιμοποιηθεί ο αριθμός 4494.

6.2.4 Πίνακας interaction

Το σχήμα του πίνακα interaction έχει ως εξής:

- **gene_id**: το ID του γονιδίου που συμμετέχει στην αλληλεπίδραση.
- **mirna_id**: το ID του miRNA που συμμετέχει στην αλληλεπίδραση.
- **score_diana**: η βαθμολογία της αλληλεπίδρασης από τον αλγόριθμο DIANA-microT ή NULL αν δεν προβλέπεται.
- **score_targetscan**: ομοίως για τον TargetScan.
- **score_mirtarget**: ομοίως για τον MirTarget.

6.2.4.1 Πρωτεύον κλειδί και ξένα κλειδιά

Πρωτεύον κλειδί αποτελεί το ζεύγος gene_id, mirna_id καθώς κάθε αλληλεπίδραση πρέπει να είναι μοναδική.
Υπάρχουν δύο ξένα κλειδιά στον πίνακα, από το gene_id προς το πεδίο gene.id και, αντίστοιχα, από το mirna_id προς το πεδίο mirna.id. Ο περιορισμός είναι προφανής, δηλαδή το γονίδιο και το miRNA μιας αλληλεπίδρασης πρέπει να υπάρχουν στους αντίστοιχους πίνακες.

6.3 Προεπεξεργασία δεδομένων (data clean–up)

Έως εδώ έχουν συλλεχθεί τα δεδομένα από όλες τις πηγές και έχει σχεδιαστεί η βάση δεδομένων. Το επόμενο βήμα είναι να γίνει η επεξεργασία και προετοιμασία των δεδομένων ώστε να μετατραπούν σε μορφή κατάλληλη για αποθήκευση στο σχήμα της βάσης δεδομένων. Σε γενικές γραμμές, η προεπεξεργασία των δεδομένων είναι απαραίτητη για τους εξής τρεις λόγους:

i. **Να γίνει επιλογή των δεδομένων που ενδιαφέρουν**. Για την εφαρμογή δεν χρειάζονται όλα τα δεδομένα που θα βρίσκονται σε κάθε αρχείο που έχει αποκτηθεί.
ii. **Τα αρχεία με τα αποτελέσματα των αλγορίθμων δεν έχουν ίδια μορφή**. Η διαμόρφωση των αρχείων με τα αποτελέσματα των διαφόρων αλγορίθμων είναι εντελώς διαφορετική από αλγόριθμο σε αλγόριθμο. Επομένως πρέπει να μετατραπούν τα δεδομένα από όλους τους αλγορίθμους σε μία κοινή μορφή και να ενοποιηθούν.
iii. **Τα δεδομένα από όλες τις πηγές πρέπει να μετατραπούν σε μορφή σύμφωνη με το σχήμα της βάσης δεδομένων**. Αυτό είναι αναγκαίο προκειμένου να είναι δυνατή η φόρτωση της βάσης δεδομένων με τα εν λόγω δεδομένα.

Για αυτό το κομμάτι της προεπεξεργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού AWK. Όπως εξηγήθηκε στην ενότητα 3.2, το γεγονός ότι όλα τα αρχεία που αποκτήθηκαν είναι αρχεία κειμένου αποτελεί μεγάλη διευκόλυνση. Αυτό που διέφερε από αρχείο σε αρχείο, ωστόσο, ήταν η δομή του. Όμως η AWK

είναι ιδανική στο να επεξεργάζεται αρχεία κειμένου και ειδικά όταν οι γραμμές των αρχείων είναι διαχωρισμένες σε πεδία, γι' αυτό και επιλέχθηκε η συγκεκριμένη γλώσσα για αυτό το στάδιο της εργασίας. Ό,τι περιγράφεται σε αυτή την ενότητα, λοιπόν, έχει πραγματοποιηθεί με δημιουργία προγραμμάτων AWK (AWK scripts) τα οποία εκτελέστηκαν με είσοδο τα αρχεία δεδομένων που αποκτήθηκαν.
Στη συνέχεια παρουσιάζονται τα βήματα προεπεξεργασίας των δεδομένων με τη σειρά που εκτελέστηκαν.

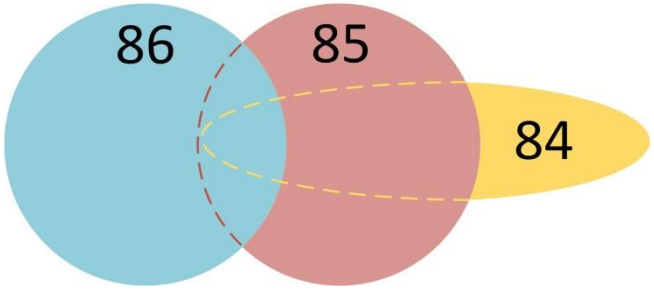
6.3.1 Επεξεργασία δεδομένων Ensembl και δημιουργία του πίνακα gene

Αρχικά έπρεπε να δημιουργηθούν οι πίνακες gene και mirna για δύο λόγους:

i) ο πίνακας interaction βασίζεται σε αυτούς (έχει ξένα κλειδιά),
ii) όλοι οι έλεγχοι ακεραιότητας δεδομένων που πραγματοποιήθηκαν πριν καν φορτωθεί η βάση, βασίζονται στους πίνακες gene και mirna. Ένας τέτοιος έλεγχος είναι, για παράδειγμα, ότι δεν υπάρχει αλληλεπίδραση που να αναφέρεται σε γονίδιο ή miRNA που δεν υπάρχει στη βάση.

6.3.1.1 Καταγραφή γονιδίων ανά έκδοση Ensembl

Για να δημιουργηθεί ο πίνακας gene, λοιπόν, αξιοποιήθηκαν τα δεδομένα της Ensembl από την έκδοση 86 και κατόπιν οπισθοχωρώντας μέχρι την 75. Για καλύτερη κατανόηση της λογικής με την οποία φορτώθηκαν τα δεδομένα της Ensembl στη βάση δεδομένων παρατίθεται και ένα σχήμα.



Εικόνα 6.2: ποια γονίδια λαμβάνονται από κάθε έκδοση

Αρχικά προστίθενται στον πίνακα gene όλα τα γονίδια της τελευταίας έκδοσης της Ensembl (έκδοση 86, μπλε σύνολο στην εικόνα) με το πεδίο ens_version = 86. Ακολούθως λαμβάνεται η αμέσως προηγούμενη έκδοση (85, κόκκινο σύνολο). Όσα γονίδια υπάρχουν και στις δύο εκδόσεις, δηλαδή η τομή κόκκινου και μπλε συνόλου, λαμβάνονται από την πιο πρόσφατη έκδοση, δηλαδή την 86. Αυτό γίνεται διότι για τα κοινά γονίδια των δύο εκδόσεων θέλουμε να ληφθούν τα πιο πρόσφατα δεδομένα τους.
Τώρα, όσα γονίδια της 85 δεν περιλαμβάνονται στην 86 λαμβάνονται όντως από την 85 με το πεδίο ens_version = 85. Αυτά, δηλαδή, είναι όσα γονίδια καταργήθηκαν μετά την έκδοση 85.
Στη συνέχεια συνεχίζεται η διαδικασία με την έκδοση 84 (κίτρινο σύνολο). Όσα γονίδια της έκδοσης 84 έχουν ήδη συμπεριληφθεί από τις δύο προηγούμενες εκδόσεις θα αγνοηθούν. Όσα γονίδια της έκδοσης 84 δεν υπάρχουν στις επόμενες εκδόσεις θα προστεθούν με το πεδίο ens_version = 84.
Αυτή η διαδικασία, λοιπόν, επαναλαμβάνεται με τον ίδιο τρόπο μέχρι την έκδοση 75. Ο λόγος που η διαδικασία επαναλαμβάνεται μέχρι την έκδοση 75 είναι επειδή αυτή είναι η παλαιότερη έκδοση που

χρησιμοποιείται από τους αλγορίθμους που υποστηρίζονται από την εφαρμογή. Συνεπώς δε θα είχε νόημα να συνεχιστεί η διαδικασία για εκδόσεις πριν από την 75.

Αυτή η διαδικασία προσδίδει δύο σημαντικά πλεονεκτήματα στη βάση δεδομένων και γι' αυτό επιλέχθηκε:

- για κάθε γονίδιο θα είναι γνωστό ποια ήταν η τελευταία έκδοση της Ensembl που το περιείχε,
- η βάση δεδομένων θα διαθέτει τις τρέχουσες πληροφορίες για τα γονίδια που υπάρχουν ακόμη ενώ, για τα καταργηθέντα, θα διαθέτει τις πιο ενημερωμένες πληροφορίες τη στιγμή κατάργησής τους.

6.3.1.2 Πεδία του πίνακα gene

Ολοκληρώνοντας αυτή τη διαδικασία, ο πίνακας gene είναι σχεδόν ολοκληρωμένος. Υπάρχουν όλες οι εγγραφές που θα τον αποτελούν και 6 από τα 9 πεδία κάθε εγγραφής. Τα τρία πεδία που λείπουν είναι τα πεδία καταμέτρησης αλληλεπιδράσεων (count_diana, count_targetscan, count_mirtarget). Αυτά θα προκύψουν λίγο αργότερα.

6.3.2 Επεξεργασία δεδομένων miRBase και δημιουργία του πίνακα mirna

6.3.2.1 Καταγραφή των miRNAs

Αυτό που ενδιαφέρει εδώ είναι η εξαγωγή της λίστας όλων των miRNAs κάθε οργανισμού. Η διαδικασία που ακολουθήθηκε ήταν η εξής:

- Εντοπίζεται μία νέα εγγραφή που αφορά ανθρώπινο πρόδρομο miRNA.
- Ακολούθως εντοπίζονται τα ώριμα miRNAs που προκύπτουν από αυτό. Για καθένα εξ αυτών καταγράφεται το όνομά του, το μοναδικό αναγνωριστικό του και οι συντεταγμένες αρχής και τέλους του.
- Εντοπίζεται η ακολουθία του πρόδρομου miRNA. Με βάση τις συντεταγμένες των ώριμων miRNAs επάνω σε αυτό, εξάγονται οι ακολουθίες των ώριμων miRNAs.
- Επαναλαμβάνεται αυτή η διαδικασία για κάθε ανθρώπινο πρόδρομο miRNA που θα εντοπιστεί στο αρχείο.

Με την ολοκλήρωση της διαδικασίας έχουν καταγραφεί όλα τα ανθρώπινα miRNAs. Η διαδικασία επαναλαμβάνεται και για τον ποντικό επακριβώς με τον ίδιο τρόπο.

6.3.2.2 Πεδία του πίνακα mirna

Έως εδώ έχουν παραχθεί τα 3 από τα 6 πεδία του πίνακα. Λείπουν, όπως και για τον πίνακα gene, τα πεδία καταμέτρησης των αλληλεπιδράσεων που θα παραχθούν στο τέλος. Κατά τ' άλλα, ο πίνακας mirna όπως είναι τώρα, περιέχει ήδη όλες τις εγγραφές που πρέπει.

6.3.3 Εξαγωγή αντιστοιχιών Ensembl – RefSeq

Αυτό το βήμα είναι απαραίτητο προτού προχωρήσει η επεξεργασία των αποτελεσμάτων των αλγορίθμων δεδομένου ότι ο MirTarget χρησιμοποιεί αναγνωριστικά κατά RefSeq ενώ στη βάση δεδομένων χρησιμοποιούνται αναγνωριστικά κατά Ensembl. Το πρόβλημα των αντιστοιχίσεων επεξηγήθηκε αναλυτικά στην παράγραφο 3.1.3.

Ο τρόπος που αντιμετωπίστηκε εδώ ήταν να δοθεί προτεραιότητα σε όσες αντιστοιχίες εντοπίζει η CCDS, έπειτα να δοθεί προτεραιότητα στις αντιστοιχίες που εντοπίζει η RefSeq και τελευταίες να ληφθούν οι αντιστοιχίσεις που εντοπίζει η Ensembl.

Για κάθε μετάγραφο της RefSeq που περιέχεται στη CCDS βρέθηκε το αντίστοιχο μετάγραφό στην Ensembl. Για εκείνα τα μετάγραφα της RefSeq που η CCDS εντοπίζει περισσότερα από ένα αντίστοιχα μετάγραφα της Ensembl (το πρόβλημα των «1 προς 1» αντιστοιχιών) παρατηρήθηκε ότι όλα τα μετάγραφα Ensembl που αναφέρονταν ανήκουν στο ίδιο γονίδιο. Άρα, παρ' ότι η αντιστοιχία ενός μεταγράφου της RefSeq δεν μπορεί να γίνει «1 προς 1» με μετάγραφα της Ensembl, το γονίδιο στο οποίο τελικά αποδίδεται το μετάγραφο είναι μοναδικό. Και δεδομένου ότι στην εργασία αυτή ενδιαφέρουν οι αλληλεπιδράσεις σε επίπεδο γονιδίου, το σημείο αυτό δεν ενοχλεί καθόλου.

Μάλιστα, με αφορμή αυτή τη διαπίστωση, ακολουθήθηκε αυτή η τακτική και για τις αντιστοιχίες που βρέθηκαν στη συνέχεια. Δηλαδή, δεν αναζητήθηκαν απλώς αντιστοιχίες από μετάγραφο της RefSeq προς μετάγραφο της Ensembl. Αντίθετα, αναζητήθηκε επιπλέον και σε ποιο γονίδιο της Ensembl ανήκει το κάθε μετάγραφο που θα βρεθεί. Έτσι οι αντιστοιχίες όχι μόνο μετατρέπουν τα αποτελέσματα σε μορφή Ensembl αλλά, επιπλέον, εκφράζουν κατευθείαν τις αλληλεπιδράσεις σε επίπεδο γονιδίου.

Στη συνέχεια για όσα μετάγραφα της RefSeq δεν περιλαμβάνονται στη CCDS, λήφθηκαν υπ' όψη οι αντιστοιχίες που ορίζει η RefSeq. Ο λόγος που δόθηκε αυτή η προτεραιότητα ήταν ότι αφού πρέπει να βρεθούν αντιστοιχίες **από** RefSeq **προς** Ensembl, θεωρήθηκε προτιμότερο να ληφθούν οι αντιστοιχίες που ορίζει η RefSeq.

Στη συνέχεια, για όσα μετάγραφα της RefSeq δεν παρέχει αντιστοιχία ούτε η CCDS ούτε η ίδια η RefSeq, έγινε προσπάθεια να αντληθούν αντιστοιχίες από την Ensembl.

Ωστόσο ακόμη και μετά από όλη αυτή τη διαδικασία υπήρξαν μετάγραφα που αναφέρονται στα αποτελέσματα του MirTarget τα οποία δεν κατέστη δυνατό να αντιστοιχηθούν σε κάποιο μετάγραφο (και κατ' επέκταση, γονίδιο) της Ensembl για δύο λόγους:

- Αρκετά από αυτά είναι καταργημένες εγγραφές της RefSeq.
- Υπήρχαν κάποια μετάγραφα της RefSeq τα οποία δεν κατέστη δυνατό να βρεθεί μοναδική αντιστοιχία ούτε σε επίπεδο γονιδίου. Αυτά αναγκαστικά αγνοήθηκαν.

Αφού υπήρξαν μετάγραφα που αναγκαστικά αγνοήθηκαν, οι αλληλεπιδράσεις στις οποίες συμμετέχουν αυτά τα μετάγραφα έπρεπε, αντίστοιχα, να αγνοηθούν. Ως προς τα αποτελέσματα του MirTarget, επομένως, προκύπτουν τα εξής:

	Άνθρωπος	Ποντικός
Πλήθος αναγνωριστικών RefSeq που χρησιμοποιούνται	35,869	27,517
Αντιστοιχίες που βρέθηκαν	34,532	25,065
Αντιστοιχίες που δεν βρέθηκαν	1,337	2,452
Συνολικές αλληλεπιδράσεις του MirTarget	1,873,265	925,645
Αλληλεπιδράσεις που αναγκαστικά αγνοήθηκαν	25,936 (1,4%)	26,677 (2,9%)

Πίνακας 6.6: στατιστικά στοιχεία για τις αλληλεπιδράσεις του MirTarget όπως προέκυψαν λόγω της μετατροπής των αναγνωριστικών από RefSeq προς Ensembl

6.3.4 Εξαγωγή αλληλεπιδράσεων του κάθε αλγορίθμου

Το επόμενο βήμα που έπρεπε να γίνει ήταν να μετατραπούν τα αποτελέσματα όλων των αλγορίθμων από το επίπεδο μεταγράφου στο επίπεδο γονιδίου. Από τα αρχεία του κάθε αλγορίθμου πρέπει να αντληθούν όλες οι αλληλεπιδράσεις και για κάθε αλληλεπίδραση να αντληθούν τρία πεδία: γονίδιο, miRNA και βαθμολογία.

6.3.4.1 DIANA–microT

Για τον DIANA–microT αυτό ήταν σχετικά εύκολο αφού η μορφή του αρχείου του παρέχει κατευθείαν αυτή την πληροφορία. Εκτός από το συγκεκριμένο μετάγραφο της αλληλεπίδρασης, στα αποτελέσματα αναφέρεται και το γονίδιο στο οποίο ανήκει το μετάγραφο. Επομένως, για κάθε αλληλεπίδραση, είναι έτοιμη η πληροφορία γονίδιο, miRNA, βαθμολογία.

6.3.4.2 TargetScan

Αυτός ο αλγόριθμος χρειάστηκε λίγο πιο μεθοδική προσέγγιση διότι εντοπίζει πολλαπλές αλληλεπιδράσεις μεταξύ ενός miRNA και ενός γονιδίου σε πολλαπλά μετάγραφα και πολλαπλές θέσεις πρόσδεσης. Όπως αιτιολογείται στην ενότητα 7.7, σε τέτοιες περιπτώσεις ως βαθμολογία της αλληλεπίδρασης εξελήφθη η μέγιστη βαθμολογία που συναντάται σε οποιαδήποτε θέση πρόσδεσης επάνω στα μετάγραφα του γονιδίου. Αυτή η λογική ακολουθήθηκε, φυσικά, για κάθε ξεχωριστό ζεύγος γονιδίου – miRNA. Έτσι προέκυψε και εδώ ένας προσωρινός πίνακας με τις στήλες γονίδιο, miRNA, βαθμολογία.

6.3.4.3 MirTarget

Η δυσκολία εδώ έγκειται στη μετατροπή των αναγνωριστικών από RefSeq σε Ensembl. Πέραν αυτού, η δομή του αρχείου περιέχει ήδη τις τρεις πληροφορίες που χρειάζονται.

Αρχικά, για τη μετατροπή των αναγνωριστικών, χρησιμοποιήθηκε η λίστα με τις αντιστοιχίες που παράχθηκε προηγουμένως. Επίσης, όπως εξηγήθηκε, οι αντιστοιχίες βρέθηκαν με τέτοιο τρόπο ώστε να γίνει η μετάβαση από μετάγραφα της RefSeq κατευθείαν σε γονίδια της Ensembl κι όχι απλώς σε μετάγραφα της Ensembl. Έτσι, αφότου εφαρμόστηκαν οι αντιστοιχίες, τα αποτελέσματα είχαν ήδη μετατραπεί στη μορφή γονίδιο, miRNA, βαθμολογία.

6.3.4.4 Ενοποίηση όλων των αποτελεσμάτων σε έναν πίνακα

Έχοντας επεξεργαστεί τα αποτελέσματα όλων των αλγορίθμων, επόμενο βήμα ήταν η ενοποίηση όλων των επιμέρους πινάκων σε έναν. Αν κάποια αλληλεπίδραση δεν προβλέπεται από κάποιον αλγόριθμο, τότε το αντίστοιχο πεδίο συμπληρώθηκε με "\N" το οποίο εκλαμβάνεται από την MySQL ως "null". Έτσι προέκυψε ο τελικός πίνακας interaction ο οποίος είναι, πλέον, καθ' όλα ολοκληρωμένος και έτοιμος να φορτωθεί στη βάση δεδομένων.

6.3.5 Έλεγχος ακεραιότητας δεδομένων

Σε αυτό το σημείο ελέγχθηκε αν όλα τα γονίδια και τα miRNAs που περιέχονται στον πίνακα interaction υπάρχουν ήδη στους πίνακες gene και mirna. Σκοπός αυτού του ελέγχου ήταν:

- να εντοπιστούν ενδεχόμενα σφάλματα που μπορεί να έγιναν κατά την προεπεξεργασία,
- ενδεχόμενες ασυνέπειες δεδομένων από τις πηγές,
- να γίνει έλεγχος όλων των πρωτευόντων κλειδιών,
- να γίνει έλεγχος όλων των ξένων κλειδιών.

Με αυτό τον έλεγχο θέλαμε να είμαστε βέβαιοι ότι η επικείμενη φόρτωση της βάσης δεδομένων θα ολοκληρωθεί χωρίς προβλήματα. Ο έλεγχος δεν επέστρεψε κάποιο σφάλμα ή ασυνέπεια οπότε μπορούσε να προχωρήσει η προεπεξεργασία στο τελευταίο βήμα.

6.3.6 Καταμέτρηση αλληλεπιδράσεων ανά αλγόριθμο

Καθώς ο πίνακας interaction έχει ολοκληρωθεί, έγινε μία απλή καταμέτρηση ανά γονίδιο και ανά miRNA, έτσι ώστε να καταμετρηθεί πόσες αλληλεπιδράσεις προβλέπει κάθε αλγόριθμος για κάθε αντικείμενο. Τα στοιχεία που προέκυψαν συμπληρώθηκαν στις στήλες count_diana, count_targetscan και count_mirtarget των πινάκων gene και mirna αντίστοιχα.

Με αυτό το τελευταίο βήμα ολοκληρώθηκε η προεπεξεργασία των δεδομένων και έχουν κατασκευαστεί πλήρως καθώς και ελεγχθεί και οι τρεις πίνακες της βάσης δεδομένων. Στη συνέχεια οι πίνακες φορτώθηκαν στη βάση δεδομένων και, συνεπώς, το πρώτο σκέλος της εργασίας που αφορά τα δεδομένα έχει, πλέον, ολοκληρωθεί.

6.4 Ανάπτυξη της εφαρμογής με Laravel

Σε αυτό το σημείο μπορεί, πλέον, να αρχίσει η υλοποίηση της εφαρμογής, κάτι που δεν ήταν εφικτό να γίνει χωρίς να υφίσταται το υποκείμενο στρώμα δεδομένων στο οποίο βασίζεται η εφαρμογή.

6.4.1 Γενική διάρθρωση αρχείων της εφαρμογής

Η ανάπτυξη της εφαρμογής διαρθρώνεται γύρω από αρχεία τεσσάρων ειδών τα οποία δημιουργήθηκαν:

- οι κλάσεις τύπου "controller" (ή ελεγκτές) που υλοποιούν τη λογική της εφαρμογής.
- οι κλάσεις τύπου "model" που υλοποιούν την επικοινωνία με τη βάση δεδομένων. Αξιοποιούνται από τους ελεγκτές όταν είναι αναγκαίο.
- οι κλάσεις τύπου "middleware" οι οποίες αναλαμβάνουν να διεκπεραιώσουν τον έλεγχο αν τα HTTP αιτήματα είναι έγκυρα ώστε να προχωρήσει η επεξεργασία τους από τους ελεγκτές.
- οι διάφορες προβολές (views) που αποτελούν τις ιστοσελίδες της εφαρμογής.

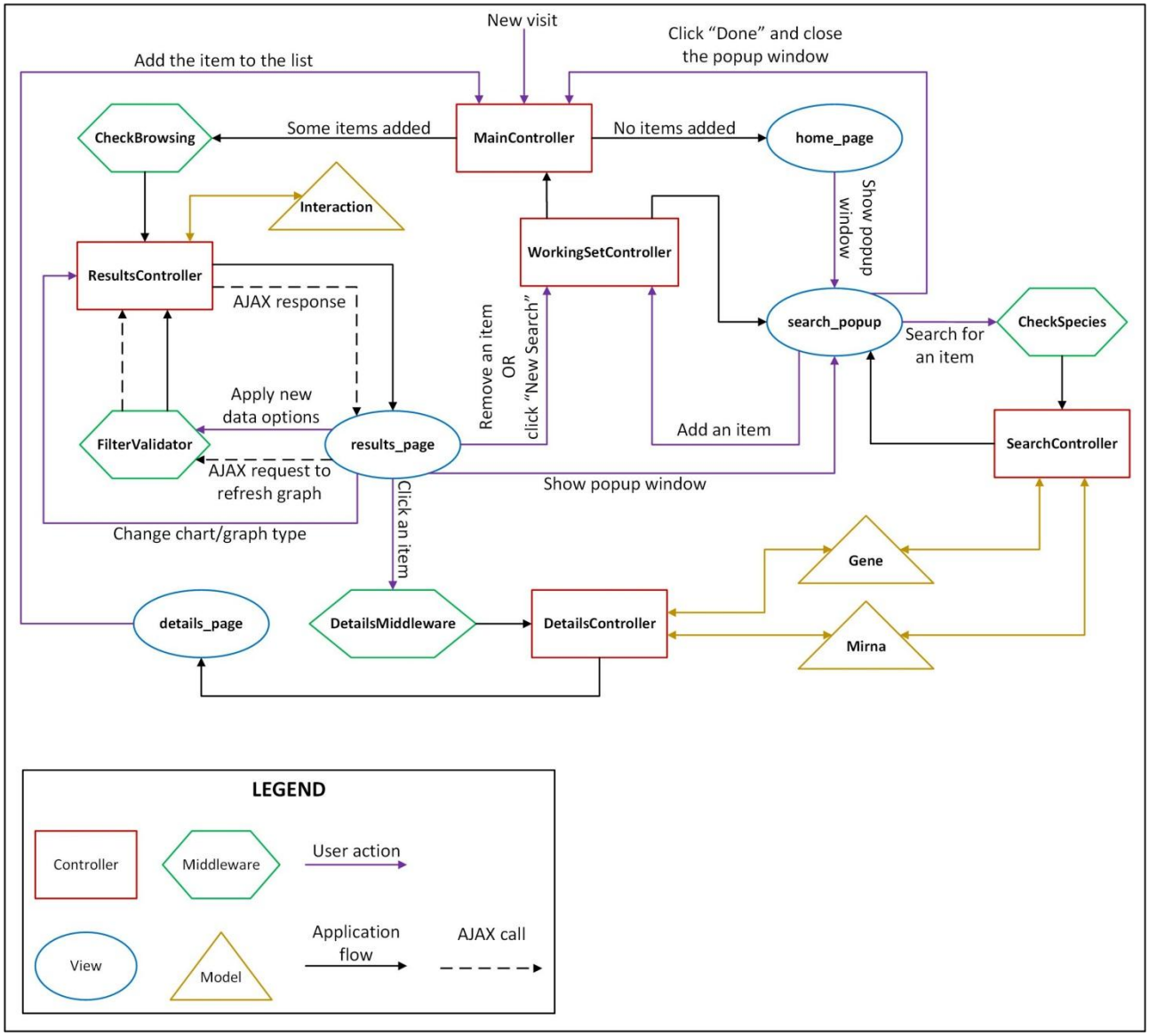
Όλες οι κλάσεις είναι αρχεία PHP ενώ οι προβολές είναι αρχεία HTML.

6.4.2 Βασική επεξήγηση λειτουργίας

Για να είναι κατανοητά όσα εξηγηθούν στη συνέχεια θα δοθεί μία συνοπτική επεξήγηση του πώς λειτουργεί η εφαρμογή. Επίσης, η εφαρμογή παρουσιάζεται σχεδιαγραμματικά. Επάνω στο σχήμα φαίνονται όλες οι κλάσεις που δημιουργήθηκαν, το είδος της κάθε κλάσης, η ροή της εφαρμογής ανάλογα με τις ενέργειες του χρήστη και οι αλληλεπιδράσεις των διαφόρων κλάσεων/προβολών μεταξύ τους.

Οι ενέργειες του χρήστη κατά σειρά είναι:

1) Αρχικά ο χρήστης φτάνει στην αρχική σελίδα η οποία θα είναι άδεια.
2) Χρησιμοποιώντας το αναδυόμενο παράθυρο μπορεί να αναζητήσει γονίδια και miRNAs και να τα προσθέσει στη «λίστα εργασίας» (working set).
3) Αφότου προσθέσει όσα γονίδια ή/και miRNAs θέλει στη λίστα εργασίας, κλείνει το αναδυόμενο παράθυρο και ανακατευθύνεται, πλέον, στη σελίδα των αποτελεσμάτων. Εκεί μπορεί να επιλέξει ποια προβολή επιθυμεί να βλέπει (προβολή πίνακα ή κάποιο από τα γραφήματα), να αλλάξει τις επιλογές στα φίλτρα ή να αλλάξει κάποια επιλογή στα γραφήματα, προκαλώντας, έτσι, μία κλήση AJAX.
4) Μπορεί και πάλι να χρησιμοποιήσει το αναδυόμενο παράθυρο από τη σελίδα αποτελεσμάτων για να προσθέσει κι άλλα αντικείμενα στη λίστα εργασίας. Ακόμη, μπορεί να αφαιρέσει αντικείμενα ή να αδειάσει εντελώς τη λίστα εργασίας για να ξεκινήσει μια καινούρια αναζήτηση.
5) Τέλος, υπάρχει διαθέσιμη και η σελίδα λεπτομερειών για οποιοδήποτε αντικείμενο η οποία είναι προσβάσιμη κάνοντας κλικ επάνω στο όνομα ενός οποιουδήποτε αντικειμένου στη σελίδα αποτελεσμάτων.



Εικόνα 6.3: σχεδιάγραμμα με τις κλάσεις της εφαρμογής και τη ροή μεταξύ τους

6.4.2.1 Υλοποίηση σύμφωνη με τη μεθοδολογία MVC

Ακολουθούν μερικές χρήσιμες παρατηρήσεις με τη βοήθεια του σχεδιαγράμματος που δείχνουν πώς η υλοποίηση της εφαρμογής ακολούθησε τη μεθοδολογία MVC. Οι παρακάτω παρατηρήσεις ανταποκρίνονται στα 5 βέλη της εικόνας 2.1 η οποία περιγράφει το μοντέλο MVC.

i) Οι ενέργειες του χρήστη ξεκινούν πάντα από προβολές. Αυτό ταιριάζει με την ιδέα ότι ο χρήστης αλληλεπιδρά με την εφαρμογή μέσω των προβολών.

ii) Οι ενέργειες του χρήστη καταλήγουν πάντα σε ελεγκτές ή middleware (ανάλογα αν η συγκεκριμένη ενέργεια ελέγχεται από κάποιο middleware ή όχι).

iii) Τα βέλη που ξεκινούν από ελεγκτές καταλήγουν πάντα σε προβολές. Δηλαδή οι ελεγκτές είναι εκείνοι που, σύμφωνα με το μοντέλο MVC, δημιουργούν και ενημερώνουν τις προβολές.

iv) Τα μοντέλα ανταλλάσσουν πληροφορίες μόνο με ελεγκτές.

v) Μετά τα middleware ακολουθεί πάντα ελεγκτής.

Δύο σημαντικές εξαιρέσεις στην παρατήρηση (iii) αποτελούν τα εξής:

- Το βέλος "Some items added" που ενώ ξεκινά από τον MainController καταλήγει σε middleware αντί για κάποια προβολή.
- Ένα βέλος που ξεκινά από τον WorkingSetController και καταλήγει στον MainController.

Σε αυτές τις δύο περιπτώσεις ο ελεγκτής δεν επιστρέφει κάποια προβολή αλλά συμβαίνει ανακατεύθυνση (redirect). Όμως, και σε αυτές τις περιπτώσεις, η ανακατεύθυνση είναι μία μορφή HTTP απόκρισης (HTTP response) που απλώς αντί να μεσολαβήσει ο χρήστης εξυπηρετείται κατευθείαν από την εφαρμογή. Επομένως μία ορθότερη και γενικότερη διατύπωση είναι ότι οι ελεγκτές επιστρέφουν, τελικά, μία HTTP απόκριση η οποία μπορεί να είναι είτε μία προβολή είτε μία ανακατεύθυνση. Άρα δεν υπάρχει αντίφαση.
Για την αποφυγή σύγχυσης, μία ακόμη εξαίρεση αποτελεί η ενέργεια "Show popup window" η οποία (και τις δύο φορές που μπορεί να εκτελεστεί) καταλήγει κατευθείαν από προβολή σε προβολή παραβιάζοντας, φαινομενικά, την παρατήρηση (ii). Αυτό συμβαίνει διότι η προβολή του αναδυόμενου παραθύρου απαιτεί μόλις μία εντολή. Οπότε, για αυτή την ενέργεια, θα ήταν πλεονασμός να δημιουργηθεί ελεγκτής αφού αυτή η μία εντολή μπορεί να εκτελεστεί κατευθείαν κατά τη δρομολόγηση (routing) αυτού του αιτήματος. Άρα ούτε εδώ υπάρχει αντίφαση.
Στη συνέχεια θα εξηγηθεί ποιες λειτουργίες εκτελεί το κάθε αρχείο της εφαρμογής.

6.4.3 Κλάσεις τύπου controller

6.4.3.1 MainController

Αυτός ο ελεγκτής επενεργεί όταν ζητείται η αρχική σελίδα της εφαρμογής. Τα καθήκοντά του είναι:

- Ανάλογα με το αν υπάρχουν αντικείμενα ή όχι στη λίστα εργασίας ανακατευθύνει το χρήστη προς τη σελίδα των αποτελεσμάτων "resultsPage" ή την αρχική σελίδα "homePage", αντίστοιχα.
- Μετά από προσθήκη ή αφαίρεση αντικειμένων από τη λίστα εργασίας, προκαλεί κατάλληλη ανακατεύθυνση ώστε να κληθεί ο ResultController για να ανανεωθούν αυτόματα τα αποτελέσματα.
- Μετά από την επιλογή "Default filters" αναλαμβάνει να θέσει τα προεπιλεγμένα φίλτρα και κατόπιν να ανανεώσει αυτόματα τα αποτελέσματα.

6.4.3.2 SearchController

Αυτός ο ελεγκτής επενεργεί όταν γίνεται αναζήτηση για κάποιον όρο στο αναδυόμενο παράθυρο. Τα καθήκοντά του είναι:

- Αναζητά στη βάση δεδομένων αντικείμενα που ανταποκρίνονται στην αναζήτηση του χρήστη. Για αυτό αξιοποιεί τα μοντέλα gene και mirna.
- Αν η αναζήτηση επιστρέφει πάρα πολλά ή κανένα αποτέλεσμα προσπαθεί να εκμαιεύσει προτεινόμενα αποτελέσματα για να διευκολύνει το χρήστη.
- Παράγει ενημερωτικά μηνύματα ανάλογα με την αναζήτηση έτσι ώστε σε ενδεχόμενη αποτυχημένη αναζήτηση ο χρήστης να διευκολυνθεί να ορίσει καλύτερα τον επόμενο όρο αναζήτησης.

- Καλεί τη σελίδα “searchPopup” και την ενημερώνει κατάλληλα ώστε να προβληθούν στο χρήστη τα αποτελέσματα της αναζήτησης.

6.4.3.3 WorkingSetController

Αυτός ο ελεγκτής επενεργεί όταν γίνεται κάποια ενέργεια σχετική με τη λίστα εργασίας. Τα καθήκοντά του είναι:

- Να προσθέτει αντικείμενα στη λίστα εργασίας. Σε αυτή την περίπτωση ενημερώνει κατάλληλα το αναδυόμενο παράθυρο.
- Να αφαιρεί αντικείμενα από τη λίστα εργασίας. Σε αυτή την περίπτωση ανακατευθύνει τη ροή της εφαρμογής στην αρχική σελίδα ώστε να προκληθεί η αυτόματη ανανέωση των αποτελεσμάτων.
- Να αρχικοποιεί τη λίστα εργασίας αν ζητηθεί καινούρια αναζήτηση. Εδώ ανακατευθύνει στην αρχική σελίδα επίσης.
- Να δημιουργεί κατάλληλα μηνύματα προς το χρήστη ανάλογα με την επιτυχή ή όχι έκβαση της κάθε ενέργειας (π.χ. “item added successfully”).

6.4.3.4 ResultsController

Αυτός ο ελεγκτής επενεργεί όταν υπάρχουν αντικείμενα στη λίστα εργασίας και ζητείται η σελίδα αποτελεσμάτων. Τα καθήκοντά του είναι:

- Ανάλογα με τα αντικείμενα που υπάρχουν στην λίστα εργασίας και τις τρέχουσες επιλογές φίλτρων, επικοινωνεί με τη βάση δεδομένων ώστε να ανακτήσει τα ανάλογα αποτελέσματα. Για αυτό αξιοποιεί το μοντέλο interaction.
- Φροντίζει για τη σελιδοποίηση των αποτελεσμάτων στην προβολή πίνακα.
- Εξυπηρετεί τα αιτήματα AJAX για να παρέχει ανανεωμένα δεδομένα στα γραφήματα. Αυτό συμβαίνει όταν ο χρήστης αλληλεπιδρά με τα γραφήματα και αλλάζει τις ρυθμίσεις τους.
- Όταν χρειάζεται, φροντίζει να ενημερώσει κατάλληλα τη σελίδα resultsPage.

6.4.3.5 DetailsController

Αυτός ο ελεγκτής επενεργεί όταν ζητείται η σελίδα λεπτομερειών ενός αντικειμένου. Τα καθήκοντά του είναι:

- Αναλαμβάνει να αντλήσει τις πληροφορίες του αντικειμένου από τη βάση δεδομένων. Για αυτό αξιοποιεί τα μοντέλα gene και mirna.
- Καλεί τη σελίδα “detailsPage” και της μεταβιβάζει τις ανακτηθείσες πληροφορίες.

6.4.4 Πλεονεκτήματα μεθόδου αναζήτησης γονιδίων & miRNAs

Ένα βασικό μειονέκτημα που παρατηρήθηκε σε ορισμένες σχετικές ιστοσελίδες είναι πως η αναζήτηση όρων δεν γίνεται με αρκετά εξυπηρετικό τρόπο για το χρήστη. Για παράδειγμα, σε πολλές ιστοσελίδες αν

επιθυμεί κάποιος να αναζητήσει πληροφορίες για ένα γονίδιο με βάση το μοναδικό του αναγνωριστικό, πρέπει να γράψει αναγκαστικά ολόκληρο το αναγνωριστικό, δηλαδή "ENSG000000000005". Ωστόσο στη δική μας εφαρμογή θα αρκούσε κάποιος να αναζητήσει απλώς για τον αριθμό «5» χωρίς καν να διευκρινίζει αν πρόκειται για γονίδιο ή miRNA. Μία αναζήτηση που επίσης θα επέστρεφε το επιθυμητό αποτέλεσμα είναι η αναζήτηση "ENSG5".

Αντίστοιχα αν αναζητήσει κανείς με τον αριθμό «4494» θα λάβει ως απάντηση τόσο το miRNA hsa-miR-4494 όσο και το hsa-miR-21-3p το οποίο έχει μοναδικό αναγνωριστικό το MIMAT0004494. Δηλαδή η αναζήτηση με αριθμούς επιστρέφει αποτελέσματα αξιοποιώντας τόσο το όνομα όσο και το μοναδικό αναγνωριστικό.

Ένα ακόμη σημείο είναι πως δεν απαιτείται να διαχωρίζει ο χρήστης τα ονόματα των miRNAs με παύλες. Π.χ. η αναζήτηση "mir21" θα επιστρέψει όλες τις απαντήσεις που ξεκινούν με "hsa-miR-21…". Επιπλέον, για την αναζήτηση ενός miRNA δεν απαιτείται να γράψει κανείς το τετριμμένο αρχικό κομμάτι του ονόματος που δηλώνει τον οργανισμό. Δηλαδή, δεν είναι αναγκαίο να αναζητήσει κάποιος το "hsa-mir-32", η αναζήτηση του "mir32" αρκεί.

Επίσης, αναφέρθηκε προηγουμένως πως σε περιπτώσεις πολλών ή καθόλου αποτελεσμάτων, ο ελεγκτής προσπαθεί να «εκμαιεύσει» προτεινόμενα γονίδια/miRNAs. Για παράδειγμα, η αναζήτηση "mir21" επιστρέφει πολλά αποτελέσματα από τη βάση δεδομένων όπως τα: hsa-miR-21-3p, hsa-miR-21-5p, hsa-miR-211-3p, hsa-miR-211-5p, hsa-miR-2110 και πολλά ακόμη. Όταν τα προτεινόμενα αποτελέσματα ξεπερνούν έναν αριθμό, ο SearchController προσπαθεί να δει αν κάποια από αυτά «μοιάζουν πολύ» με την αρχική αναζήτηση. Έτσι, τελικά, ο χρήστης σε αυτή την περίπτωση θα δει μόνο τα hsa-miR-21-3p και hsa-miR-21-5p.

Τέλος, ανάλογα με το αποτέλεσμα της κάθε αναζήτησης, εμφανίζονται στη σελίδα κατάλληλα ενημερωτικά μηνύματα για το χρήστη. Έτσι, εκτός από τα προτεινόμενα αποτελέσματα, βλέπει και μηνύματα που θα τον διευκολύνουν αν επιθυμεί να προσδιορίσει καλύτερα την επόμενη αναζήτησή του, αν δε βρήκε αυτό που ψάχνει. Αντίστοιχα, αν το αποτέλεσμα που προέκυψε είναι μοναδικό, επίσης ενημερώνεται κατάλληλα.

Με όλους αυτούς τους τρόπους, ο SearchController προσπαθεί να διευκολύνει το χρήστη ώστε να βρει το γονίδιο/miRNA που αναζητεί όσο πιο εύκολα γίνεται.

6.4.5 Κλάσεις τύπου middleware

6.4.5.1 Βασικοί έλεγχοι που εκτελούνται πάντα

Αρχικά, όλες οι ενδιάμεσες μονάδες (middleware) εκτελούν ορισμένους ελέγχους ανάλογα με το σκοπό της καθεμίας. Ωστόσο, ανεξαρτήτως των επιμέρους ελέγχων, οι ενδιάμεσες μονάδες επιτελούν πάντα δύο επιπλέον καθήκοντα:

- Πραγματοποιούν ελέγχους για να βρεθεί αν η εφαρμογή έχει περιέλθει σε ασυνεπή κατάσταση. Π.χ. αν με κάποιο τρόπο σταλεί ένα HTTP αίτημα που περιέχει επιλογή οργανισμού ενώ αυτό δεν είναι επιτρεπτό (δηλαδή έχει ήδη επιλεγεί οργανισμός) τότε η μονάδα checkSpecies θα το αποτρέψει.

- Αποτρέπουν τους χρήστες από το να προσπελάσουν κατευθείαν ιστοσελίδες της εφαρμογής πληκτρολογώντας διεύθυνση αντί να χρησιμοποιήσουν τη διεπαφή της εφαρμογής. Π.χ. αν επιχειρήσει ένας χρήστης να επισκεφθεί κατευθείαν τη σελίδα "…/results" η μονάδα checkBrowsing θα το αποτρέψει.

Η απόκριση της εφαρμογής στα παραπάνω, τις περισσότερες φορές είναι η απλή ανακατεύθυνση στην αρχική σελίδα με εμφάνιση κατάλληλου ενημερωτικού μηνύματος προς το χρήστη. Οι πληροφορίες της εφαρμογής (λίστα εργασίας και επιλογές φίλτρων) παραμένουν ανέπαφες. Ο χρήστης, αξιοποιώντας το ενημερωτικό μήνυμα, μπορεί να διορθώσει τις όποιες επιλογές του δεν επέτρεψαν την εκτέλεση μιας λειτουργίας και να συνεχίσει να χρησιμοποιεί την εφαρμογή κανονικά.
Στη σπάνια περίπτωση που η εφαρμογή περιέλθει σε ασυνεπή κατάσταση, θα παρουσιαστεί σελίδα σφάλματος ενώ η σελίδα της εφαρμογής θα εξαφανιστεί εντελώς. Σε αυτή την περίπτωση αρκεί να επισκεφθεί ο χρήστης την αρχική σελίδα της εφαρμογής, πληκτρολογώντας απλώς τη διεύθυνση της αρχικής σελίδας στο φυλλομετρητή. Κατόπιν θα είναι σε θέση να συνεχίσει τη χρήση της εφαρμογής.
Αναλυτικές λεπτομέρειες για την ανάνηψη από σφάλμα παρέχονται στην ενότητα 10.6.

6.4.5.2 CheckSpecies

Αυτή η ενδιάμεση μονάδα επενεργεί όταν γίνεται μία αναζήτηση όρου στο αναδυόμενο παράθυρο. Το καθήκον της είναι:

- Ελέγχει αν η επιλογή οργανισμού είναι έγκυρη. Π.χ. κακόβουλο αίτημα μέσω JavaScript που αποστέλλει μη έγκυρη επιλογή οργανισμού ή κάποιο σφάλμα της εφαρμογής.

6.4.5.3 CheckBrowsing

Αυτή η ενδιάμεση μονάδα επενεργεί όταν πρόκειται να προβληθεί η σελίδα αποτελεσμάτων. Καθήκον της είναι:

- Να μην επιτρέπει την κατευθείαν πρόσβαση στη σελίδα αποτελεσμάτων χωρίς να έχει προηγηθεί προσθήκη αντικειμένων στη λίστα εργασίας.
- Κάθε φορά που ζητούνται δεδομένα για κάποια σελίδα του πίνακα αποτελεσμάτων ή που ζητείται ανανέωση των γραφημάτων, επαληθεύει ότι δεν έχουν χαθεί για τον οποιονδήποτε λόγο τα δεδομένα της συνεδρίας (session), π.χ. σφάλμα στον εξυπηρετητή με τη διαχείριση της συνεδρίας. Αυτό εξασφαλίζει ότι η επακόλουθη επεξεργασία του αιτήματος από τον αντίστοιχο ελεγκτή δε θα δημιουργήσει σφάλμα .

6.4.5.4 FilterValidator

Αυτή η ενδιάμεση μονάδα επενεργεί όταν υποβάλλονται από το χρήστη καινούριες επιλογές στα φίλτρα. Όπως υπονοεί και το όνομά της, τα καθήκοντά της είναι:

- Επαληθεύει ότι όλες οι επιλογές φίλτρων είναι έγκυρες. Αν υπάρχει μη έγκυρη επιλογή φίλτρου (δηλαδή το HTTP αίτημα περιλαμβάνει δεδομένα που δεν εμπίπτουν στις αναμενόμενες τιμές) σταματά η εκτέλεση της εφαρμογής και ο χρήστης θα δει σελίδα σφάλματος.
- Επαληθεύει ότι οι επιλογές φίλτρων δεν είναι αντικρουόμενες μεταξύ τους. Π.χ. ταξινόμηση των αποτελεσμάτων με βάση αλγόριθμο που δεν είναι επιλεγμένος.
- Αν χρειάζεται, παράγει κατάλληλα ενημερωτικά μηνύματα προς το χρήστη ώστε να τον διευκολύνει να διορθώσει τις αντικρουόμενες επιλογές φίλτρων. Ακολούθως ενημερώνει κατάλληλα τη σελίδα των αποτελεσμάτων ώστε να εμφανιστούν τα ενημερωτικά μηνύματα.

6.4.5.5 DetailsMiddleware

Αυτή η ενδιάμεση μονάδα επενεργεί όταν ζητείται να προβληθεί η σελίδα λεπτομερειών ενός αντικειμένου. Τα καθήκοντά της είναι:

- Να επιβεβαιώσει ότι δεν επιχειρεί ο χρήστης να προσπελάσει τη σελίδα κατευθείαν και όχι κάνοντας κλικ επάνω στο όνομα ενός οργανισμού.
- Να επιβεβαιώσει ότι τα δεδομένα που συνοδεύουν το αίτημα ανταποκρίνονται όντως σε υπαρκτό αντικείμενο. Π.χ. ελέγχεται ότι ο κωδικός που διαχωρίζει τα γονίδια και τα miRNAs είναι έγκυρος, ότι ζητούνται λεπτομέρειες ενός μοναδικού αναγνωριστικού που είναι όντως ακέραιος αριθμός κτλ.

6.4.6 Προβολές (views)

Για τη σχεδίαση των σελίδων (προβολών) αξιοποιήθηκαν σε μεγάλο βαθμό οι δυνατότητες που παρέχει το Blade, όπως αναλύθηκαν στην ενότητα 4.5.1, και κυρίως η κληρονομικότητα και οι υποενότητες (sections). Η εφαρμογή ουσιαστικά διαθέτει τέσσερις διαφορετικές σελίδες:

- **home_page**: είναι η πρώτη, κενή σελίδα που βλέπει κάποιος όταν επισκέπτεται την εφαρμογή.
- **results_page**: είναι η σελίδα των αποτελεσμάτων. Η σελίδα αυτή, ανάλογα με το γράφημα που έχει επιλεγεί από το χρήστη, επιλέγει ποια από τις υποενότητες που αφορούν γραφήματα είναι η κατάλληλη και την ενσωματώνει.
- **search_popup**: το αναδυόμενο παράθυρο στο οποίο πραγματοποιούνται οι αναζητήσεις.
- **details_page**: η σελίδα λεπτομερειών.
- **main_template**: το βασικό πρότυπο (template) πάνω στο οποίο βασίζονται όλες οι υπόλοιπες σελίδες της εφαρμογής.

Εκτός αυτών των σελίδων έχουν οριστεί σε ξεχωριστά αρχεία και οι εξής υποενότητες (sections): table_view, d3_heat, d3_hive, d3_parallel, filters, messages, suggestions, working_set. Οι υποενότητες αυτές ενσωματώνονται από τις διάφορες σελίδες όταν αυτό είναι επιθυμητό ώστε να προκύψει η ολοκληρωμένη σελίδα και να παρουσιαστεί στο χρήστη.

Σημειώνουμε πως στη συνεχεία θα προτιμηθεί η λέξη «σελίδες» και όχι «προβολές». Μάλιστα, όπως έχει ακολουθηθεί ως τώρα, προτιμούμε να αποδίδουμε με τη λέξη «προβολή» τους διαφορετικούς τρόπους

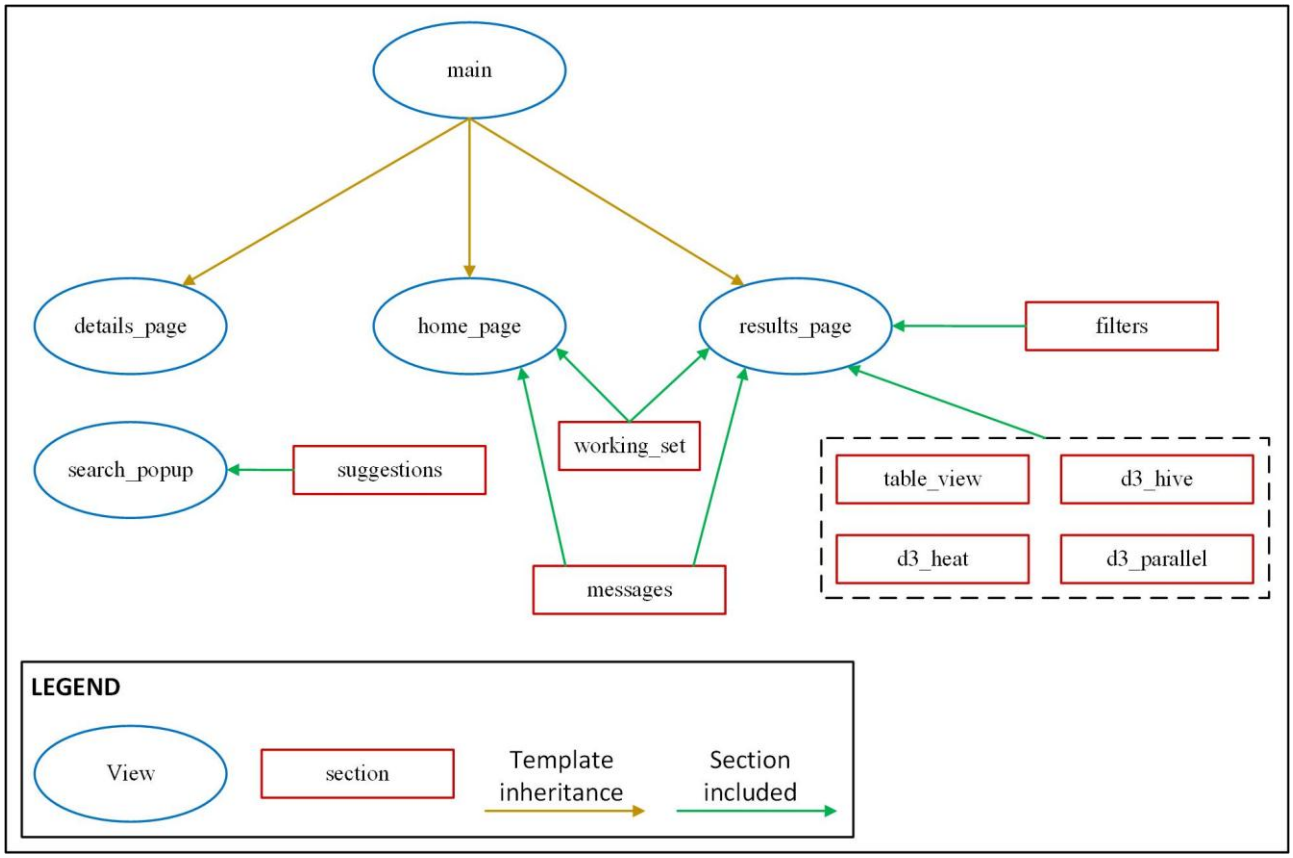
παρουσίασης των αλληλεπιδράσεων («προβολή αλληλεπιδράσεων») είτε σε προβολή πίνακα είτε με τα γραφήματα.

6.4.6.1 Χρήση υποενοτήτων

Για την καλύτερη σχεδίαση των ιστοσελίδων επιλέχθηκε τα διάφορα μέρη τους να αποτελούν ξεχωριστά HTML αρχεία. Αξιοποιήσαμε, έτσι, τη δυνατότητα που δίνει το Blade να χρησιμοποιούμε ανεξάρτητες και αυτούσιες υποενότητες. Λόγοι για αυτό είναι να υπάρχει η δυνατότητα επαναχρησιμοποίησης αυτούσιων τμημάτων σε διαφορετικές σελίδες, τα αρχεία HTML να μην προκύπτουν πολύ μεγάλα σε μέγεθος ώστε να έχουμε γενικώς μικρότερα και πιο ευανάγνωστα αρχεία και, τέλος, αφού τα διάφορα τμήματα είναι ανεξάρτητα μεταξύ τους λειτουργικά, θεωρήθηκε καλύτερο να είναι ανεξάρτητα και δομικά.

6.4.6.2 Σχεδιάγραμμα σελίδων

Στο παρακάτω διάγραμμα φαίνονται όλα τα αρχεία HTML που δημιουργήθηκαν και οι μεταξύ τους σχέσεις. Διακρίνεται, επίσης, ποια αρχεία κληρονομούνται από ποια και ποια αρχεία ενσωματώνονται από άλλα.



Εικόνα 6.4: κληρονομικότητα και υποενότητες στις σελίδες της εφαρμογής (υλοποίηση με Blade)

6.4.6.3 main

Ο σκοπός αυτής της σελίδας είναι να ορίζει το βασικό σκελετό για όλες τις σελίδες της εφαρμογής. Επί της ουσίας, δηλαδή, αποτελεί ένα πρότυπο, όπως μαρτυρά και το όνομά της. Όλες οι σελίδες κληρονομούν αυτό το πρότυπο και, έτσι, οτιδήποτε υπάρχει μέσα σε αυτό γίνεται αυτομάτως διαθέσιμο σε όλες τις σελίδες. Το βασικό στοιχείο του mainTemplate, λοιπόν, είναι ότι ορίζει την ετικέτα <head> η οποία περιλαμβάνει τις δηλώσεις για τα αρχεία JavaScript και CSS που θέλαμε να είναι διαθέσιμα σε όλες τις σελίδες της εφαρμογής. Έτσι, κάθε εξωτερικό αρχείο JavaScript ή CSS δηλώνεται μόνο μία φορά αντί να πρέπει να δηλωθεί ξεχωριστά σε κάθε σελίδα της εφαρμογής.
Αυτό το αρχείο, παρ' ότι είναι τύπου HTML, δε χρησιμοποιείται πουθενά αυτούσιο ως σελίδα της εφαρμογής.

6.4.6.4 home_page

Αποτελεί την αρχική σελίδα της εφαρμογής η οποία είναι σχεδόν κενή. Περιέχει μόνο μία άδεια λίστα εργασίας και το αναδυόμενο παράθυρο. Επίσης, ανάλογα με τις προηγούμενες ενέργειες, ενδεχομένως περιέχει και ορισμένα μηνύματα αν, για παράδειγμα, βρεθεί εδώ ο χρήστης έπειτα από ανακατεύθυνση.

6.4.6.5 results_page

Αποτελεί την κυρίως σελίδα της εφαρμογής ενώ είναι η μοναδική σελίδα που περιέχει σχεδόν όλες τις άλλες επιμέρους υποενότητες. Στο χώρο που παρουσιάζονται τα αποτελέσματα, ανάλογα με το τι έχει επιλέξει ο χρήστης, ενσωματώνεται κάθε φορά η κατάλληλη υποενότητα (είτε ο πίνακας είτε κάποιο απ' τα γραφήματα). Η υποενότητα που αφορά τις αλληλεπιδράσεις ανανεώνεται με AJAX.

6.4.6.6 search_popup

Είναι το αναδυόμενο παράθυρο στο οποίο πραγματοποιούνται οι αναζητήσεις γονιδίων/miRNAs. Η σελίδα αυτή χρησιμοποιείται από τις σελίδες home_page και results_page χωρίς, αρχικά, να είναι ορατή. Κατόπιν εμφανίζεται μόνο όταν ζητηθεί από το χρήστη και ο τρόπος εμφάνισής της ορίζεται με ιδιότητες απόλυτης τοποθέτησης (absolute positioning) στο CSS. Έτσι εμφανίζεται ως αναδυόμενο παράθυρο στο χρήστη.
Με τη σειρά της, ενσωματώνει την υποενότητα "suggestions" η οποία περιέχει τα προτεινόμενα αποτελέσματα αναζήτησης. Τα περιεχόμενα ολόκληρου του αναδυόμενου παραθύρου ανανεώνονται με AJAX όταν γίνεται μία αναζήτηση ή μια ενέργεια προσθήκης αντικειμένου.

6.4.6.7 details_page

Αποτελεί τη μοναδική στατική σελίδα της εφαρμογής κάτι που φαίνεται και από το γεγονός πως δεν ενσωματώνει καμία άλλη υποενότητα. Παρουσιάζει απλώς τις λεπτομέρειες ενός αντικειμένου.

6.4.7 Κλάσεις τύπου model (μοντέλα)

Τα μοντέλα είναι κλάσεις που αναλαμβάνουν την επικοινωνία με τη βάση δεδομένων. Καλούνται από τους ελεγκτές όταν απαιτείται να ανακτηθούν δεδομένα από τη βάση. Σε αυτή την εφαρμογή δεν απαιτείται τα μοντέλα να προβαίνουν σε συναλλαγές εγγραφής ή ενημέρωσης πληροφοριών στη βάση δεδομένων.
Κατασκευάστηκαν τρία μοντέλα, ένα για κάθε πίνακα της βάσης δεδομένων τα Gene, Mirna και Interaction. Καθένα εξ αυτών αναλαμβάνει την επικοινωνία με τη βάση δεδομένων όταν απαιτούνται πληροφορίες από τον ομώνυμο πίνακα.
Η αρμοδιότητα του κάθε μοντέλου είναι να κατασκευάσει ένα κατάλληλο ερώτημα προς τη βάση δεδομένων αναλόγως με τις πληροφορίες που αναμένει η εφαρμογή. Όταν καλείται ένα μοντέλο από έναν ελεγκτή τότε αναμένει από αυτόν να του παράσχει κάποια ορίσματα. Ανάλογα με τα ορίσματα που λαμβάνουν τα μοντέλα, «αντιλαμβάνονται» τι είδους ερώτημα πρέπει να εκτελεστεί προς τη βάση δεδομένων και κατασκευάζουν αναλόγως το SQL ερώτημα που θα μεταβιβαστεί στη βάση προς εκτέλεση.
Στη συνέχεια, μόλις είναι διαθέσιμη η απάντηση από τη βάση δεδομένων, τα μοντέλα επιστρέφουν τα αποτελέσματα πίσω στους ελεγκτές σε κατάλληλη μορφή (κλάση Collection του Laravel). Στη συνέχεια οι ελεγκτές μπορούν να εργαστούν με τα δεδομένα αυτά ώστε να κατασκευάσουν τις ιστοσελίδες, να κάνουν υπολογισμούς κτλ.
Ένα πολύ σημαντικό μέτρο ασφαλείας που υλοποιείται από τα μοντέλα είναι η προστασία της βάσης δεδομένων από κακόβουλη έγχυση κώδικα SQL (SQL injection). Αυτό μπορεί να συμβεί όπου ο χρήστης έχει τη δυνατότητα να εισάγει δεδομένα, όπως για παράδειγμα στο αναδυόμενο παράθυρο κατά την αναζήτηση γονιδίων/miRNAs. Τα μοντέλα, κατά την κατασκευή των ερωτημάτων SQL λαμβάνουν μέτρα προστασίας από αυτή την ευπάθεια (vulnerability).

6.4.8 Αρχείο διαμόρφωσης της εφαρμογής (configuration file)

Κατά την ανάπτυξη του κώδικα PHP βασικός σκοπός μας ήταν η εφαρμογή να είναι εύκολα επεκτάσιμη. Για παράδειγμα, θέλαμε να είναι πολύ εύκολο να προστεθεί στην εφαρμογή ένας νέος αλγόριθμος πρόβλεψης ή ένας νέος οργανισμός. Επίσης, θέλαμε να είναι πολύ εύκολο να προστεθούν επιλογές στα φίλτρα της σελίδας αποτελεσμάτων.
Για αυτόν ακριβώς το λόγο δημιουργήθηκε ένα αρχείο διαμόρφωσης στο οποίο αποθηκεύονται κάποιες βασικές πληροφορίες γύρω από την εφαρμογή. Έτσι, όλες οι επιλογές στη διεπαφή της εφαρμογής, ο πίνακας αλληλεπιδράσεων, τα γραφήματα κτλ., όλα κατασκευάζονται δυναμικά αντλώντας πληροφορίες από το αρχείο διαμόρφωσης. Για παράδειγμα, η εφαρμογή μπορεί πάρα πολύ εύκολα να υποστηρίξει έναν επιπλέον οργανισμό, προσθέτοντας απλά τα στοιχεία του οργανισμού αυτού στο αρχείο διαμόρφωσης (υποθέτοντας, βέβαια, πως υπάρχουν οι πληροφορίες του οργανισμού αυτού στη βάση δεδομένων)!
Το αρχείο διαμόρφωσης περιέχει τις εξής βασικές πληροφορίες:

- Λίστα των αντικειμένων που χειρίζεται η εφαρμογή (gene, mirna) και ιδιότητες αυτών. Όπου παράγονται ενημερωτικά μηνύματα προς το χρήστη, οι κατάλληλες λέξεις αντλούνται από αυτό τον

πίνακα. Επίσης, το όνομα των μοναδικών αναγνωριστικών καθώς και το μήκος του αριθμητικού μέρους τους.

```
define('GENE', 21);
define('MIRNA', 22);
'types' => [
      GENE => [
            'word' => 'gene',
            'start_word' => 'Gene',
            'plural' => 'genes',
            'start_plural' => 'Genes',
            'id' => 'Ensembl Gene ID',
            'digits' => '12'
      ],
      MIRNA => [
            'word' => 'miRNA',
            'start_word' => 'miRNA',
            'plural' => 'miRNAs',
            'start_plural' => 'miRNAs',
            'id' => 'miRBase accession number',
            'digits' => '7'
      ]
];
```

- Λίστα των οργανισμών που υποστηρίζονται και ποια βάση δεδομένων περιέχει τα δεδομένα κάθε οργανισμού. Έτσι η εφαρμογή κάνει μία αναφορά μόνο σε αυτό το σημείο του αρχείου διαμόρφωσης, ορίζει με ποια βάση δεδομένων θα γίνει όλη η επακόλουθη επικοινωνία και, ακολούθως, η εφαρμογή χειρίζεται και τους δύο οργανισμούς με επακριβώς τον ίδιο τρόπο, «αδιαφορώντας» στη συνέχεια ποιος είναι ο τρέχων οργανισμός (το σημείο αυτό συζητείται εκτενώς στην ενότητα 7.2).

```
define('HUMAN', 11);
define('MOUSE', 12);
'dbConnections' => [
      HUMAN => 'hsadb',
      MOUSE => 'mmudb'
];
```

- Λίστα των αλγορίθμων που υποστηρίζονται. Εδώ καταγράφονται πληροφορίες όπως αν ένας αλγόριθμος χρησιμοποιεί θετικές ή αρνητικές βαθμολογίες, ποια είναι η ελάχιστη και η μέγιστη πιθανή βαθμολογία, το όριο για τις προεπιλογές των φίλτρων, η ακρίβεια των βημάτων για τους συρόμενους επιλογείς στα φίλτρα κτλ.

```
define('DIANA', 1);
define('TARGETSCAN', 2);
define('MIRTARGET', 3);
```

```
define('COMBO', 99);
'tpas' => [
	DIANA => [
		'name' => 'DIANA-microT',
		'positive' => true,
		'min' => 0,
		'max' => 1,
		'threshold' => 0.7,
		'step' => 0.01
	],
	TARGETSCAN => [
		'name' => 'TargetScan',
		'positive' => false,
		'min' => -2.950,
		'max' => 0,
		'threshold' => -1.1,
		'step' => 0.001
	],
	MIRTARGET => [
		'name' => 'MirTarget',
		'positive' => true,
		'min' => 50,
		'max' => 100,
		'threshold' => 70,
		'step' => 1
	],
	COMBO => [
		'name' => 'Combo score',
		'positive' => true,
		'min' => 0,
		'max' => 3,
		'threshold' => 0,
		'step' => 0.01
	]
];
```

Με αυτό τον τρόπο επετεύχθη οι πληροφορίες που πρέπει να είναι στατικά ορισμένες, να βρίσκονται όλες σε ένα και μόνο αρχείο της εφαρμογής. Σε κανένα άλλο αρχείο δεν υπάρχουν καταγεγραμμένες (hard coded) τέτοιου είδους πληροφορίες – τα πάντα λειτουργούν και κατασκευάζονται δυναμικά με βάση το αρχείο διαμόρφωσης και τα αποθηκευμένα δεδομένα της βάσης δεδομένων.

6.5 Υλοποίηση γραφημάτων

Μέχρι αυτό το σημείο έχει ολοκληρωθεί η υλοποίηση των δύο εκ των τριών μερών της εφαρμογής: η βάση δεδομένων είναι έτοιμη ενώ και η το προγραμματιστικό κομμάτι της PHP έχει ολοκληρωθεί. Το τελευταίο μέρος που απομένει να υλοποιηθεί είναι η κατασκευή των γραφημάτων.

6.5.1 Πώς κατασκευάζεται ένα γράφημα

Τα γραφήματα υλοποιούνται με JavaScript με τη βοήθεια της βιβλιοθήκης D3. Ο κώδικας JavaScript που δημιουργεί το γράφημα αποτελεί μέρος της HTML σελίδας που αποστέλλεται από τον εξυπηρετητή προς το φυλλομετρητή του χρήστη. Η κατασκευή των γραφημάτων, όμως, γίνεται εξ ολοκλήρου στη μεριά του χρήστη (client–side) από τη JavaScript. Το μόνο που προέρχεται από τον εξυπηρετητή είναι τα δεδομένα βάσει των οποίων θα κατασκευαστεί το γράφημα και, φυσικά, ο προς εκτέλεση κώδικας JavaScript.
Όλα τα γραφήματα ακολουθούν τα ίδια βασικά βήματα για την κατασκευή τους, τα οποία είναι τα εξής:

1) **Φόρτωση της σελίδας του γραφήματος από τον εξυπηρετητή**. Εδώ εννοείται το φόρτωμα του κώδικα HTML και των αντίστοιχων αρχείων (scripts) της JavaScript που απαιτούνται. Η κατασκευή των γραφημάτων ξεκινά αφότου η σελίδα έχει φορτωθεί πλήρως.
2) **Επικοινωνία με τον εξυπηρετητή για ανάκτηση των αλληλεπιδράσεων.** Όταν αρχίσει να εκτελείται ο κώδικας που κατασκευάζει το γράφημα, το πρώτο που γίνεται είναι να ζητηθούν από τον εξυπηρετητή τα δεδομένα που θα απεικονιστούν. Με άλλα λόγια, τα προς απεικόνιση δεδομένα δεν είναι μέρος του HTML κώδικα. Τα δεδομένα παρέχονται από τον εξυπηρετητή ως ένα TSV αρχείο το οποίο φορτώνεται στο παρασκήνιο από τη JavaScript. Το αρχείο αυτό είναι σχηματοποιημένο σε πίνακα όπου κάθε γραμμή αρχείου αντιπροσωπεύει μία αλληλεπίδραση. Ο πίνακας παρέχει τις εξής στήλες: όνομα και αναγνωριστικό γονιδίου, όνομα και αναγνωριστικό miRNA και από μία στήλη για τον κάθε αλγόριθμο. Οι στήλες των βαθμολογιών περιέχουν είτε μια βαθμολογία, αν η αλληλεπίδραση προβλέπεται από τον εκάστοτε αλγόριθμο, ή παύλα αν δεν προβλέπεται.
3) **Προεπεξεργασία δεδομένων: βήμα 1.** Η D3 επεξεργάζεται αυτό το αρχείο (parse) και δημιουργεί αντικείμενα (objects) της JavaScript καθένα από τα οποία αντιπροσωπεύει μία αλληλεπίδραση. Τα αντικείμενα αυτά θα χρησιμοποιηθούν αργότερα ως στοιχεία που θα προβληθούν επάνω στο γράφημα.
4) **Προεπεξεργασία δεδομένων: βήμα 2.** Σε κάποια γραφήματα πρέπει να τηρηθεί επιπλέον λίστα με τα γονίδια και τα miRNAs. Αν απαιτείται, τότε αυτές οι λίστες δημιουργούνται σε αυτό το σημείο. Αυτό απαιτείται από τα γραφήματα hive plot και heat map καθώς αυτά τα δύο περιέχουν άξονες που αντιπροσωπεύουν αντίστοιχα γονίδια και miRNAs.
5) **Κατασκευή του σκελετού του γραφήματος.** Για τη δημιουργία του βασικού σκελετού λαμβάνονται υπ' όψη τα αντικείμενα και οι λίστες που δημιουργήθηκαν προηγουμένως. Για παράδειγμα, σε αυτό το στάδιο δημιουργούνται οι άξονες του parallel coordinates, ένας άξονας για

κάθε αλγόριθμο. Αντίστοιχα, σε αυτό το στάδιο δημιουργούνται οι άξονες του hive plot καθώς και οι άξονες και το πλέγμα του heat map. Οι άξονες λαμβάνουν τιμές και κλίμακες έτσι ώστε να καθορίζονται οι θέσεις των αντικειμένων που θα τοποθετηθούν επάνω στο γράφημα. Οι άξονες τοποθετούνται επάνω στο γράφημα.

6) **Αντιστοίχιση και «δέσιμο» (bind) των αντικειμένων που δημιουργήθηκαν κατά την προεπεξεργασία με στοιχεία του γραφήματος**. Στο parallel coordinates σε αυτό το σημείο δημιουργείται μία γραμμή για κάθε αλληλεπίδραση και η γραμμή αυτή «δένεται» (γίνεται bind όπως είναι η ορολογία της D3) με το αντίστοιχο αντικείμενο JavaScript. Θυμίζουμε πως τα αντικείμενου δημιουργήθηκαν στο βήμα (3), ένα αντικείμενο για κάθε αλληλεπίδραση. Αντίστοιχα, στο hive plot δημιουργείται μία ακμή για κάθε αλληλεπίδραση και στο heat map δημιουργείται ένα κελί για κάθε αλληλεπίδραση.
7) **Τοποθέτηση των αντικειμένων επάνω στο σκελετό του γραφήματος.** Κάθε γραμμή ή κελί που δημιουργήθηκε θα πρέπει να λάβει τη σωστή θέση επάνω στο γράφημα. Η θέση του κάθε αντικειμένου θα οριστεί με βάση τις τιμές των αξόνων. Σε αυτό το βήμα γίνεται κατανοητό, επιπλέον, γιατί η σχεδίαση των αξόνων έπρεπε να προηγηθεί.
8) **Προσθήκη των διάφορων χαρακτηριστικών αλληλεπίδρασης με το χρήστη**. Δηλαδή, σε αυτό το βήμα μετατρέπεται το γράφημα από στατικό σε δυναμικό. Δημιουργούνται οι κατάλληλες επιλογές (μενού για χρωματισμό ανά αλγόριθμο, επιλογές για τονισμό γραμμών σε mouseover κτλ.) έτσι ώστε ο χρήστης να είναι σε θέση να προσαρμόσει την εμφάνιση του γραφήματος**.** Σε αυτό το βήμα δημιουργούνται οι κατάλληλες συναρτήσεις JavaScript οι οποίες προσαρμόζουν την εμφάνιση του γραφήματος ανάλογα με αυτές τις επιλογές.
9) **Δημιουργία των κατάλληλων κλήσεων AJAX για την ασύγχρονη ανανέωση των δεδομένων.** Εδώ δημιουργούνται κλήσεις AJAX οι οποίες αξιοποιούνται όταν ο χρήστης ανανεώνει τα δεδομένα του γραφήματος. Αυτό συμβαίνει σε περίπτωση αλλαγής των φίλτρων ή αλλαγής του πλήθους των προβαλλόμενων αλληλεπιδράσεων. Για να γίνει η ανανέωση του γραφήματος με βάση τα νέα δεδομένα πρέπει να επανεκτελεστούν τα βήματα 2, 3, 4, 6 και 7. Ως προς το βήμα 5, ενδεχομένως να απαιτείται εκ νέου βαθμονόμηση κάποιου άξονα ή αλλαγή της διάταξής του – επομένως εκτελείται και ένα μέρος του βήματος 5.

Η διαδικασία αυτή ακολουθήθηκε για όλα τα γραφήματα. Τα βήματα 2 έως 7 υλοποιούνται με χρήση της D3, για το βήμα 8 χρησιμοποιούνται από κοινού η D3 με την jQuery ενώ για το βήμα 9 χρησιμοποιείται αποκλειστικά η jQuery και μερικές συναρτήσεις απλής (pure) JavaScript.

Σημείωση: για τα βήματα 2 έως 7 μπορεί να γίνει αναφορά και στην ενότητα 4.6 στην οποία παρουσιάζεται ο τρόπος με τον οποίο λειτουργεί η D3. Επιπλέον, οι επιμέρους δυνατότητες αλληλεπίδρασης με το χρήστη καθώς και οι δυνατότητες προσαρμογής της εμφάνισης κάθε γραφήματος παρουσιάζονται αναλυτικότατα στην ενότητα 10.4. Τέλος, για το βήμα 9, οι κλήσεις AJAX που ανανεώνουν τα γραφήματα παρουσιάζονται στο σχεδιάγραμμα ροής της εφαρμογής στην ενότητα 6.4.2. Επομένως με μια αναφορά στο διάγραμμα αυτό γίνεται σαφές με ποιο τρόπο λειτουργεί η ανανέωση των γραφημάτων, ποιες κλάσεις καλούνται κτλ.

6.5.2 Τα αντικείμενα των γραφημάτων

Έχοντας εξηγήσει πώς κατασκευάζονται τα γραφήματα, είναι χρήσιμο να εξηγηθεί και πώς απεικονίζονται τα δεδομένα επάνω στα γραφήματα ως αντικείμενα. Στις επόμενες παραγράφους, λοιπόν, θα παρουσιαστεί συνοπτικά από ποια στοιχεία αποτελούνται τα γραφήματα (σκελετός και αντικείμενα).

6.5.2.1 Parallel coordinates

Σκελετός: 2 έως 4 κάθετοι άξονες συνολικά. Ένας κάθετος άξονας πάντα ορατός με τα ονόματα των αλληλεπιδράσεων και ένας κάθετος άξονας/αλγόριθμο πρόβλεψης. Κάθετοι άξονες εμφανίζονται μόνο για τους επιλεγμένους αλγορίθμους.
Αντικείμενα: γραμμές όπου κάθε γραμμή αντιπροσωπεύει και μία αλληλεπίδραση. Οι αλληλεπιδράσεις που περιέχονται στο γράφημα αντιστοιχούν στο επιλεγμένο πλήθος.

6.5.2.2 Hive Plot

Σκελετός: 2 κάθετοι άξονες. Ο ένας αντιπροσωπεύει τα γονίδια και ο άλλος τα miRNAs.
Αντικείμενα:

- Κόμβοι. Οι κόμβοι βρίσκονται επάνω στους άξονες. Κάθε κόμβος αντιστοιχεί σε ένα γονίδιο ή miRNA.
- Ακμές. Κάθε ακμή συνδέει ένα γονίδιο με ένα miRNA και αντιστοιχεί σε μία αλληλεπίδραση.

Τα γονίδια και τα miRNAs που φαίνονται ως κόμβοι είναι εκείνα τα γονίδια και miRNAs που συμμετέχουν στο προβαλλόμενο πλήθος αλληλεπιδράσεων. Αν υπάρχουν γονίδια/miRNAs που υπάρχουν στη λίστα εργασίας αλλά «λείπουν» από το γράφημα, αυτό συμβαίνει απλώς επειδή τα εν λόγω γονίδια/miRNAs δε συμμετέχουν στις προβαλλόμενες αλληλεπιδράσεις.

6.5.2.3 Heat Map

Σκελετός: 2 άξονες κάθετοι μεταξύ τους. Ο ένας αντιπροσωπεύει τα γονίδια και ο άλλος τα miRNAs. Οι δύο άξονες ορίζουν έναν ορθογώνιο παραλληλόγραμμο χώρο ο οποίος διαχωρίζεται σε στήλες και γραμμές. Έτσι διαμορφώνονται κελιά.
Αντικείμενα: τα κελιά. Κάθε κελί αντιστοιχεί σε μία αλληλεπίδραση που ενδεχομένως βρίσκεται μεταξύ των x κορυφαίων αλληλεπιδράσεων που προβάλλονται. Η αλληλεπίδραση, αν υπάρχει, θα είναι μεταξύ του γονιδίου και του miRNA που ορίζουν το εν λόγω κελί.

Για τη διαμόρφωση των αξόνων λαμβάνονται υπ' όψη όλα τα γονίδια και τα miRNAs που συναντώνται στις προβαλλόμενες αλληλεπιδράσεις.

7

Σχεδιαστικά ζητήματα της βάσης δεδομένων

Το τελικώς επιλεχθέν σχήμα της βάσης δεδομένων, όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο, είναι αρκετά απλό. Ο καθορισμός αυτού του σχήματος, ωστόσο, απεδείχθη αρκετά απαιτητική διαδικασία. Λόγω της φύσης των δεδομένων, κατά τη σχεδίαση της βάσης δεδομένων ανέκυψαν διάφορα σχεδιαστικά ζητήματα. Για αυτά τα ζητήματα εντοπίσθηκαν οι παράγοντες που τα επηρεάζουν, αναζητήθηκαν οι βέλτιστες σχεδιαστικές επιλογές και περιγράφηκαν οι σχεδιαστικοί συμβιβασμοί που έπρεπε να γίνουν. Είναι πολύ πιθανό, άλλωστε, σε μελλοντικές σχετικές εργασίες που περιλαμβάνουν σχεδίαση βιολογικής βάσης δεδομένων, να πρέπει να αντιμετωπιστούν ξανά παρόμοια προβλήματα· και, μάλιστα, όχι κατ' ανάγκην σε εργασίες που σχετίζονται με miRNAs. Δεδομένου, λοιπόν, πως αυτές οι δυσκολίες ήδη εντοπίστηκαν και επιλύθηκαν, θεωρήθηκε σημαντικό να αναφερθεί ο τρόπος που αντιμετωπίστηκαν. Τα ζητήματα παρουσιάζονται με τη σειρά που προέκυψαν κατά τη σχεδιαστική μελέτη της βάσης δεδομένων.

7.1 Γενικές σχεδιαστικές αρχές

Στη σχεδίαση μιας βάσης δεδομένων υπάρχουν πολλοί παράγοντες που επηρεάζουν όπως η επεκτασιμότητα (scalability), η ευκολία συντήρησης και ενημερώσεων της βάσης, ζητήματα διάβρωσης δεδομένων (data corruption), ζητήματα ταχύτητας, η πολυπλοκότητα των ερωτημάτων κτλ. Ένα παράδειγμα επεκτασιμότητας που απασχολεί την εφαρμογή μας είναι το πόσο εύκολα μπορεί να προστεθεί στο σύστημα ένας νέος αλγόριθμος ή ένας νέος οργανισμός. Ένα παράδειγμα πιθανής διάβρωσης δεδομένων είναι κατά πόσον οι πληροφορίες των επιμέρους οργανισμών βρίσκονται αποθηκευμένες στην ίδια βάση δεδομένων ή όχι και, συνεπώς, αν «κινδυνεύουν» τα δεδομένα ενός οργανισμού να καταστούν απροσπέλαστα αν συμβεί κάτι στα δεδομένα ενός άλλου οργανισμού.

Για να επιτευχθούν ταυτόχρονα όλα τα παραπάνω απαιτείται να γίνουν σχεδιαστικοί συμβιβασμοί σε διάφορα στάδια της σχεδίασης. Μία γενική παρατήρηση που προέκυψε, πάντως, από αυτή την εργασία είναι ότι οι περισσότερες σχεδιαστικές λεπτομέρειες που αφορούν το σχήμα της βάσης, ανάγονται, τελικά, σε δύο θεμελιώδεις παράγοντες:

1) πόσο σύνθετες πληροφορίες θα αποθηκεύονται στη βάση,
2) πόσο γρήγορα πρέπει η βάση να απαντά στα ερωτήματα.

Σε όλα τα ζητήματα που θα συζητηθούν στις επόμενες ενότητες ο βασικός γνώμονας για τις επιλογές μας ήταν να αποθηκευτούν όλες οι ζητούμενες πληροφορίες στη βάση δεδομένων αλλά με τέτοιο τρόπο ώστε να καθίσταται εφικτή η κατά το δυνατόν γρηγορότερη εκτέλεση των ερωτημάτων. Το τελικό συμπέρασμα ως προς τη σχεδίαση της βάσης δεδομένων είναι πως όσο προσπαθεί κανείς να ενισχύσει τον ένα παράγοντα εκ των δύο, αναγκαστικά θα μειώνει την αποτελεσματικότητα του άλλου. Όσο πιο σύνθετες πληροφορίες εισάγονται στη βάση δεδομένων και όσο πιο πολύπλοκα ερωτήματα πρέπει να υποστηρίζονται, τόσο πιο πολύ θα καθυστερεί η βάση δεδομένων να απαντά. Και, στον αντίποδα, όσο απλοποιείται μία σχεδίαση προς όφελος της ταχύτητας, τόσο λιγότερο σύνθετες πληροφορίες και λιγότερο πολύπλοκα ερωτήματα θα μπορούν να υποστηρίζονται από τη βάση δεδομένων.
Εν κατακλείδι, όλες οι σχεδιαστικές αποφάσεις αποτελούν αναγκαστικούς συμβιβασμούς μεταξύ αντικρουόμενων παραγόντων. Η δουλειά του μηχανικού, λοιπόν, είναι να επιλέξει την κατάλληλη ισορροπία μεταξύ όλων αυτών των παραγόντων έτσι ώστε το σύστημα να ανταποκρίνεται με τον «καλύτερο δυνατό τρόπο» στις προδιαγραφές που τίθενται. Βέβαια, σημείο κλειδί για την επιλογή αυτών των ισορροπιών είναι πώς ορίζεται για ένα σύστημα αυτός ο «καλύτερος δυνατός τρόπος». Δηλαδή, δεν παίζουν ρόλο μόνο οι προδιαγραφές που θα τεθούν αλλά και η σχετική προτεραιότητα μεταξύ τους, η οποία είναι και αυτή που θα καθορίσει ποιος πράγματι είναι ένας «καλός» σχεδιαστικός συμβιβασμός.

7.2 Ξεχωριστές βάσεις δεδομένων ανά οργανισμό ή κοινή βάση δεδομένων

Η πρώτη σχεδιαστική επιλογή που έπρεπε να γίνει ήταν κατά πόσον τα δεδομένα του κάθε οργανισμού θα βρίσκονται σε ξεχωριστή βάση δεδομένων ή αν θα βρίσκονται τα δεδομένα όλων των οργανισμών σε μία, κοινή βάση δεδομένων. Στη δεύτερη περίπτωση θα έπρεπε για κάθε οργανισμό να δημιουργηθεί ένα «αντίγραφο» του βασικού σχήματος μέσα στην ίδια βάση δεδομένων και οι πίνακες του κάθε «αντιγράφου» να φέρουν προθέματα ανά οργανισμό, π.χ. human_gene, mouse_gene, human_mirna, mouse_mirna κτλ. Έτσι θα ήταν σαφές ποιοι πίνακες αφορούν τον κάθε οργανισμό.

7.2.1 Πλεονεκτήματα σχεδίασης με ξεχωριστή βάση δεδομένων ανά οργανισμό

Η λύση που επιλέχθηκε είναι να υπάρχουν ξεχωριστές βάσεις δεδομένων για κάθε οργανισμό καθώς αυτό είναι επωφελές για πολλούς λόγους:

- Οι πληροφορίες των επιμέρους οργανισμών είναι εντελώς ανεξάρτητες μεταξύ τους. Αφού, σε λογικό επίπεδο, ο διαχωρισμός των πληροφοριών είναι πλήρης, κρίνεται ορθό να υπάρχει διαχωρισμός των πληροφοριών και σε επίπεδο αποθήκευσης.
- Αν οι πληροφορίες όλων των οργανισμών εισαχθούν σε μία βάση δεδομένων τότε δημιουργείται το πρόβλημα ότι πρέπει να βρεθεί τρόπος να διαχωρίζονται σαφώς οι πληροφορίες των επιμέρους οργανισμών. Δημιουργούμε, δηλαδή, ένα πρόβλημα που δεν υπήρχε αρχικά αφού οι πληροφορίες των οργανισμών είναι εξ ορισμού ανεξάρτητες. Για να επιλυθεί το πρόβλημα αυτό θα πρέπει, για παράδειγμα, να εισάγονται προθέματα στους πίνακες της βάσης που αφορούν τον κάθε οργανισμό.
- Η εφαρμογή δεν ζητείται να υποστηρίζει συγκριτική παρουσίαση πληροφοριών από διαφορετικούς οργανισμούς. Άρα το σύστημα ουδέποτε θα κληθεί να συνδυάσει πληροφορίες από διαφορετικές βάσεις δεδομένων την ίδια στιγμή. Επομένως η επιλογή αυτή δεν επιβαρύνει την εκτέλεση της εφαρμογής. Είναι γνωστό πως τα ερωτήματα σε πίνακες από διαφορετικές βάσεις δεδομένων είναι αρκετά επιβαρυντικά σε θέματα απόδοσης. Ωστόσο δεν υπάρχει τέτοια περίπτωση στη δική μας εφαρμογή.
- Οι βάσεις δεδομένων των οργανισμών, όντας ανεξάρτητες μεταξύ τους, είναι καλύτερα προστατευμένες η μία από την άλλη από ενδεχόμενα διάβρωσης δεδομένων (corrupt data).
- Μπορούν και οι δύο βάσεις δεδομένων να έχουν επακριβώς το ίδιο σχήμα, ίδια ονόματα πινάκων, ίδια ονόματα πεδίων, ίδια ονόματα κλειδιών κτλ. Άρα το σχήμα της βάσης δεδομένων ορίζεται μία φορά μόνο. Επίσης, οποιαδήποτε συντήρηση απαιτηθεί στο μέλλον (π.χ. αλλαγή σχήματος ενός πίνακα) ορίζεται μία φορά και εκτελείται αυτούσια σε όλες τις βάσεις δεδομένων. Σε αντίθετη περίπτωση, ο κώδικας SQL που ορίζει το σχήμα της βάσης δεδομένων θα περιείχε επαναλήψεις, μία επανάληψη για κάθε οργανισμό που πρέπει να προστεθεί. Επιπρόσθετα, σε κάθε επανάληψη θα έπρεπε τα ονόματα των πινάκων, των πρωτευόντων κλειδιών, των ξένων κλειδιών, ορισμένα πεδία κτλ. να λάβουν το σωστό πρόθεμα. Είναι ευνόητο ότι σε αρκετά σημεία αυτής της διαδικασίας μπορούν να υπεισέλθουν σφάλματα.
- Η ανταλλαγή πληροφοριών μεταξύ της εφαρμογής και της βάσης δεδομένων είναι πλήρως απαλλαγμένη από την αναγκαιότητα να ελέγχει η εφαρμογή για ποιον οργανισμό ζητά πληροφορίες ώστε να προσαρμόζει διαρκώς τα προθέματα των πινάκων. Για παράδειγμα, αν χρησιμοποιηθεί κοινή βάση δεδομένων θα πρέπει να γίνονται έλεγχοι σε διάφορα σημεία της εφαρμογής για το αν πρέπει να εκτελεστεί ένα ερώτημα στο human_gene ή το mouse_gene. Σύμφωνα με τη σχεδίασή μας, αυτό δε συμβαίνει ποτέ. Αντίθετα, η ύπαρξη ξεχωριστών βάσεων επιτρέπει στην εφαρμογή να ορίσει μία φορά με ποια βάση δεδομένων θα συνδεθεί (hsadb ή mmudb) και κατόπιν η εφαρμογή μπορεί να «αδιαφορήσει» για το ποια είναι η βάση με την οποία επικοινωνεί. Η επακόλουθη ανταλλαγή πληροφοριών είναι επακριβώς η ίδια ανεξαρτήτως οργανισμού.
- Για τον ίδιο λόγο, αφού όλες οι βάσεις έχουν επακριβώς το ίδιο σχήμα, η αντιμετώπιση τους από την εφαρμογή γίνεται πάρα πολύ εύκολη. Πολύ βασικό είναι ότι μπορούν να αξιοποιηθούν τα ίδια ακριβώς μοντέλα (κλάσεις που επικοινωνούν με τη βάση δεδομένων) για οποιονδήποτε οργανισμό, κάτι το οποίο θα παρουσίαζε περιπλοκές αν επιλεγόταν η σχεδίαση με προθέματα. Σαφώς και θα

μπορούσαν να υλοποιηθούν μοντέλα που προσαρμόζουν τον πίνακα της βάσης από τον οποίο θα αντλήσουν δεδομένα. Όμως, σε κάθε περίπτωση, εισάγεται επιπλέον κώδικας και, συνεπώς, επιπλέον πολυπλοκότητα, επιπλέον πιθανότητες λαθών κτλ.

- Η προσθήκη νέων οργανισμών είναι πολύ εύκολη διαδικασία και δεν επηρεάζει καθόλου τις προϋπάρχουσες βάσεις δεδομένων. Ένας νέος οργανισμός μπορεί να προστεθεί χωρίς να σταματήσουν να λειτουργούν οι βάσεις των προηγούμενων οργανισμών.
- Η ενημέρωση των δεδομένων κάθε οργανισμού είναι πλήρως ανεξάρτητη.
- Τελικά, για όλους τους παραπάνω λόγους, η σχεδίαση αυτή επιτρέπει να γραφτεί λιγότερος κώδικας (άρα και λιγότερος κώδικας προς αποσφαλμάτωση και συντήρηση) τόσο σε επίπεδο βάσης δεδομένων (SQL) όσο και σε επίπεδο εφαρμογής (PHP).

Για όλους αυτούς τους λόγους επιλέχθηκε οι πληροφορίες κάθε οργανισμού να είναι αποθηκευμένες σε ξεχωριστή βάση δεδομένων. Βεβαίως και οι δύο βάσεις δεδομένων βρίσκονται στην ίδια εγκατάσταση (ίδιο instance) της MySQL.

7.2.2 Εναλλακτική προσέγγιση για ενιαία βάση δεδομένων

Αν ήταν αναγκαίο να υπάρχει μία μόνο βάση δεδομένων για όλους τους οργανισμούς, υπάρχει και μια άλλη εναλλακτική προσέγγιση. Αυτή θα ήταν να αποθηκεύεται στη βάση δεδομένων και ο κωδικός ταξινόμησης του οργανισμού (9606 για τον άνθρωπο, 10090 για τον ποντικό). Ο κωδικός οργανισμού θα έπρεπε να υπάρχει σε όλους τους πίνακες της βάσης ώστε να ορίζεται σαφώς ποιον οργανισμό αφορά ένα γονίδιο, miRNA ή αλληλεπίδραση.

Ένα σημαντικό πλεονέκτημα εδώ είναι πως αποφεύγεται η επαναληψιμότητα της προηγούμενης εναλλακτικής. Δηλαδή δε θα χρειαζόταν να υπάρχουν «αντίγραφα» του σχήματος για κάθε οργανισμό. Αντίθετα, το σχήμα της βάσης θα ήταν μόνο ένα και θα μπορούσε να αποθηκεύει τα δεδομένα όλων των οργανισμών.

Ωστόσο και αυτή η λύση παρουσιάζει μειονεκτήματα. Πρώτον, όλα τα πρωτεύοντα κλειδιά θα αποκτούσαν ως δεύτερο πεδίο τον κωδικό του οργανισμού. Όμως η σχεδίαση πρωτευόντων κλειδιών με πολλαπλά πεδία δεν είναι προτιμητέα λύση αν μπορεί να αποφευχθεί. Αντίστοιχα, όλες οι επακόλουθες λειτουργίες της βάσης θα έπρεπε να ελέγχουν δύο πεδία αντί για ένα και θα γίνονταν πιο αργές. Αυτό ισχύει κυρίως για τις πράξεις που χρησιμοποιούν τα πρωτεύοντα ή δευτερεύοντα κλειδιά και, ειδικότερα, για τις ενώσεις οι οποίες, μάλιστα, είναι και υπολογιστικά «ακριβές» πράξεις. Τρίτον, ένας πίνακας interaction που θα περιείχε όλους τους οργανισμούς θα έφτανε (μόνο με τα τρέχοντα δεδομένα) τα 40 εκατομμύρια γραμμές. Αν, μάλιστα, επιλεγόταν ένα διαφορετικό σχήμα με αποθήκευση δεδομένων σε επίπεδο μεταγράφου αντί γονιδίου (βλέπε ενότητα 7.4) τότε το μέγεθος του πίνακα interaction θα πλησίαζε τα 200 εκατομμύρια γραμμές μόνο για τους 2 οργανισμούς που έχουν επιλεχθεί.

Επομένως γίνεται αντιληπτό πως αυτή η σχεδιαστική επιλογή ενέχει πολύ σημαντικά μειονεκτήματα. Παρ' ότι δεν προτιμήθηκε, κρίθηκε σκόπιμο να αναφερθεί διότι λήφθηκε υπ' όψη και αξιολογήθηκε.

7.3 Συντεταγμένες θέσεων πρόσδεσης (binding sites)

Μία επιλογή που έπρεπε να γίνει είναι αν η εφαρμογή θα παρουσιάζει πληροφορίες για τις θέσεις πρόσδεσης των miRNAs επάνω στα γονίδια. Εάν κάτι τέτοιο είναι επιθυμητό, τότε προκύπτουν δύο προβλήματα. Το πρώτο πρόβλημα, που θα συζητηθεί εδώ, σχετίζεται με τις συντεταγμένες των θέσεων πρόσδεσης. Το δεύτερο σχετίζεται με το αν θα παρουσιάζονται οι βαθμολογίες αλληλεπίδρασης ανά γονίδιο ή ανά θέση πρόσδεσης και θα συζητηθεί στην ενότητα 7.7.

7.3.1 Το πρόβλημα των συντεταγμένων ανάλογα με την έκδοση του γονιδίου

Διαφορετικοί αλγόριθμοι χρησιμοποιούν διαφορετικές εκδόσεις της Ensembl και, συνεπώς, σε ορισμένα γονίδια θα υπάρχουν διαφορές μεταξύ των εκδόσεων. Αυτό σημαίνει ότι μπορεί δύο αλγόριθμοι να προβλέπουν την ίδια αλληλεπίδραση γονιδίου–miRNA αλλά αφού έχουν χρησιμοποιήσει διαφορετική έκδοση του γονιδίου, είναι πολύ πιθανό η αλληλουχία που έχουν χρησιμοποιήσει οι δύο αλγόριθμοι να είναι διαφορετική (πιθανό και όχι βέβαιο διότι η έκδοση ενός γονιδίου μπορεί να έχει αλλάξει για άλλο λόγο και όχι εξ αιτίας αλλαγής στην ακολουθία του). Επομένως, σε μία τέτοια περίπτωση, με τι κριτήριο θα συγκριθούν οι θέσεις πρόσδεσης που δίνει ο ένας αλγόριθμος με τις θέσεις πρόσδεσης που δίνει ο δεύτερος;

Οι θέσεις πρόσδεσης χαρακτηρίζονται από τις εξής συντεταγμένες επάνω στην ακολουθία του γονιδίου: αριθμός βάσης της αλληλουχίας του γονιδίου στην οποία αρχίζει η πρόσδεση και το μήκος της θέσης πρόσδεσης. Σύγκριση συντεταγμένων μπορεί να γίνει μόνο αν το σύστημα αναφοράς είναι το ίδιο και σε αυτή την περίπτωση το «σύστημα αναφοράς» είναι η ακολουθία του γονιδίου. Άρα, αν δεν έχει χρησιμοποιηθεί η ίδια ακολουθία του γονιδίου από δύο αλγορίθμους, δεν μπορούν να συγκριθούν οι συντεταγμένες των θέσεων πρόσδεσης που προβλέπονται.

Αυτός ήταν ο λόγος που επιλέχθηκε η εργασία αυτή να εστιάσει μόνο στις βαθμολογίες των αλληλεπιδράσεων και όχι στα χαρακτηριστικά των θέσεων πρόσδεσης.

7.3.2 Προτεινόμενη αντιμετώπιση

Εάν σε κάποια μελλοντική εργασία γίνει προσπάθεια να παρουσιάζονται συγκριτικά και οι θέσεις πρόσδεσης τότε η προτεινόμενη λύση του προβλήματος είναι να χρησιμοποιηθούν εργαλεία «ευθυγράμμισης ακολουθιών» (sequence alignment). Από την ευθυγράμμιση των ακολουθιών των διαφορετικών εκδόσεων του γονιδίου θα προκύψει ένα «υπερσύνολο» της ακολουθίας όλων των εκδόσεων ώστε να δημιουργηθεί ένα κοινό σύστημα αναφοράς. Στην ευθυγραμμισμένη ακολουθία εντοπίζονται τόσο οι κοινές περιοχές των επιμέρους ακολουθιών όσο και οι μη κοινές περιοχές. Έτσι, επάνω στην ευθυγραμμισμένη ακολουθία, θα μπορούν να συγκριθούν με μεγάλη σαφήνεια οι θέσεις πρόσδεσης, όπως προσδιορίζονται από τους διάφορους αλγορίθμους.

Έστω, για παράδειγμα, ότι δύο αλγόριθμοι προβλέπουν την ίδια αλληλεπίδραση γονιδίου–miRNA αλλά με διαφορετική έκδοση του γονιδίου. Σε αυτή την περίπτωση, αυτό που έχει σημασία για τη σύγκριση των

θέσεων πρόσδεσης δεν είναι οι «απόλυτες» συντεταγμένες τους, π.χ. ο αριθμός νουκλεοτιδίου της θέσης πρόσδεσης στη μία έκδοση του γονιδίου και στην άλλη. Αυτό που είναι σημαντικό να εντοπιστεί είναι αν η θέση πρόσδεσης στις δύο περιπτώσεις βρίσκεται επάνω σε κοινό τμήμα των δύο ακολουθιών, δηλαδή σε περιοχή του γονιδίου που δεν μεταβλήθηκε μεταξύ των δύο εκδόσεων. Αν αυτό ισχύει τότε οι δύο αλγόριθμοι έχουν εντοπίσει πράγματι την ίδια θέση πρόσδεσης, παρ' ότι χρησιμοποιήθηκαν διαφορετικές εκδόσεις του γονιδίου. Αν όχι, τότε είτε δεν πρόκειται για ίδια θέση πρόσδεσης είτε η θέση πρόσδεσης εξαρτάται από κάποιο μη κοινό τμήμα. Χρησιμοποιώντας, λοιπόν, την τεχνική ευθυγράμμισης ακολουθιών μπορεί να γίνει αυτή η διαπίστωση.

Τέλος, πρέπει να διευκρινιστεί πως τόσο η πληροφορία των θέσεων πρόσδεσης όσο και το πρόβλημα που συζητήθηκε στην ενότητα αυτή, δεν εξαρτώνται από το αν η αποθήκευση των αλληλεπιδράσεων στη βάση δεδομένων γίνεται σε επίπεδο γονιδίου ή μεταγράφου. Και αυτό διότι, ούτως ή άλλως, κάθε σημείο πάνω σε ένα μετάγραφο μπορεί, με κατάλληλη διαδικασία, να βρεθεί σε ποιο σημείο του γονιδίου αντιστοιχεί. Στην ενότητα αυτή εξηγήσαμε το πρόβλημα σε επίπεδο γονιδίου. Αν επιλεγόταν το επίπεδο μεταγράφου δεν αλλάζει τίποτα απολύτως στη θεώρηση του προβλήματος ούτε και στην προτεινόμενη λύση – εφαρμόζεται ακριβώς η ίδια λογική απλά επάνω σε συγκεκριμένο μετάγραφο αντί επάνω σε ολόκληρο το γονίδιο.

7.4 Αποθήκευση αλληλεπιδράσεων ανά γονίδιο ή ανά μετάγραφο

Η δεύτερη πολύ βασική σχεδιαστική επιλογή που έπρεπε να γίνει είναι αν η αποθήκευση των αλληλεπιδράσεων θα γίνει σε επίπεδο γονιδίου ή μεταγράφου. Παρακάτω παρουσιάζουμε τα τρία κριτήρια που έπαιξαν ρόλο στην επιλογή καθώς και το σκεπτικό της τελικής λύσης.

7.4.1 Πληροφορίες που πρέπει να είναι διαθέσιμες

Όπως εξηγήθηκε, μία αλληλεπίδραση συμβαίνει μεταξύ ενός μεταγράφου και ενός miRNA. Υπάρχουν περιπτώσεις, όμως, που πολλά μετάγραφα ενός γονιδίου αλληλεπιδρούν με κάποιο miRNA. Ακόμη, υπάρχουν περιπτώσεις που μπορεί μόνο κάποια συγκεκριμένα μετάγραφα ενός γονιδίου να αλληλεπιδρούν με κάποιο miRNA ενώ άλλα μετάγραφά του όχι. Επίσης, αν ενδιαφέρουν πληροφορίες σχετικά με τις θέσεις πρόσδεσης (binding sites), τότε μπορεί ορισμένες να υπάρχουν σε πολλά μετάγραφα (αν η θέση πρόσδεσης βρίσκεται πάνω σε κοινό τμήμα των μεταγράφων, δηλαδή σε εξόνιο που υπάρχει σε όλα τα μετάγραφα) ενώ άλλες να υπάρχουν μόνο πάνω σε συγκεκριμένο μετάγραφο του γονιδίου.

Όλες αυτές οι περιπτώσεις καταδεικνύουν το πρώτο κριτήριο που πρέπει να τεθεί σχετικά με αυτή την επιλογή και αυτό είναι ποιες πληροφορίες ενδιαφέρει να παρουσιάζονται στο χρήστη.

7.4.2 Μέγεθος του πίνακα interaction σε επίπεδο γονιδίου ή μεταγράφου

Ο δεύτερος παράγοντας που παίζει σημαντικό ρόλο είναι το πλήθος των εγγραφών που θα έχει ο πίνακας interaction σε συνάρτηση με το τι ερωτήματα θα γίνονται προς τη βάση δεδομένων. Ο πίνακας interaction στην σχεδίαση ανά γονίδιο περιέχει περίπου 20 εκατομμύρια εγγραφές (για κάθε οργανισμό). Στη σχεδίαση

ανά μετάγραφο έχει περίπου 200 εκατομμύρια εγγραφές. Αυτό συμβαίνει επειδή κάθε αλληλεπίδραση γονιδίου–miRNA μπορεί να περιλαμβάνει πολλαπλές αλληλεπιδράσεις μετάγραφων–miRNA. Έτσι το μέγεθος του πίνακα interaction αυξάνεται σημαντικά.

7.4.3 Αποθήκευση ξεχωριστών αλληλεπιδράσεων ανά αλγόριθμο ή όχι

Για τη σχεδίαση του πίνακα interaction, εκτός από την επιλογή αποθήκευσης ανά γονίδιο ή ανά μετάγραφο, πρέπει να γίνει και επιλογή για την αποθήκευση των αλληλεπιδράσεων ξεχωριστά ανά αλγόριθμο ή για όλους μαζί. Αυτό το κριτήριο επίσης επηρεάζει το μέγεθος του πίνακα interaction καθώς και την ταχύτητα εκτέλεσης των ερωτημάτων.
Για καλύτερη επεξήγηση παραθέτουμε το τρέχον σχήμα του πίνακα interaction:

```
gene_id, mirna_id, score_diana, score_targetscan, score_mirtarget
```

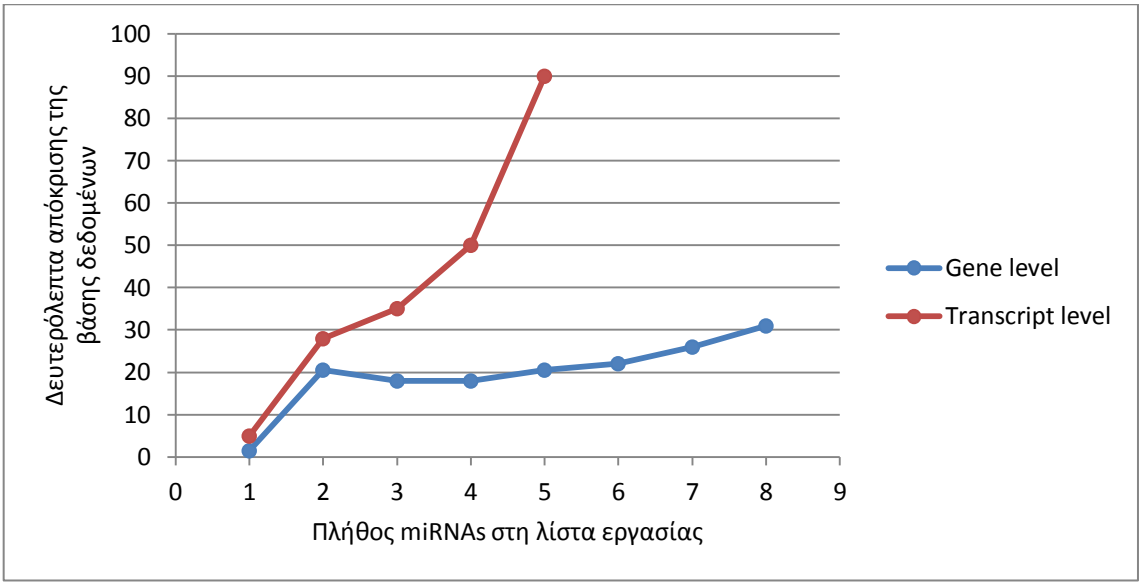
και ένα εναλλακτικό σχήμα:

```
gene_id, mirna_id, tpa_id, score
```

Όταν μια αλληλεπίδραση προβλέπεται από 3 αλγορίθμους, στο πρώτο σχήμα χρειάζεται μία εγγραφή στον πίνακα interaction ενώ, στη δεύτερη περίπτωση, χρειάζονται 3 εγγραφές. Άρα αυτή η επιλογή επίσης παίζει σημαντικό ρόλο για το πόσο θα είναι το τελικό μέγεθος του πίνακα. Στην περίπτωσή μας επιλέχθηκε το πρώτο σχήμα για να περιοριστεί το μέγεθος του πίνακα και να είναι γρηγορότερη η απόκριση της βάσης δεδομένων.
Ωστόσο είναι σαφές πως το πρώτο σχήμα δεν επιτρέπει αποθήκευση πληροφορίας σχετικά με διαφορετικές θέσεις πρόσδεσης που ενδέχεται να προβλέπουν οι διαφορετικοί αλγόριθμοι. Στον αντίποδα, το δεύτερο σχήμα επιτρέπει (αν προστεθούν κατάλληλα πεδία) περισσότερες εξατομικευμένες πληροφορίες ανά αλγόριθμο για την εκάστοτε αλληλεπίδραση. Επίσης είναι σαφές πως το πρώτο σχήμα θα απαντά στα ερωτήματα σαφώς γρηγορότερα συγκριτικά με το δεύτερο.

7.4.4 Σύγκριση ταχύτητας (benchmark) για τη σχεδίαση ανά γονίδιο και ανά μετάγραφο

Για να ληφθεί οριστική απόφαση αποφασίστηκε να γίνει ένα πείραμα το οποίο θα συγκρίνει την ταχύτητα των δύο πιθανών σχεδιάσεων. Η αρχική σχεδίαση του πίνακα interaction ήταν ανά μετάγραφο ενώ, όταν διαπιστώθηκε ότι αργεί αρκετά, μεταβλήθηκε το σχήμα του πίνακα και το πείραμα επαναλήφθηκε.
Κατά την υλοποίηση της εφαρμογής και με τις δοκιμές που πραγματοποιήθηκαν, παρατηρήθηκε πως το πιο απαιτητικό ερώτημα προκύπτει όταν στη λίστα εργασίας υπάρχουν μόνο miRNAs. Αυτό, μάλιστα, ήταν το πιο απαιτητικό ερώτημα και στις δύο σχεδιάσεις, είτε ανά γονίδιο είτε ανά μετάγραφο. Επομένως αποφασίστηκε να συγκριθεί η ταχύτητα των δύο σχεδιάσεων για αυτό το ερώτημα και προέκυψαν τα παρακάτω αποτελέσματα.



Εικόνα 7.1: σύγκριση ταχύτητας της βάσης δεδομένων με δύο εναλλακτικά σχήματα του πίνακα interaction

Η διαδικασία του πειράματος ήταν η εξής: προστέθηκε το πρώτο miRNA στη λίστα και κατόπιν τέθηκαν όλα τα κατώφλια στο ελάχιστο. Έτσι προσπαθήσαμε να δημιουργήσουμε τις πλέον δυσμενείς συνθήκες για τη βάση δεδομένων. Στη συνέχεια προσθέταμε στη λίστα ένα miRNA κάθε φορά. Τέλος, καταγράφαμε το χρόνο από τη στιγμή που πατήθηκε το πλήκτρο "Done" στο αναδυόμενο παράθυρο προσθήκης αντικειμένων μέχρι να προβληθεί η ιστοσελίδα με τα ανανεωμένα αποτελέσματα. Με το πάτημα του πλήκτρου "Done" αποστέλλεται αίτημα στον εξυπηρετητή για ανανέωση των αποτελεσμάτων, οπότε είναι μία έγκυρη αφετηρία μέτρησης χρόνου.

Ο χρόνος καταγράφηκε με βάση την ένδειξη της κονσόλας (console) στον Mozilla Firefox όπου καταγράφεται πόσος χρόνος απαιτήθηκε ώστε να εξυπηρετηθεί το αίτημα από τον εξυπηρετητή. Σημειώνουμε, εδώ, πως κατεγράφη μόνο ο χρόνος που αφορούσε την εξυπηρέτηση του αιτήματος και όχι ο χρόνος φόρτωσης των αρχείων JavaScript και CSS.

7.4.5 Συμπέρασμα σύγκρισης

Τα αποτελέσματα του πειράματος δείχνουν ότι για λίγα miRNAs η ταχύτητα είναι συγκρίσιμη. Όμως, η διαφορά των δύο σχεδιάσεων στις αναζητήσεις με πολλά miRNAs είναι παραπάνω από εμφανής και απετέλεσε το βασικό λόγο που η τελική σχεδίαση επιλέχθηκε να είναι ανά γονίδιο (και κατ' επέκταση καταλήξαμε σε ένα τόσο απλό σχήμα για τη βάση δεδομένων). Μάλιστα, στη σχεδίαση ανά μετάγραφο, το ερώτημα για 5 miRNAs δεν είχε καν ολοκληρωθεί στα 90 δευτερόλεπτα οπότε, αναγκαστικά, το πείραμα εγκαταλείφθηκε σε εκείνο το σημείο για τη σχεδίαση ανά μετάγραφο.

Όπως γίνεται κατανοητό, δεν είναι αποδεκτό μία βάση δεδομένων να καθυστερεί τόσο πολύ να απαντήσει σε ένα ερώτημα που εμπίπτει στις βασικές περιπτώσεις χρήσης. Συνεπώς η επιλογή αποθήκευσης ανά γονίδιο ήταν επιβεβλημένη.

Το βασικό κίνητρο για να επιλεγεί η αποθήκευση ανά γονίδιο ήταν η ταχύτητα. Παρ' όλα αυτά, η επιλογή αυτή δεν αντιβαίνει στις υπόλοιπες σχεδιαστικές επιλογές που έχουν ήδη γίνει αφού, έτσι κι αλλιώς, δεν απαιτούνται από την εφαρμογή πληροφορίες ανά μετάγραφο.

7.4.6 Τελική επιλογή

Το συμπέρασμα που προκύπτει, λοιπόν, είναι ότι στη σχεδίαση του πίνακα interaction υπήρχαν τέσσερις πιθανές επιλογές:

- αποθήκευση ανά γονίδιο με όλους τους αλγορίθμους μαζί,
- αποθήκευση ανά γονίδιο και ανά αλγόριθμο,
- αποθήκευση ανά μετάγραφο και ανά αλγόριθμο,
- αποθήκευση ανά μετάγραφο με όλους τους αλγορίθμους μαζί.

Κάθε επιλογή έχει τα πλεονεκτήματα και τα μειονεκτήματά που ήδη παρουσιάστηκαν και σχετίζονται τόσο με την ταχύτητα όσο και με τη δυνατότητα αποθήκευσης επιμέρους πληροφοριών για κάθε μετάγραφο ή θέση πρόσδεσης. Ανάλογα με τις πληροφορίες που είναι επιθυμητό να αποθηκεύονται και ανάλογα με το ποια ερωτήματα γίνονται συχνότερα στη βάση δεδομένων, επιλέγεται αναλόγως μία από αυτές τις λύσεις.
Για την εργασία αυτή, λοιπόν, επιλέχθηκε η πρώτη επιλογή για τους εξής τρεις λόγους:

- Δεν ήταν αναγκαίο να παρουσιάζονται ξεχωριστές πληροφορίες ανά μετάγραφο ή ανά θέση πρόσδεσης.
- Ως συνέπεια του προηγούμενου δεν υπήρχε άλλη πληροφορία, πέραν της βαθμολογίας, που να αλλάζει ανά αλγόριθμο.
- Η ταχύτητα της βάσης δεδομένων ανά γονίδιο ήταν συντριπτικά καλύτερη απ' ότι ανά μετάγραφο.

Όλοι οι παραπάνω λόγοι οδήγησαν, τελικά, στην αποθήκευση ανά γονίδιο και για όλους τους αλγορίθμους μαζί. Έτσι για κάθε αλληλεπίδραση αρκεί μία εγγραφή στον πίνακα interaction και σε αυτή τη μία εγγραφή περιέχονται όλες οι πληροφορίες τις αλληλεπίδρασης.

7.5 Η οντότητα «γονίδιο»

Σύμφωνα με τη θεωρία των σχεσιακών βάσεων δεδομένων [1], κάθε πίνακας μιας βάσης αποτελεί ένα σύνολο όμοιων οντοτήτων, δηλαδή οντοτήτων που έχουν τις ίδιες ιδιότητες και χαρακτηρίζονται μοναδικά από ένα συγκεκριμένο υποσύνολο των ιδιοτήτων τους. Κάθε εγγραφή ενός πίνακα (πρέπει να) αποτελεί μία, διακριτή, αδιαίρετη και μοναδική οντότητα αυτού του συνόλου. Επομένως, για να σχεδιαστεί κατάλληλα ο πίνακας "gene" της βάσης δεδομένων, πρέπει πρώτα να καθοριστεί σαφώς τι ορίζουμε ως οντότητα «γονίδιο». Αυτός ο ορισμός, όμως, περιπλέκεται δεδομένου ότι τα γονίδια παρουσιάζουν εκδόσεις. Επομένως, υπάρχουν δύο πιθανοί ορισμοί της οντότητας «γονίδιο».

7.5.1 Ορισμός πρώτος: μία οντότητα ανά γονίδιο

Σε αυτή τη θεώρηση, δύο (ή περισσότερες) διαφορετικές εκδόσεις του ίδιου γονιδίου, αποτελούν μία οντότητα και δεν θεωρούνται ξεχωριστά αντικείμενα. Ανεξαρτήτως έκδοσης γονιδίου και, συνεπώς, ανεξαρτήτως έκδοσης Ensembl, κάθε γονίδιο (δηλαδή κάθε μοναδικό αναγνωριστικό) θα αποτελεί μία εγγραφή στον πίνακα gene.

Πλεονέκτημα: σημαντική απλότητα στο μοντέλο, συμφωνεί με τη διαισθητική αντιμετώπιση του όρου «γονίδιο» και είναι απαλλαγμένο από τις όποιες περιπλοκές εισάγουν οι εκδόσεις.

Μειονέκτημα: δυσχεραίνει τη μοντελοποίηση καταστάσεων όπου ενδιαφέρουν επιμέρους ιδιότητες του γονιδίου οι οποίες αλλάζουν από έκδοση σε έκδοση, όπως η αλληλουχία του.

7.5.2 Ορισμός δεύτερος: μία οντότητα ανά έκδοση γονιδίου

Σε αυτή τη θεώρηση κάθε διαφορετική έκδοση γονιδίου θεωρείται ξεχωριστή οντότητα και επομένως θα αποτελεί ξεχωριστή εγγραφή στον πίνακα gene.

Πλεονέκτημα: μπορεί να μοντελοποιήσει καταστάσεις όπου ενδιαφέρουν επιμέρους ιδιότητες του γονιδίου. Μπορεί να αποθηκευθεί σημαντικά μεγαλύτερη ποικιλία πληροφοριών.

Μειονέκτημα: εισάγει περιπλοκή ως προς τα ξένα κλειδιά που αναφέρονται στον πίνακα gene. Η περιπλοκή αφορά το σε ποια απ' όλες τις εγγραφές ενός γονιδίου πρέπει να αναφέρονται τα ξένα κλειδιά από άλλους πίνακες της βάσης.

Αν επιλεχθεί ο ορισμός αυτός πρέπει, με κάποιο τρόπο, να αντιμετωπιστεί το μειονέκτημά του. Σε αυτή την περίπτωση μία ορθότερη σχεδίαση μάλλον απαιτεί ο πίνακας gene να διασπαστεί σε δύο επιμέρους πίνακες. Ο ένας θα αφορά τα σταθερά στοιχεία των γονιδίων (μοναδικό αναγνωριστικό, όνομα, περιγραφή) και ο δεύτερος θα αφορά τις επιμέρους εκδόσεις των γονιδίων και όσες ιδιότητες αλλάζουν (λίστα μεταγράφων, αλληλουχία, θέση στο γονιδίωμα).

7.5.3 Ανάλυση επιλογών

Κατά τη μελέτη των αλγορίθμων πρόβλεψης διαπιστώθηκε ότι κανείς από τους αλγορίθμους δε χρησιμοποιεί την τρέχουσα έκδοση της Ensembl. Επομένως, προκύπτει ως πιθανότητα να πρέπει να χρησιμοποιηθούν για την εφαρμογή προηγούμενες εκδόσεις της Ensembl και όχι η τρέχουσα. Ωστόσο, δεν είναι επιθυμητό (αν δε συντρέχει σοβαρότατος λόγος) να βασιστεί η εφαρμογή σε παλαιότερα δεδομένα τα οποία, επί της ουσίας, δεν θεωρούνται πλέον έγκυρα.

Όπως μπορεί να υποθέσει κανείς, υφίστανται γονίδια που, παρ' ότι οι αλγόριθμοι τα άντλησαν από παλαιότερες εκδόσεις της Ensembl, υπάρχουν ακόμη και οι ιδιότητές τους έχουν ενημερωθεί. Μάλιστα, αυτό συμβαίνει στην πολύ μεγάλη πλειοψηφία των αποτελεσμάτων των αλγορίθμων. Άρα προκύπτει το ερώτημα: υπάρχει κάποιος λόγος να φορτωθεί στην εφαρμογή μία παλαιότερη έκδοση της Ensembl και όχι η τρέχουσα; Με άλλα λόγια, υπάρχει λόγος να χρησιμοποιηθούν οι παλαιότερες ιδιότητες των γονιδίων ακόμη και αν αυτά τα γονίδια υπάρχουν ακόμη και έχουν υποστεί αλλαγές;

Η απάντηση είναι η «χειρότερη» δυνατή: εξαρτάται. Εξαρτάται από τις καταστάσεις που θέλουμε να μοντελοποιήσουμε, από τις πληροφορίες που θέλουμε να διατηρήσουμε και από τις δυνατότητες που θέλουμε να υποστηρίζει η εφαρμογή. Αν, λοιπόν, για να υλοποιηθούν οι επιθυμητές δυνατότητες της εφαρμογής απαιτείται η πληροφορία της ακριβούς έκδοσης ενός γονιδίου, τότε αναγκαστικά πρέπει να επιλεχθεί οντότητα ανά έκδοση και, επιπλέον, να φορτωθούν όλες οι εκδόσεις της Ensembl που χρησιμοποιήθηκαν από τους αλγορίθμους. Αν, όμως, αυτό δεν απαιτείται, τότε μπορεί να επιλεχθεί οντότητα απλώς ανά γονίδιο, μπορούν να αγνοηθούν οι εκδόσεις και, συνεπώς, μπορεί να φορτωθεί η τρέχουσα έκδοση της Ensembl.

7.5.4 Τελική επιλογή ορισμού

Σε αυτή την εργασία έγιναν οι εξής τρεις επιλογές:

1) εστιάσαμε μόνο στη βαθμολογία των αλληλεπιδράσεων και όχι στις θέσεις πρόσδεσης (binding sites). Άρα αφού δεν ενδιέφερε να παρουσιαστεί στο χρήστη η θέση πρόσδεσης, απαλλασσόμαστε από την αναγκαιότητα να κρατηθεί η αλληλουχία των γονιδίων, χαρακτηριστικό που εξαρτάται από την έκδοση.
2) επιλέχθηκε να παρουσιαστούν στο χρήστη οι αλληλεπιδράσεις σε επίπεδο γονιδίου και όχι σε επίπεδο μεταγράφου. Άρα δεν επηρεάζει την εφαρμογή η λίστα μεταγράφων του γονιδίου, χαρακτηριστικό που επίσης εξαρτάται από την έκδοση.
3) γενικότερα στην εφαρμογή, δεν ενδιέφερε να κρατηθεί ιστορικό της εξέλιξης των εκδόσεων των γονιδίων καθώς σε κανένα σημείο της δεν χρειάζονται ιδιότητες των γονιδίων που μεταβάλλονται μεταξύ εκδόσεων.

Αυτά τα τρία σημεία, λοιπόν, επιτρέπουν να θεωρηθεί μία οντότητα ανά γονίδιο αγνοώντας τις εκδόσεις. Κατ' επέκταση, ήταν εφικτό να χρησιμοποιηθεί η πιο πρόσφατη έκδοση του κάθε γονιδίου ακόμη κι αν οι αλγόριθμοι εκτελέστηκαν επάνω σε προηγούμενες εκδόσεις του.

7.6 Εκδόσεις Ensembl

Ο κάθε αλγόριθμος πρόβλεψης λαμβάνει τα δεδομένα των γονιδίων από κάποια πηγή. Κυριότερο στοιχείο που ενδιαφέρει τους αλγορίθμους είναι η αλληλουχία του γονιδίου πάνω στην οποία θα εκτελεστεί ο εκάστοτε αλγόριθμος. Η Ensembl, ως βάση δεδομένων για γονιδιώματα, αποτελεί τη συχνότερα επιλεγόμενη πηγή δεδομένων από τους αλγορίθμους πρόβλεψης[1].

Όπως, όμως, εξηγήθηκε ήδη στο κεφάλαιο 3, υπάρχουν συγκεκριμένοι λόγοι για τους οποίους τα γονίδια διαθέτουν εκδόσεις καθώς και λόγοι για τους οποίους και η ίδια η Ensembl ανανεώνεται σε εκδόσεις και όχι διαρκώς. Για αυτόν ακριβώς το λόγο προκύπτουν τα ακόλουθα τέσσερα προβλήματα:

[1] Άλλες δημοφιλείς πηγές για αλληλουχίες γονιδίων είναι η "NCBI" και η "UCSC Genome Browser". Ωστόσο, κατά την έρευνα που πραγματοποιήθηκε γύρω από τους αλγορίθμους πρόβλεψης, παρατηρήθηκε πως οι περισσότεροι και, κυρίως, οι πιο ευρέως χρησιμοποιούμενοι, ακολουθούν την Ensembl.

i) **Δε χρησιμοποιούν όλοι οι αλγόριθμοι πρόβλεψης την ίδια έκδοση Ensembl**. Το ποια έκδοση χρησιμοποιεί ο κάθε αλγόριθμος εξαρτάται από το πότε δημοσιεύθηκε ή πότε έγινε η πιο πρόσφατη ανανέωση των αποτελεσμάτων του.

ii) **Κανείς από τους αλγορίθμους δε χρησιμοποιεί την τρέχουσα έκδοση της Ensembl**. Άρα εξ ορισμού υπάρχει ζήτημα σχετικά με το αν θα καταστεί αναγκαίο να χτιστεί η εφαρμογή επάνω σε προηγούμενη (outdated) έκδοση της Ensembl.

iii) **Μεταξύ δύο διαφορετικών εκδόσεων Ensembl, ορισμένα γονίδια μπορεί να βρίσκονται στην ίδια έκδοσή τους (έκδοση γονιδίου) ενώ άλλα σε διαφορετική**. Αυτό εξαρτάται από το αν το εκάστοτε γονίδιο υπέστη μεταβολές ή όχι στο διάστημα που μεσολάβησε μεταξύ των δύο εκδόσεων της Ensembl.

iv) **μεταξύ δύο διαφορετικών εκδόσεων Ensembl, η μεταγενέστερη θα περιέχει ενδεχομένως νέα γονίδια ενώ δεν θα περιέχει γονίδια που στο μεσοδιάστημα καταργήθηκαν**.

Δεδομένων αυτών των τεσσάρων προβλημάτων, το δεύτερο σχεδιαστικό ζήτημα που ανακύπτει συνοψίζεται στην εξής ερώτηση: από ποια έκδοση της Ensembl πρέπει να αντληθούν οι πληροφορίες για τα γονίδια;

7.6.1 Πιθανές λύσεις

Η απάντηση στην προηγούμενη ερώτηση δεν μπορεί να είναι μία συγκεκριμένη έκδοση Ensembl. Αυτό θα αποτελούσε επιλογή μόνο αν όλοι οι αλγόριθμοι χρησιμοποιούσαν την ίδια έκδοση Ensembl – κάτι που, όπως επισημάνθηκε στο πρόβλημα (i), δεν ισχύει. Εάν επιλεγόταν μία συγκεκριμένη έκδοση Ensembl και κάποιος αλγόριθμος δεν τη χρησιμοποιούσε, τότε λόγω του προβλήματος (iv) θα προέκυπτε η εξής κατάσταση: ο αλγόριθμος να προβλέπει αλληλεπιδράσεις γονιδίων τα οποία δεν υπάρχουν στη βάση δεδομένων. Άρα, ο συνδυασμός των προβλημάτων (i) και (iv) δεν επιτρέπει να επιλεχθεί αποκλειστικά μία έκδοση της Ensembl και να χρησιμοποιηθεί αυτούσια.

Πιθανή δεύτερη λύση θα μπορούσε να αποτελέσει να φορτωθούν στη βάση δεδομένων όλες οι εκδόσεις της Ensembl που έχουν χρησιμοποιηθεί από τους αλγορίθμους. Π.χ. να φορτωθεί η βάση δεδομένων με τις Ensembl 75 & 77. Αυτό, ωστόσο, και πάλι δεν αποτελεί εφικτή επιλογή διότι όσα γονίδια τυχαίνει να υπάρχουν και στις δυο εκδόσεις θα καταγράφονταν στη βάση δεδομένων δύο φορές. Μάλιστα θα υπήρχε σύγκρουση πληροφοριών (conflict) αφού στη βάση δεδομένων ορισμένα γονίδια θα εμφανίζονταν διπλή φορά με ίδια έκδοση ενώ άλλα θα εμφανίζονταν διπλή φορά με άλλη έκδοση και, συνεπώς, αντικρουόμενα στοιχεία σχετικά με τις ιδιότητες του γονιδίου. Αυτό προκύπτει λόγω του προβλήματος (iii).

Λόγω όλων αυτών θα διαμορφωνόταν μία ασυνεπής κατάσταση για τη βάση δεδομένων. Για παράδειγμα: αν ένα γονίδιο εμφανίζεται διπλή φορά, σε ποια από τις δύο εγγραφές θα πρέπει να αποδοθούν οι αλληλεπιδράσεις του γονιδίου; Τα ξένα κλειδιά που αναφέρονται από έναν άλλο πίνακα της βάσης προς τον πίνακα gene, σε ποια από τις δύο εγγραφές του ίδιου γονιδίου θα έπρεπε να αναφέρονται;

7.6.2 Τελική λύση

Το σκεπτικό που οδήγησε στην τελική λύση στηρίζεται στα ακόλουθα σημεία:

- Όπως εξηγήθηκε στην ενότητα 7.5, μπορούν να αγνοηθούν οι εκδόσεις των γονιδίων.
- Δε συντρέχει κανένας λόγος που να καθιστά αναγκαίο να χρησιμοποιηθούν παλαιότερες ή, εν γένει, συγκεκριμένες εκδόσεις της Ensembl και όχι η τρέχουσα.
- Είναι επιθυμητό οι λεπτομερείς κάθε γονιδίου να ανταποκρίνονται στην τρέχουσα έκδοση Ensembl ή, για τα καταργηθέντα γονίδια, στην τελευταία Ensembl έκδοση που τα περιείχε. Αυτό επιλέγεται έτσι ώστε οι ιδιότητες κάθε γονιδίου να είναι οι πιο πρόσφατα ενημερωμένες.
- Για όσα γονίδια υπάρχουν στα αποτελέσματα των αλγορίθμων αλλά έχουν πλέον καταργηθεί, είναι επιθυμητό να διατηρηθούν οι προβλεφθείσες αλληλεπιδράσεις τους και να είναι διαθέσιμες στην εφαρμογή.

Έτσι επιλέχθηκε η εξής λύση: η φόρτωση της βάσης δεδομένων θα ξεκινήσει με την έκδοση 86 και θα συνεχίσει «προς τα πίσω» μέχρι την έκδοση 75, περιλαμβάνοντας σε κάθε προηγούμενη έκδοση μόνο όσα γονίδια καταργήθηκαν. Το πώς έγινε αυτό αναλύθηκε λεπτομερώς ήδη στην παράγραφο 6.3.1.

7.7 Βαθμολογίες αλληλεπιδράσεων ανά γονίδιο

Ο τρόπος που οι αλγόριθμοι βαθμολογούν τις αλληλεπιδράσεις δεν είναι κοινός. Αυτό προκύπτει από τον τρόπο με τον οποίο εντοπίζουν τις αλληλεπιδράσεις οι διάφοροι αλγόριθμοι. Παρακάτω εξηγείται πώς αποδίδουν βαθμολογίες στις αλληλεπιδράσεις οι τρεις αλγόριθμοι που χρησιμοποιήθηκαν στην εργασία αυτή.

Ο DIANA–microT λαμβάνει το κυρίαρχο μετάγραφο κάθε γονιδίου και εντοπίζει επάνω σε αυτό τις πιθανές θέσεις πρόσδεσης (binding sites) ενός miRNA. Κατόπιν βαθμολογεί κάθε θέση πρόσδεσης ξεχωριστά και, τέλος, εξάγεται μία συνολική βαθμολογία που χαρακτηρίζει την αλληλεπίδραση. Έτσι ο DIANA–microT για κάθε αλληλεπίδραση παρουσιάζει δύο ειδών βαθμολογίες: μία συνολική (προϋπολογισμένη) και μία ανά θέση πρόσδεσης.

Ο TargetScan για κάθε γονίδιο λαμβάνει όλα τα μετάγραφά του και τα ελέγχει όλα για πιθανές θέσεις πρόσδεσης. Για όσες θέσεις πρόσδεσης εντοπιστούν αποδίδει μία βαθμολογία στην κάθε μία εξ αυτών χωρίς, όμως, να αποδίδει και συνολική βαθμολογία για την αλληλεπίδραση. Σε αυτή την περίπτωση, λοιπόν, για κάθε αλληλεπίδραση υπάρχει ενός είδους βαθμολογία: μία ανά θέση πρόσδεσης. Μάλιστα, υπάρχει η ιδιομορφία ότι μπορεί μία αλληλεπίδραση να συμβαίνει σε πολλές διαφορετικές θέσεις πρόσδεσης και σε πολλά μετάγραφα του γονιδίου – όχι μόνο σε ένα μετάγραφο. Εδώ πρέπει να σημειωθεί και κάτι ακόμη: ενδέχεται μία θέση πρόσδεσης που εντοπίζεται σε δύο ή περισσότερα μετάγραφα, να είναι στην πραγματικότητα η ίδια θέση πρόσδεσης επάνω στην ακολουθία ολόκληρου του γονιδίου. Επομένως, ο τρόπος που ο TargetScan αποδίδει τις βαθμολογίες εισάγει επιπλέον περιπλοκές.

Ο MirTarget για κάθε αλληλεπίδραση γονιδίου–miRNA αποδίδει μία βαθμολογία μόνο, τη βαθμολογία της αλληλεπίδρασης, χωρίς να δίνει καθόλου θέση πρόσδεσης.

Συμπερασματικά, οι διαφορετικοί τρόποι βαθμολόγησης μιας αλληλεπίδρασης μπορεί να είναι:

- μία βαθμολογία ανά γονίδιο,
- μία βαθμολογία ανά μετάγραφο,
- μία βαθμολογία ανά θέση πρόσδεσης,
- στην περίπτωσης της βαθμολογίας ανά μετάγραφο ή ανά θέση πρόσδεσης, μπορεί ο αλγόριθμος να αποδίδει και συνολική βαθμολογία στο γονίδιο ή όχι.

Όπως δείχτηκε, λοιπόν, κάθε αλγόριθμος πρόβλεψης ακολουθεί διαφορετική στρατηγική για τη βαθμολόγηση μιας αλληλεπίδρασης. Όμως, για να μπορούν να έχουν νόημα συγκρίσεις μεταξύ των αλγορίθμων, έπρεπε να βρεθεί ένας κοινός τόπος μεταξύ των αλγορίθμων. Κι αφού αυτός ο κοινός τόπος δεν υφίσταται εκ των προτέρων, έπρεπε να επιλεχθεί μία σύμβαση για αυτό.

7.7.1 Επιλεγμένη λύση

Αφού επιλέχθηκε να παρουσιάζονται οι αλληλεπιδράσεις σε επίπεδο γονιδίου–miRNA, λογικό είναι και η βαθμολογία κάθε αλληλεπίδρασης να αφορά το γονίδιο συνολικά και όχι ένα συγκεκριμένο μετάγραφό του ή μια μεμονωμένη θέση πρόσδεσης.

Για τον DIANA–microT και τον MirTarget δεν ήταν αναγκαίο να γίνει κάποια συμβιβαστική επιλογή αφού, εξ ορισμού, και οι δύο αλγόριθμοι δίνουν βαθμολογία ανά γονίδιο. Για τον DIANA–microT συγκεκριμένα, δεν ενοχλεί το γεγονός ότι δίνει βαθμολογία και ανά θέση πρόσδεσης αφού η βαθμολογία του γονιδίου συνολικά (που είναι αυτή που μας ενδιαφέρει) είναι ήδη προϋπολογισμένη από τον ίδιο τον αλγόριθμο – άρα δεν γίνεται καμία επέμβαση επ' αυτού.

Ο TargetScan, όμως, δίνει βαθμολογία μόνο ανά θέση πρόσδεσης. Επομένως, εδώ επιλέχθηκε η εξής σύμβαση: για κάθε αλληλεπίδραση γονιδίου–miRNA θα βρεθούν όλες οι θέσεις πρόσδεσης σε οποιοδήποτε μετάγραφο κι αν εμφανίζονται και ως βαθμολογία της αλληλεπίδρασης θα ληφθεί η καλύτερη βαθμολογία των θέσεων πρόσδεσης.

7.7.2 Αιτιολόγηση

Η σύμβαση που επιλέχθηκε πρέπει να αιτιολογηθεί για δύο πιθανές περιπτώσεις, δηλαδή να υπάρχει μία θέση πρόσδεσης ή περισσότερες. Στην περίπτωση που κάποια αλληλεπίδραση παρουσιάζει μόνο μία θέση πρόσδεσης τότε, προφανώς, η βαθμολογία της αλληλεπίδρασης αυτής θα είναι ίση με τη βαθμολογία της μοναδικής θέσης πρόσδεσης. Οπότε εδώ το κριτήριο της καλύτερης βαθμολογίας αρκεί.

Σχετικά με την ύπαρξη πολλαπλών θέσεων πρόσδεσης, σύμφωνα με τη μελέτη [22] (η οποία αναφέρεται και στη δημοσίευση του TargetScan), όταν υπάρχουν πολλαπλές θέσεις πρόσδεσης τότε συνήθως δρουν συνεργατικά και επαυξάνουν την αποτελεσματικότητα της αλληλεπίδρασης. Στη μελέτη επισημαίνεται πως η συνδυαστική επίδραση δύο ταυτόχρονων προσδέσεων **ίδιου miRNA** σε ένα μετάγραφο είναι σχεδόν τόσο αποτελεσματική όσο θα ήταν οι δύο επιμέρους προσδέσεις ξεχωριστά. Μάλιστα, αν δύο θέσεις πρόσδεσης είναι σε μικρή απόσταση μεταξύ τους*, παρατηρήθηκε πως η συνδυαστική επίδρασή τους είναι αποτελεσματικότερη απ' ότι οι δύο επιμέρους θέσεις ξεχωριστά. Όταν οι δύο θέσεις βρίσκονται αρκετά

μακριά τότε χάνεται το συνδυαστικό πλεονέκτημα αλλά, σε κάθε περίπτωση πάντως, παραμένει η ανεξάρτητη αποτελεσματικότητα των δύο επιμέρους θέσεων.

Επομένως, για αλληλεπιδράσεις που παρουσιάζουν πολλαπλές θέσεις πρόσδεσης, είναι γνωστό πως η συνδυαστική επίδραση θα είναι τουλάχιστον τόσο ισχυρή όσο η ισχυρότερη εκ των θέσεων πρόσδεσης. Άρα, αν πρέπει να αποδοθεί μία βαθμολογία στο γονίδιο συνολικά, αυτή θα είναι τουλάχιστον ίση με την καλύτερη βαθμολογία των επιμέρους θέσεων πρόσδεσης. Έτσι, αιτιολογείται η επιλογή να ληφθεί ως βαθμολογία της αλληλεπίδρασης σε επίπεδο γονιδίου η καλύτερη βαθμολογία των θέσεων πρόσδεσης.

7.7.3 Πιθανός τρόπος υπολογισμού συνδυαστικής επίδρασης πολλαπλών θέσεων πρόσδεσης

Όπως εξηγήθηκε, η σύμβαση που επιλέχθηκε ίσως να υποτιμά τη συνδυαστική αλληλεπίδραση των πολλαπλών θέσεων πρόσδεσης σε κάποιες περιπτώσεις. Παρ' όλα αυτά δε θα μπορούσε να οριστεί ένας τεκμηριωμένος υπολογισμός για τη συνδυαστική επίδραση των πολλαπλών θέσεων πρόσδεσης διότι δεν υπάρχει σαφής τρόπος να συνδέσουμε τα συμπεράσματα της ανωτέρω μελέτης με τον τρόπο βαθμολόγησης του TargetScan. Η βαθμολογία που αποδίδει ο TargetScan σε κάθε θέση πρόσδεσης βασίζεται σε 14 διαφορετικούς παράγοντες και δεν είναι εφικτό να καθοριστεί αν τα ανωτέρω συμπεράσματα επηρεάζουν κάποιους από αυτούς ή όλους. Άρα δε θα μπορούσε να οριστεί π.χ. μία απλή πρόσθεση των βαθμολογιών των επιμέρους θέσεων πρόσδεσης ως συνδυαστική βαθμολογία.

Επιπλέον, ορισμένα από τα κριτήρια του TargetScan ήδη αφορούν το πλήθος των θέσεων πρόσδεσης επάνω σε ένα μετάγραφο. Επομένως, αυτή η συνδυαστική επίδραση έχει ήδη ληφθεί υπόψη ως ένα βαθμό από τον ίδιο τον αλγόριθμο. Άρα ενισχύεται η ορθότητα της επιλογής μας να μην οριστεί κάποιος αυθαίρετος, συνδυαστικός υπολογισμός αλλά να αρκεστούμε στην καλύτερη βαθμολογία που δίνεται από τον ίδιο τον TargetScan.

Σημείωση: κλείνοντας την ενότητα παρατίθεται μία σημείωση για τον πολύ διερευνητικό αναγνώστη. Η ανωτέρω μελέτη επισημαίνει ότι αν δύο θέσεις πρόσδεσης είναι αρκετά κοντά μεταξύ τους (<8 νουκλεοτίδια απόσταση) και αφορούν **διαφορετικά miRNAs**, τότε αυτές οι δύο θέσεις πρόσδεσης δρουν ανταγωνιστικά μεταξύ τους*. Αυτή η παρατήρηση μοιάζει να αντιβαίνει στην έως τώρα επιλογή μας. Ωστόσο, στη δική μας περίπτωση αυτή η παρατήρηση δεν ισχύει διότι αφορά θέσεις πρόσδεσης **διαφορετικών miRNAs** ενώ το πρόβλημα που προσπαθήσαμε να επιλύσουμε εδώ αφορά θέσεις πρόσδεσης **ίδιων miRNAs**. Όταν στα αποτελέσματα του TargetScan εντοπίζονται πολλαπλές θέσεις πρόσδεσης για μια δεδομένη αλληλεπίδραση γονιδίου–miRNA, αυτές οι πολλαπλές θέσεις αφορούν το ίδιο miRNA.

Παρατηρήστε, λοιπόν, τη διαφορά στα δύο σημεία με αστερίσκο: αν κοντινές θέσεις πρόσδεσης αφορούν το ίδιο miRNA τότε δρουν συνεργατικά. Αν αφορούν διαφορετικά miRNAs, τότε δρουν ανταγωνιστικά. Η διαφορά αυτή επισημαίνεται στη μελέτη [22].

7.8 Εκτεταμένο μοντέλο

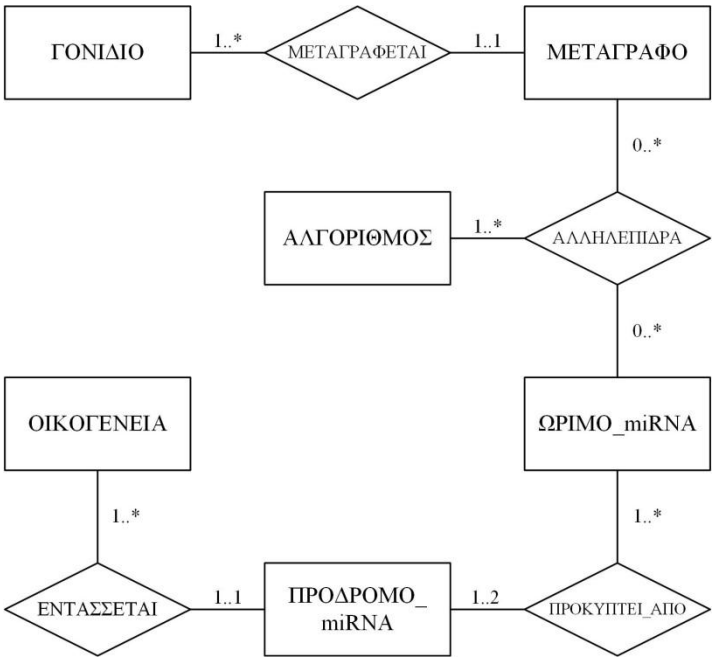
Κλείνοντας το κεφάλαιο, παρουσιάζεται ένα εκτεταμένο μοντέλο οντοτήτων–συσχετίσεων. Αυτό το μοντέλο προέκυψε από την ανάλυση που προηγήθηκε και αποτελεί υπερσύνολο του μοντέλου που χρησιμοποιήθηκε στην εφαρμογή. Το εκτεταμένο μοντέλο κρίθηκε σκόπιμο να καταγραφεί για τρεις λόγους:

1) αποθηκεύει ένα μεγαλύτερο σύνολο βιολογικών πληροφοριών γύρω από τις αλληλεπιδράσεις,
2) μπορεί να υποστηρίξει ενισχυμένη λειτουργικότητα σε πιθανή μελλοντική επέκταση της εφαρμογής,
3) ενσωματώνει όλες τις παραμέτρους που συζητήθηκαν στο παρόν κεφάλαιο και έτσι παρέχει κατάλληλη αντιμετώπιση για όλα τα σχεδιαστικά ζητήματα που εντοπίστηκαν.

7.8.1 Διάγραμμα οντοτήτων–συσχετίσεων

Οι διαφορές με το μοντέλο που χρησιμοποιήθηκε στην εφαρμογή είναι οι εξής:

- αποθήκευση αλληλεπιδράσεων σε επίπεδο μεταγράφου,
- πληροφορίες για τις αλληλεπιδράσεις σχετικά με τις θέσεις πρόσδεσης,
- πληροφορίες για τα πρόδρομα miRNAs,
- πληροφορίες για τις οικογένειες των πρόδρομων miRNAs.



Εικόνα 7.2: το διάγραμμα οντοτήτων συσχετίσεων του εκτεταμένου μοντέλου

8

Επίλογος

8.1 Σύνοψη

Τα miRNAs αποτελούν ένα πεδίο της βιολογίας με αυξημένο ερευνητικό και πρακτικό ενδιαφέρον ενώ ένα πολύ σημαντικό εργαλείο που αξιοποιείται για τη μελέτη τους είναι οι αλγόριθμοι πρόβλεψης αλληλεπιδράσεων. Οι προβλέψεις των αλγορίθμων, ωστόσο, δεν επαληθεύονται πάντα στην πραγματικότητα ενώ αρκετές φορές διαφορετικοί αλγόριθμοι παρουσιάζουν αντικρουόμενα αποτελέσματα. Συνεπώς, η συνδυαστική μελέτη προβλέψεων από διαφορετικούς αλγορίθμους είναι ιδιαίτερα χρήσιμη ώστε να εντοπίζονται με μεγαλύτερη ακρίβεια αλληλεπιδράσεις που πράγματι συμβαίνουν. Παρ' όλα αυτά, όμως, αυτή τη στιγμή δεν υπάρχουν διαθέσιμα κατάλληλα εργαλεία που να υποστηρίζουν συγκριτική παρουσίαση προβλέψεων από διάφορους αλγορίθμους.

Το αντικείμενο αυτής της διπλωματικής εργασίας, ως μία προσπάθεια συνεισφοράς προς αυτήν την κατεύθυνση, ήταν να αναπτυχθεί μία διαδικτυακή εφαρμογή η οποία να υποστηρίζει την παρουσίαση προβλέψεων από πολλούς αλγορίθμους ταυτόχρονα. Παρέχονται, επίσης, 3 διαφορετικά γραφήματα τα οποία διευκολύνουν την κατανόηση των αποτελεσμάτων. Μάλιστα, το ακόμη σημαντικότερο είναι πως τα γραφήματα είναι διαδραστικά και ο χρήστης μπορεί να αλληλεπιδράσει μαζί τους. Έτσι μπορεί να αλλάζει τις επιλογές της αναζήτησης και να βλέπει αμέσως πώς αυτό επηρεάζει τα αποτελέσματα που παρουσιάζονται. Η χρήση των γραφημάτων αποτελεί ένα περαιτέρω βήμα για την καλύτερη δυνατή αξιοποίηση των συγκρίσεων μεταξύ των αλγορίθμων.

Εκτός από τα οφέλη που προσφέρει η εφαρμογή, ο τρόπος υλοποίησής της μπορεί να αποτελέσει οδηγό του πώς να κατασκευάζονται εφαρμογές βιοπληροφορικής οι οποίες υποστηρίζουν συγκριτική παρουσίαση πληροφοριών.

8.2 Συμπεράσματα

Τα κυριότερα συμπεράσματα που προέκυψαν από αυτή την εργασία είναι τα εξής:

- **Απλά και μόνο το γεγονός ότι δημιουργείται μία γραφική απεικόνιση δε σημαίνει κατ' ανάγκη ότι η απεικόνιση αυτή είναι και χρήσιμη.** Για να είναι οι απεικονίσεις πράγματι χρήσιμες, πρέπει να επιλέγονται τα κατάλληλα γραφήματα έτσι ώστε να αναδεικνύονται οι «κρυμμένες» κανονικότητες (μοτίβα) που υπάρχουν στα δεδομένα. Είναι, επίσης, αναγκαίο να παρέχονται γραφήματα που εξετάζουν διαφορετικές πτυχές των δεδομένων ώστε να ενισχύεται η πιθανότητα να προκύψει πράγματι ένα συμπέρασμα από κάποιο γράφημα. Τα συμπεράσματα που θα προκύψουν από κάθε τύπο γραφήματος είναι διαφορετικά και επομένως όσο περισσότερα είναι τα γραφήματα τόσο ευρύτερα είναι τα πιθανά αποτελέσματα. Για παράδειγμα, μπορεί δύο γραφήματα να έχουν διαφορετική εμφάνιση αλλά να οδηγούν στα ίδια συμπεράσματα ενώ ο ίδιος τύπος γραφήματος απεικονίζοντας διαφορετικές πτυχές του ζητήματος να οδηγήσει σε πολλαπλά συμπεράσματα.
- **Οι στατικές απεικονίσεις δεν συνεισφέρουν σημαντικά**. Τα γραφήματα πρέπει να είναι διαδραστικά ώστε ο χρήστης να μπορεί μέσω του γραφήματος κυριολεκτικά να «χειριστεί» τα δεδομένα. Να μπορεί, δηλαδή, να δει σε πραγματικό χρόνο πώς οι επιλογές της αναζήτησής του επηρεάζουν τα δεδομένα και να εντοπίσει κανονικότητες που μπορεί να προκύπτουν. Άλλωστε, τα συμπεράσματα σπανίως προκύπτουν από την εξέταση ενός μεμονωμένου συνόλου δεδομένων. Αντίθετα, η αντιπαραβολή διαφορετικών δεδομένων και οι μεταβολές που συμβαίνουν είναι τα στοιχεία που οδηγούν σε χρήσιμα συμπεράσματα.
- **Οι πηγές των βιολογικών δεδομένων παρουσιάζουν εγγενείς «ασυμβατότητες» οι οποίες παίζουν σημαντικό ρόλο, αφενός, στο πώς θα παρουσιάζονται οι πληροφορίες προς το χρήστη και, αφετέρου, στο πώς θα αποθηκεύονται οι πληροφορίες κατάλληλα στη βάση δεδομένων.** Η σχεδίαση μιας βιολογικής βάσης δεδομένων απαιτεί ιδιαίτερη επιμέλεια ως προς τις συμβάσεις που αναγκαστικά θα γίνουν για να επιτευχθεί ο συγκερασμός των δεδομένων απ' τις διάφορες πηγές. Ειδικότερα, για τους αλγορίθμους πρόβλεψης, είναι σημαντικό να επιλεχθεί κατάλληλο σχήμα για τη βάση δεδομένων το οποίο να μπορεί να υποστηρίξει την αποθήκευση των πληροφοριών όλων των επιμέρους αλγορίθμων. Και αυτό είναι και δύσκολο αλλά και σημαντικό καθότι κάθε αλγόριθμος παρέχει τα αποτελέσματά του με διαφορετική δομή.

  Επιπλέον, με δεδομένο ότι τα αποτελέσματα των αλγορίθμων παρουσιάζουν διαφορές οι οποίες δεν ευνοούν πάντα τις συγκρίσεις, πρέπει να βρεθεί κατάλληλος τρόπος συγκριτικής παρουσίασης των αποτελεσμάτων έτσι ώστε αυτό που θα βλέπει ο χρήστης, τελικά, να έχει πράγματι νόημα.

  Επίσης, οι ασυμβατότητες είχαν να κάνουν και με τις βιολογικές βάσεις δεδομένων αυτές καθαυτές, όπως η Ensembl και η miRBase. Στα κεφάλαια 6 και 7 αναλύθηκαν εκτενέστατα διάφορα προβλήματα που σχετίζονται με τις εκδόσεις των βιολογικών βάσεων δεδομένων και πώς αυτά τα προβλήματα επηρέασαν τη σχεδίαση.

  Όλοι αυτοί οι παράγοντες εισάγουν πολυπλοκότητες που επηρεάζουν τη σχεδίαση της εφαρμογής,

τη σχεδίαση της βάσης δεδομένων αλλά και τον τρόπο που θα επιλεχθεί να παρουσιάζονται οι πληροφορίες προς το χρήστη.

- **Η σχεδίαση μιας βιολογικής βάσης δεδομένων είναι αρκετά απαιτητική διαδικασία ως προς την ταχύτητά της βάσης**. Υπάρχουν δύο θεμελιώδεις παράγοντες που έρχονται σε ευθεία αντίθεση με την ταχύτητα της βάσης δεδομένων. Ο πρώτος είναι η διατήρηση της πιστότητας* των βιολογικών δεδομένων και, ο δεύτερος, η πολυπλοκότητα των ερωτημάτων που η βάση πρέπει να υποστηρίζει. Η καλύτερη επιλογή σε τέτοιες περιπτώσεις είναι η χρήση υλοποιημένων προβολών (materialised views). Δηλαδή, η βάση δεδομένων, εκτός από τους πίνακες του σχήματος καλό είναι να περιέχει προβολές οι οποίες θα χρησιμοποιούνται έτσι ώστε να βελτιωθεί η ταχύτητα των απαιτητικών ερωτημάτων.

  Αυτό αντιβαίνει, ίσως, στη θεωρία των βάσεων δεδομένων καθώς, με κάθε υλοποιημένη προβολή, διπλασιάζεται κάποιο υποσύνολο των πληροφοριών της βάσης. Ωστόσο, αν αυτό παρέχει σημαντικά οφέλη στην ταχύτητα, τότε θεωρούμε πως είναι αποδεκτή λύση, <u>ειδικά αν στη βάση δεδομένων γίνονται σπάνια εγγραφές</u>.

  Στην περίπτωση αυτής της εφαρμογής, οι χρήστες απλώς διαβάζουν από τη βάση δεδομένων· δε γράφουν **ποτέ** σε αυτήν. Άρα, αν η ακεραιότητα των δεδομένων έχει ελεγχθεί επαρκώς πριν τη φόρτωση της βάσης, τότε μία τέτοια σχεδίαση δεν ενέχει κινδύνους να έρθει η βάση σε ασυνεπή κατάσταση (διαβρωμένα δεδομένα – corrupted data). Αυτό ισχύει, φυσικά, υπό την προϋπόθεση ότι η σχεδίαση του όλου συστήματος (δικαιώματα χρηστών στη βάση δεδομένων, σωστή επικοινωνία της εφαρμογής με τη βάση κτλ.) δεν περιέχει σφάλματα (bugs) που ενδεχομένως επιφέρουν ασυνέπειες στη βάση δεδομένων κατά τη χρήση της εφαρμογής.

* Με τον όρο «πιστότητα» εννοούμε να μη γίνονται απλουστεύσεις στα βιολογικά δεδομένα προκειμένου να εξυπηρετηθεί η αποθήκευσή τους. Αυτές οι απλουστεύσεις ενδεχομένως πηγάζουν από τις εγγενείς διαφορές που έχουν τα βιολογικά δεδομένα των διάφορων πηγών, όπως ακριβώς επισημάνθηκε στο προηγούμενο συμπέρασμα.

8.3 Μελλοντικές επεκτάσεις

Κατά τη διάρκεια εκπόνησης της εργασίας προέκυψαν διάφορες προεκτάσεις του θέματος οι οποίες θα ήταν και ενδιαφέρον αλλά και χρήσιμο να μελετηθούν περαιτέρω στο μέλλον. Οι μελλοντικές επεκτάσεις που προτείνονται απορρέουν από τρεις, κυρίως, παράγοντες. Πρώτον, τα miRNAs είναι ένα πεδίο της βιολογίας με αποδεδειγμένα αρκετό βάθος που απομένει να εξερευνηθεί ενώ ήδη παρουσιάζει έντονο ερευνητικό αλλά και πρακτικό ενδιαφέρον. Δεύτερον, οι αλληλεπιδράσεις γονιδίων–miRNAs αποτελούν ιδανικό πεδίο αξιοποίησης εργαλείων βιοπληροφορικής. Τρίτον, ως άμεσο αποτέλεσμα των δύο προηγούμενων, πιστεύουμε πως το ενδιαφέρον για εργαλεία βιοπληροφορικής σε αυτό το πεδίο θα παραμείνει έντονο. Μάλιστα, τα δύο τελευταία σημεία έχουν ήδη αποδειχτεί αν αναλογιστεί κανείς τον αντίκτυπο που είχαν οι αλγόριθμοι πρόβλεψης στην εξέλιξη της έρευνας γύρω απ’ τα miRNAs.

Ορισμένες προτάσεις, λοιπόν, για μελλοντική μελέτη επάνω στο θέμα είναι οι εξής:

- **Οπτικοποιημένη εξερεύνηση αλληλεπιδράσεων** και όχι μόνο οπτικοποιημένη παρουσίαση των συγκρίσεων. Θα μπορούσαν όλα τα γονίδια και τα miRNAs να τοποθετηθούν εξ αρχής σε έναν μεγάλο γράφο ο οποίος θα απεικονίζει όλες τις γνωστές αλληλεπιδράσεις. Κατόπιν, ο χρήστης θα έχει τη δυνατότητα να περιηγηθεί επάνω στον γράφο αναζητώντας πληροφορίες. Η ποσότητα των απεικονιζόμενων αλληλεπιδράσεων θα μπορούσε ίσως να περιορίζεται με κατάλληλα φίλτρα. Ένα πρόβλημα που θα έπρεπε σίγουρα να αντιμετωπιστεί σε μία τέτοια εργασία είναι ο τρόπος αποθήκευσης ενός τόσο μεγάλου γράφου, πώς αυτός θα κατασκευάζεται τμηματικά (render) στην οθόνη και πώς η κίνηση θα είναι ομαλή χωρίς η εφαρμογή να καθυστερεί. Για αυτά θα μπορούσαν να αξιοποιηθούν συναφείς διπλωματικές εργασίες του εργαστηρίου σχετικά με την εξερεύνηση μεγάλων γράφων.
- **Επέκταση της εφαρμογής με υποστήριξη συγκρίσεων και των θέσεων πρόσδεσης (binding sites).** Στα προηγούμενα κεφάλαια έχουν ήδη εξηγηθεί οι λόγοι για τους οποίους θα ήταν χρήσιμες οι συγκρίσεις των θέσεων πρόσδεσης καθώς, κάτι τέτοιο, θα αύξανε τις δυνατότητες ανάλυσης των αλληλεπιδράσεων σε ακόμη μεγαλύτερο βάθος. Αυτή η επέκταση της εφαρμογής θα ήταν ιδιαίτερα ενδιαφέρουσα διότι:
    i) Διαφορετικοί αλγόριθμοι εξετάζουν διαφορετικές περιοχές των γονιδίων (5΄ UTR, 3΄ UTR, CDS/ORF). Επομένως η σύγκριση των θέσεων πρόσδεσης μπορεί να οδηγήσει σε συμπεράσματα σχετικά με το ποιες περιοχές ευνοούν ή όχι τις αλληλεπιδράσεις. Σε ευρύτερο πλαίσιο, θα μπορούσαν να προκύψουν συμπεράσματα για συγκεκριμένα τμήματα των γονιδίων που είναι «επιρρεπή» ή «ανθεκτικά» σε αλληλεπιδράσεις.
    ii) Κάποιοι αλγόριθμοι εντοπίζουν πολλαπλές θέσεις πρόσδεσης ενός miRNA επάνω στο ίδιο γονίδιο. Αυτές οι πολλαπλές θέσεις πρόσδεσης εντοπίζονται, κατά περίπτωση, επάνω σε ένα μετάγραφο του γονιδίου ή σε περισσότερα. Έτσι, θα μπορούσαν ίσως να εξαχθούν συμπεράσματα που θα συσχετίζουν τις θέσεις πρόσδεσης με τη βαθμολογία που αποδίδεται στην εκάστοτε αλληλεπίδραση ή να εντοπιστούν θέσεις πρόσδεσης που είναι πιο αποτελεσματικές από άλλες.

  Στην ενότητα 7.8 παρουσιάστηκε ήδη ένα εκτεταμένο μοντέλο βάσης δεδομένων που μπορεί να υποστηρίξει την επιπλέον πληροφορία για τις θέσεις πρόσδεσης και το οποίο θα μπορούσε να αξιοποιηθεί από μία τέτοια εργασία.
- **Επέκταση της εφαρμογής με δυνατότητα συγκρίσεων αλληλεπιδράσεων βάσει οικογενειών miRNAs.** Τα miRNAs εντάσσονται σε οικογένειες με βάση την περιοχή πρόσδεσης (seed region). Μία οικογένεια αποτελείται από εκείνα τα miRNAs που έχουν ίδια περιοχή πρόσδεσης αλλά παρουσιάζουν διαφορές στην υπόλοιπη ακολουθία τους (διαφορετικά θα ήταν το ίδιο miRNA). Με κατάλληλη επέκταση της εφαρμογής θα μπορούσαν να παρουσιάζονται συγκρίσεις π.χ. μεταξύ οικογενειών, μεταξύ αντιπροσωπευτικών miRNAs διαφορετικών οικογενειών, να γίνεται εξαγωγή στατιστικών στοιχείων για γονίδια που αλληλεπιδρούν με συγκεκριμένες οικογένειες, ποια miRNAs

μιας οικογένειας είναι πιο «δραστήρια» από άλλα κτλ. Ιδιαίτερη χρησιμότητα σε κάτι τέτοιο θα είχαν τα hive plots όπου θα μπορούσε να δημιουργηθεί ένας άξονας για κάθε οικογένεια miRNA.

- **Εισαγωγή στη βάση δεδομένων των πειραματικώς επαληθευμένων αλληλεπιδράσεων από τη βάση δεδομένων DIANA–TarBase**. Για την προσθήκη αυτή απαιτείται ελάχιστη προσπάθεια καθώς δεν χρειάζεται σχεδόν καμία αλλαγή στη βάση δεδομένων (απαιτείται σίγουρα η προσθήκη μίας στήλης ακόμη στον πίνακα interaction αλλά πιθανότατα καμία άλλη αλλαγή). Για την υποστήριξη από την εφαρμογή, αρκεί η προσθήκη μερικών γραμμών κώδικα στο αρχείο διαμόρφωσης (configuration file). Ακολουθώντας τη διαδικασία όπως περιγράφηκε στις ενότητες 6.1 και 6.3, μπορούν να συλλεχθούν τα δεδομένα του DIANA–TarBase και κατόπιν να φορτωθούν στη βάση δεδομένων. Έτσι, τα δεδομένα των πειραματικώς επαληθευμένων αλληλεπιδράσεων θα είναι διαθέσιμα από την εφαρμογή, ακριβώς σαν να αποτελούσαν έναν ακόμα αλγόριθμο πρόβλεψης.

9

Βιβλιογραφία

Βιβλία & επιστημονικές δημοσιεύσεις

[1] Silberschatz, A., Korth, H. & Sudarshan, S. (2011), Συστήματα Βάσεων Δεδομένων (6η έκδοση), εκδόσεις Μ. Γκιούρδας

[2] Lee, R., Feinbaum, R. & Ambros, V. (1993, December 3), "The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14", *Cell*, 75(5), 843–854, http://dx.doi.org/10.1016/0092-8674(93)90529-Y

[3] Dong Yue, Hui Liu & Yufei Huang (2009, November), "Survey of Computational Algorithms for MicroRNA Target Prediction", *Current Genomics*, 10(7), 478–492, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2808675/

[4] Bartel, D. (2004, January 23), "MicroRNAs: Genomics, Biogenesis, Mechanism, and Function", *Cell*, 116(2), 281–297, http://dx.doi.org/10.1016/S0092-8674(04)00045-5

[5] Cai, Y., Yu, X., Hu, S. & Yu, J. (2009, December), "A Brief Review on the Mechanisms of miRNA Regulation", "*Genomics, Proteomics & Bioinformatics*", 7(4), 147–154, https://dx.doi.org/10.1016%2FS1672-0229(08)60044-3

[6] Witkos. T., Koscianska, E. & Krzyzosiak W. (2011, March), "Practical Aspects of microRNA Target Prediction", "*Current molecular medicine*", 11(2), 93–109 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3182075/

[7] Ekimler, S. & Sahin, K. (2014, September), "Computational Methods for MicroRNA Target Prediction" , *Genes*, 5(3), 671–683, https://dx.doi.org/10.3390/genes5030671

[8] Pio, G., Malerba, D., D' Elia, D. & Ceci, M. (2013, January 10), "Integrating microRNA target predictions for the discovery of gene regulatory networks: a semi-supervised ensemble learning approach", *BMC Bioinformatics*, 15(suppl 1), S4, http://dx.doi.org/10.1186/1471-2105-15-S1-S4

[9] Xiao Fan & Kurgan, L. (2014, December 2), "Comprehensive overview and assessment of computational prediction of microRNA targets in animals", *Briefings in bioinformatics*, 16(5),780–794, https://doi.org/10.1093/bib/bbu044

[10] Zhang, Y. & Verbeek, F. (2010), "Comparison and Integration of Target Prediction Algorithms for microRNA Studies", *Journal of Integrative bioinformatics*, 7(3), 127 http://dx.doi.org/10.2390/biecoll-jib-2010-127

[11] Dong Zou, Lina Ma, Jun Yu & Zhang Zhang (2015, February), "Biological Databases for Human Research", "*Genomics, Proteomics & Bioinformatics*", 13(1), 55–63 http://dx.doi.org/10.1016/j.gpb.2015.01.006

[12] Pruitt, K., Brown, G., Tatusova, T. & Maglott, D. (2012, April 6), "Chapter 18, The Reference Sequence (RefSeq) Database", "*The NCBI handbook [Internet]*" https://www.ncbi.nlm.nih.gov/books/NBK21091/

[13] Kozomara, A. & Griffiths–Jones, S. (2013, November 25), "miRBase: annotating high confidence microRNAs using deep sequencing data", "*Nucleic Acids Research*", 42(D1), D68–D73, https://doi.org/10.1093/nar/gkt1181

[14] Paraskevopoulou, M., Georgakilas, G., Kostoulas, N., Vlachos, I., Vergoulis, T., Reczko, M., Filippidis, C., Dalamagas, T. & Hatzigeorgiou, AG. (2013, July), "DIANA-microT web server v5.0: service integration into miRNA functional analysis workflows", *Nucleic Acids Res*earch, 41(Web Server issue), W169–W173, https://dx.doi.org/10.1093/nar/gkt393

[15] Agarwal, Bell, G., Nam, JW & Bartel, D. (2015, August 12), "Predicting effective microRNA target sites in mammalian mRNAs", *eLife*, http://dx.doi.org/10.7554/eLife.05005

[16] Wong, N. & Wang, X. (2014, November 5), "miRDB: an online resource for microRNA target prediction and functional annotations", *Nucleic Acids Research*, 43(D1), D146–D152, https://doi.org/10.1093/nar/gku1104

[17] Kiriakidou, M., Nelson, P., Kouranov, A., Fitziev, P., Bouyioukos, C., Mourelatos, Z. & Hatzigeorgiou, A. (2004, May 15), "A combined computational-experimental approach predicts human microRNA targets", *Genes & Development*, 18(10), 1165–1178, https://dx.doi.org/10.1101/gad.1184704

[18] Lewis, B., Shih, I., Jones–Rhoades, M., Bartel, D. & Burge, C. (2003, December 26), "Prediction of Mammalian MicroRNA Targets", *Cell*, 115(7), 787–798, http://dx.doi.org/10.1016/S0092-8674(03)01018-3

[19] Wang, X. (2008, June), "miRDB: A microRNA target prediction and functional annotation database with a wiki interface", *RNA journal*, 14(6), 1012–1017, https://dx.doi.org/10.1261/rna.965408

[20] Wang, R., Perez–Riverol, Y. Hermjakob, H. & Vizcaíno, JA (2015,April 8), "Open source libraries and frameworks for biological data visualisation: A guide for developers", *Proteomics*, 15(8), 1356–1374, http://onlinelibrary.wiley.com/doi/10.1002/pmic.201400377/full

[21] Yachdav, G. et al. (2015, July 8), "Cutting Edge: Anatomy of BioJS, an open source community for the life sciences", *eLife*, http://dx.doi.org/10.7554/eLife.07009

[22] Grimson, A., Kai–How Farh, K., Johnston, W., Garrett–Engele, P., Lim, L. & Bartel, D. (2007, July 6), "MicroRNA Targeting Specificity in Mammals: Determinants beyond Seed Pairing", *Molecular Cell*, 27(1), 91–105, http://dx.doi.org/10.1016/j.molcel.2007.06.017

Ιστότοποι

[23] miRWalk 2.0, http://zmf.umm.uni-heidelberg.de/apps/zmf/mirwalk2/index.html

[24] MiRNABlog, "*microRNA Target Prediction Tools*", http://mirnablog.com/microrna-target-prediction-tools/

[25] OmicTools, "*MicroRNA target prediction software tools* " https://omictools.com/mirna-target-prediction-category

[26] VIZBI, "*Visualizing biological data*" https://vizbi.org/

[27] Internet Live Stats, homepage, http://www.internetlivestats.com/

[28] Symfony, "*Why should I use a framework?*", http://symfony.com/why-use-a-framework

[29] Pages of Trygve M. H. Reenskaug, "*MVC XEROX PARC 1978-79*", http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html

[30] Mozilla Developer Network, "*MVC Architecture*", https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture

[31] Google Chrome Developer (documentation website), "*MVC Architecture*" https://developer.chrome.com/apps/app_frameworks

[32] Microsoft Developer Network, "*ASP.NET MVC Overview*", https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx

[33] Ensembl, "*About the Ensembl Project*", http://www.ensembl.org/info/about/index.html

[34] miRBase, *Homepage*, http://mirbase.org/

[35] miRBase, "*What do the miRNA names/identifiers mean?*", http://mirbase.org/help/nomenclature.shtml

[36] Ensembl, "*How to use BioMart*", http://www.ensembl.org/info/data/biomart/how_to_use_biomart.html

[37] BioMart, homepage, http://www.biomart.org/

[38] The European Bioinformatics Institute ftp website, "*EMBL outstation – User manual*", ftp://ftp.ebi.ac.uk/pub/databases/embl/doc/usrman.txt

[39] GitHub, BioJS repository, https://github.com/biojs/biojs

[40] grnet, "*Okeanos*", https://grnet.gr/services/cloud-services/okeanos/

[41] Okeanos, "*About*", https://okeanos.grnet.gr/about/

[42] Microsoft Azure, "*What is Iaas?*", https://azure.microsoft.com/en-us/overview/what-is-iaas/

[43] Interoute, "*What is Iaas?*", http://www.interoute.com/what-iaas

[44] GNU Operating System, "*Gawk: Effective AWK Programming*", https://www.gnu.org/software/gawk/manual/

[45] Grymoire, "*AWK tutorial*", http://www.grymoire.com/Unix/Awk.html

[46] Ubuntu documentation Community Help Wiki, "*ApacheMySQLPHP*", https://help.ubuntu.com/community/ApacheMySQLPHP

[47] Webopedia, "*LAMP*", http://www.webopedia.com/TERM/L/LAMP.html

[48] MySQL, "*Top Reasons to Use MySQL*", https://www.mysql.com/why-mysql/topreasons.html

[49] MySQL, "*Case studies*", http://www.mysql.com/why-mysql/case-studies/

[50] jQuery, http://jquery.com/

[51] Bootstrap, http://getbootstrap.com/

[52] GitHub, Bootstrap repository, https://github.com/twbs/bootstrap

[53] NCBI RefSeq, "*About RefSeq*", https://www.ncbi.nlm.nih.gov/refseq/about/

Άλλες πηγές (άρθρα στον τύπο, εγχειρίδια, παρουσιάσεις κτλ.)

[54] Callaway, E. (2016, July 4), "The visualizations transforming biology", *nature*, http://www.nature.com/news/the-visualizations-transforming-biology-1.20201

[55] Wayner, P. (2015, March 30), "7 reasons why frameworks are the new programming languages", *Infoworld*, http://www.infoworld.com/article/2902242/application-development/7-reasons-why-frameworks-are-the-new-programming-languages.html

[56] Ensembl, "Ensembl introduction: genomes with Ensembl", http://may2012.archive.ensembl.org/info/website/tutorials/Ensembl_introduction.pdf

[57] Kepes, B., "Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS", Rackspace website, https://support.rackspace.com/white-paper/understanding-the-cloud-computing-stack-saas-paas-iaas/

[58] Stutz, M. (2006, September 19), "Get started with GAWK: AWK language fundamentals", *IBM developerWorks* (website), ανακτήθηκε από https://www6.software.ibm.com/developerworks/education/au-gawk/au-gawk-a4.pdf

[59] Skvorc, B. (2015, March 30), "The Best PHP Framework for 2015: SitePoint Survey Results", *Sitepoint*, https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/

[60] Way, J. (2012, November 28), "Why Laravel is Taking the PHP Community by Storm", *envatoTuts+*, https://code.tutsplus.com/tutorials/why-laravel-is-taking-the-php-community-by-storm--pre-52639

[61] Castledine, E. (2011, March 4), "What's so good about jQuery?", *Sitepoint*, https://www.sitepoint.com/whats-so-good-about-jquery/

10

Παράρτημα Ι: Εγχειρίδιο χρήσης της εφαρμογής

Το παρόν κεφάλαιο αποτελεί το αναλυτικό εγχειρίδιο χρήσης της εφαρμογής με επεξήγηση του τρόπου με τον οποίο λειτουργεί η εφαρμογή και παρουσίαση όλων των χαρακτηριστικών και των δυνατοτήτων που αυτή προσφέρει. Στη συνέχεια του κεφαλαίου, για συντομία θα αναφερόμαστε στα γονίδια και τα miRNAs με τον γενικότερο όρο «αντικείμενα», υπό την έννοια ότι αυτά είναι τα αντικείμενα γύρω από τα οποία είναι δομημένη η λειτουργία της εφαρμογής.

10.1 Βασικά βήματα λειτουργίας της εφαρμογής

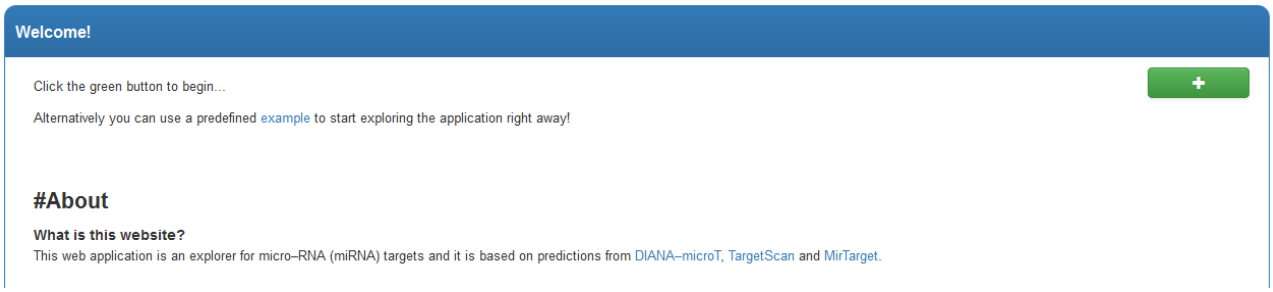
Η εφαρμογή οργανώνει την αναζήτηση, εξερεύνηση και παρουσίαση των αλληλεπιδράσεων σε 3 βασικά βήματα:

1. Επιλογή των αντικειμένων (γονίδια/miRNA) για τα οποία θέλουμε να μελετήσουμε τις αλληλεπιδράσεις.
2. Ορισμός κριτηρίων αναζήτησης (φίλτρα). Έτσι, από όλες τις πιθανές αλληλεπιδράσεις των επιλεγμένων αντικειμένων, θα παρουσιαστούν μόνον όσες πληρούν τα κριτήρια αυτά.
3. Χρήση των διαθέσιμων γραφημάτων ώστε να αξιολογηθούν τα αποτελέσματα που προέκυψαν.

Στη συνέχεια θα εξηγηθούν ένα – ένα αυτά τα βασικά βήματα.

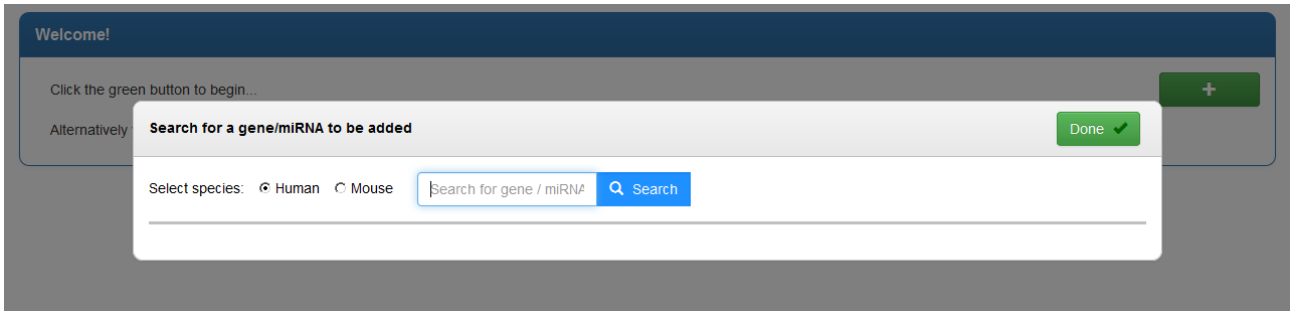
10.2 Ξεκινώντας

Στην εικόνα φαίνεται η αρχική σελίδα της εφαρμογής.



Εικόνα 10.1: το άνω μέρος της αρχικής σελίδας της εφαρμογής

Η πρώτη ενέργεια που πρέπει να κάνει ο χρήστης (όπως υποδεικνύει και το σχετικό μήνυμα) είναι να προσθέσει ένα ή περισσότερα αντικείμενα στη «λίστα εργασίας» του.

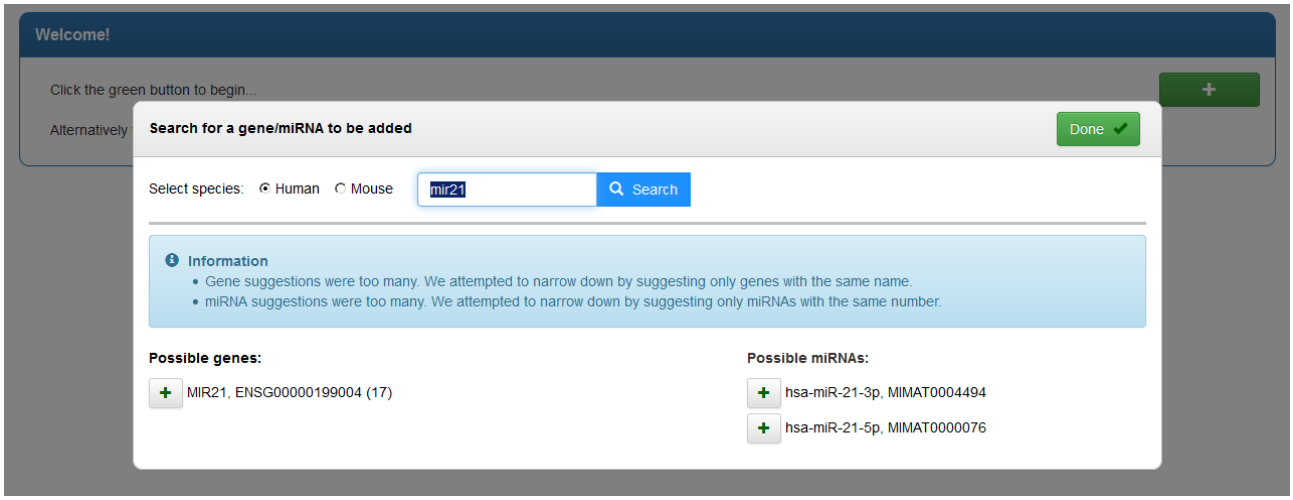


Εικόνα 10.2: το αναδυόμενο παράθυρο αναζήτησης γονιδίων και miRNAs

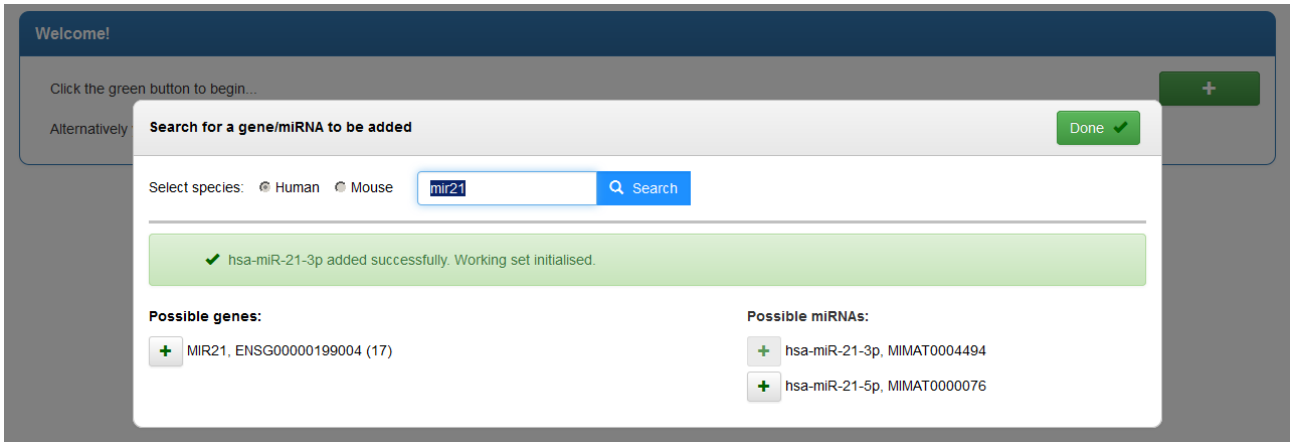
1. Κάνουμε κλικ επάνω στο κουμπί «+» ώστε να εμφανιστεί το αναδυόμενο παράθυρο.
2. Επιλέγουμε για ποιον οργανισμό επιθυμούμε να αναζητήσουμε αντικείμενο.
3. Εισάγουμε τον όρο που θέλουμε να αναζητήσουμε. Τα πιθανά αποτελέσματα της αναζήτησης είναι:
    i. Κανένα αποτέλεσμα. Σε αυτή την περίπτωση θα εμφανιστεί κατάλληλο μήνυμα που επεξηγεί για ποιο λόγο δεν βρέθηκε κανένα αποτέλεσμα στη βάση δεδομένων έτσι ώστε να βοηθήσει το χρήστη να εισάγει έναν καταλληλότερο όρο αναζήτησης.
    ii. Ένα ή περισσότερα προτεινόμενα αποτελέσματα. Ανάλογα με τον όρο αναζήτησης μπορεί να προκύψουν προτεινόμενα γονίδια, miRNAs ή και τα δύο. Ενδεχομένως να εμφανιστούν και διευκρινιστικά μηνύματα που εξηγούν με ποιο τρόπο προέκυψαν τα προτεινόμενα αποτελέσματα. Έτσι, αν το αντικείμενο που αναζητούσε ο χρήστης δεν είναι μέσα στα προτεινόμενα, ο χρήστης προτρέπεται να επαναλάβει κατάλληλα την αναζήτηση.
4. Αν θέλουμε να προσθέσουμε κάποιο από τα διαθέσιμα αποτελέσματα:
    i. Πατάμε το κουμπί "+" αριστερά από το επιθυμητό αντικείμενο ώστε να το προσθέσουμε στη λίστα εργασίας.
    ii. Θα εμφανιστεί μήνυμα επιβεβαίωσης ότι το αντικείμενο προστέθηκε επιτυχώς στη λίστα εργασίας. Επίσης το κουμπί "+" αυτού του αντικειμένου δεν είναι πλέον διαθέσιμο.

iii. Μπορούμε να επιλέξουμε να προσθέσουμε όσα από τα διαθέσιμα αντικείμενα θέλουμε.

5. Είτε προσθέσαμε κάποιο αντικείμενο είτε όχι, μπορούμε να επαναλάβουμε τη διαδικασία από το βήμα 3 κάνοντας νέα αναζήτηση.
6. Μόλις προσθέσουμε όλα τα αντικείμενα που επιθυμούμε κλείνουμε το παράθυρο από το κουμπί "Done" επάνω δεξιά ή πατώντας το πλήκτρο "Esc". Η εφαρμογή θα μας ανακατευθύνει στην κεντρική οθόνη.



Εικόνα 10.3: παράδειγμα αναζήτησης όρου που επιστρέφει προτεινόμενα αποτελέσματα και από τις δύο κατηγορίες αντικειμένων. Είναι εμφανές το διευκρινιστικό μήνυμα και τα κουμπιά προσθήκης.



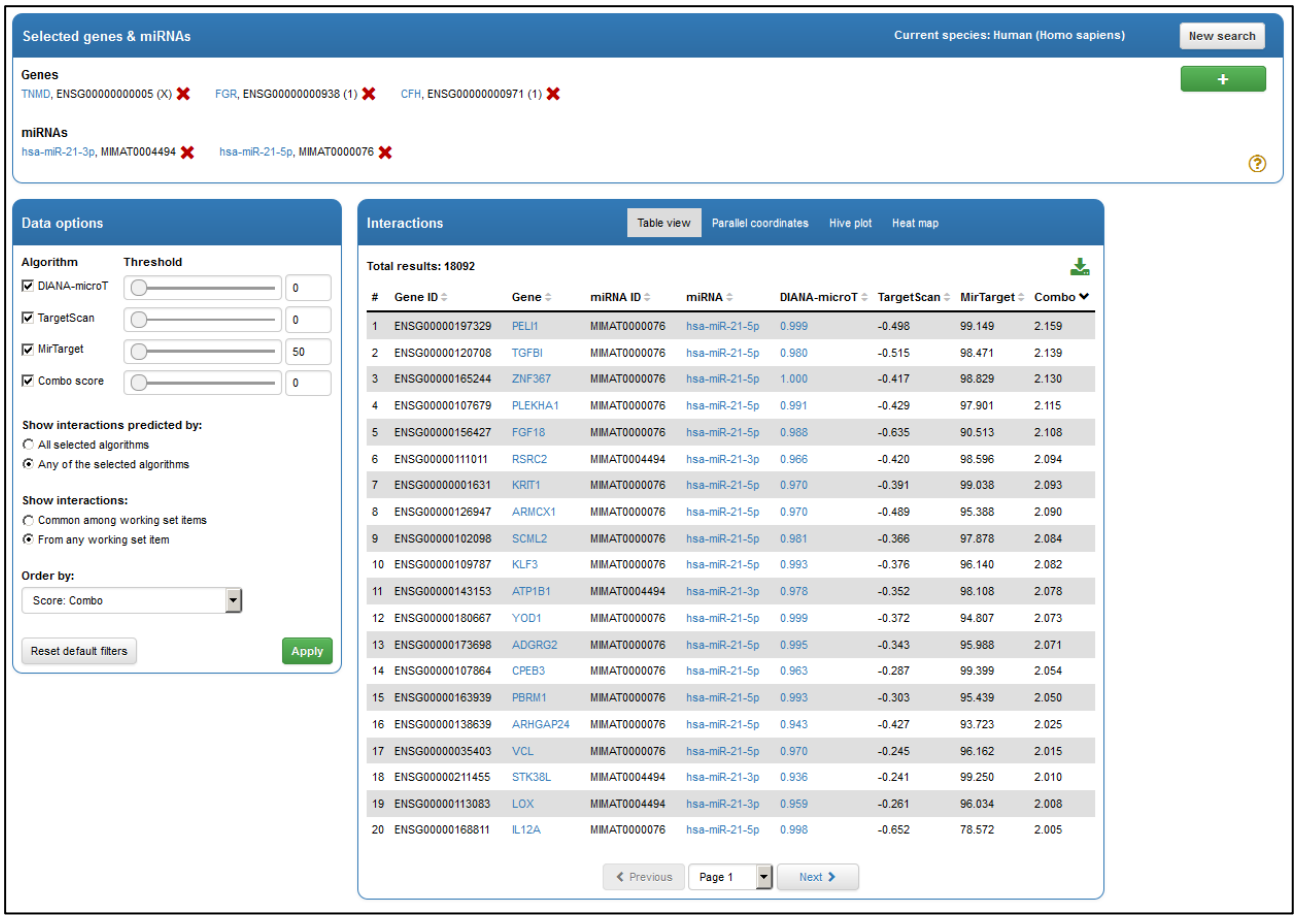
Εικόνα 10.4: επιτυχής προσθήκη αντικειμένου στη λίστα. Εμφανές το ενημερωτικό μήνυμα και το ανενεργό πλέον κουμπί της πρώτης επιλογής δεξιά.

Μόλις προσθέσουμε το πρώτο αντικείμενο στη λίστα εργασίας, παρατηρείστε ότι η επιλογή οργανισμού δεν είναι πλέον διαθέσιμη. Αυτό συμβαίνει διότι κάθε στιγμή μπορούμε να μελετάμε αλληλεπιδράσεις για έναν μόνο οργανισμό.

10.3 *Η κεντρική οθόνη της εφαρμογής*

Η κεντρική οθόνη της εφαρμογής αποτελείται από 4 μέρη:

1. Στο πάνω μέρος βρίσκεται η «λίστα εργασίας». Εκεί φαίνονται όλα τα επιλεγμένα αντικείμενα των οποίων τις αλληλεπιδράσεις πρόκειται να μελετήσουμε.
2. Στην αριστερή λωρίδα βρίσκεται το πλαίσιο "Data options" το οποίο περιέχει τα διαθέσιμα κριτήρια αναζήτησης και ταξινόμησης των αποτελεσμάτων.
3. Το κυρίως μέρος της οθόνης στο οποίο προβάλλονται τα αποτελέσματα. Η επιλογή της παρουσίασης μεταξύ του πίνακα και των γραφημάτων γίνεται με καρτέλες (tabs).



Εικόνα 10.5: η κεντρική οθόνη της εφαρμογής

10.3.1 *Επιλεγμένα αντικείμενα (λίστα εργασίας – working set)*

10.3.1.1 *Αφαίρεση αντικειμένου από τη λίστα*

Δίπλα σε κάθε αντικείμενο της λίστας εργασίας υπάρχει κουμπί "X" που αφαιρεί το εκάστοτε αντικείμενο από τη λίστα εργασίας. Όταν ένα αντικείμενο αφαιρεθεί, οι αλληλεπιδράσεις του δεν θα περιλαμβάνονται πλέον στα αποτελέσματα.

Αν αφαιρεθεί και το τελευταίο αντικείμενο από τη λίστα εργασίας, τότε θα επιστρέψουμε στην αρχική σελίδα της εφαρμογής και μπορούμε να αρχίσουμε τη διαδικασία από την αρχή.

10.3.1.2 Προσθήκη νέου αντικειμένου στη λίστα

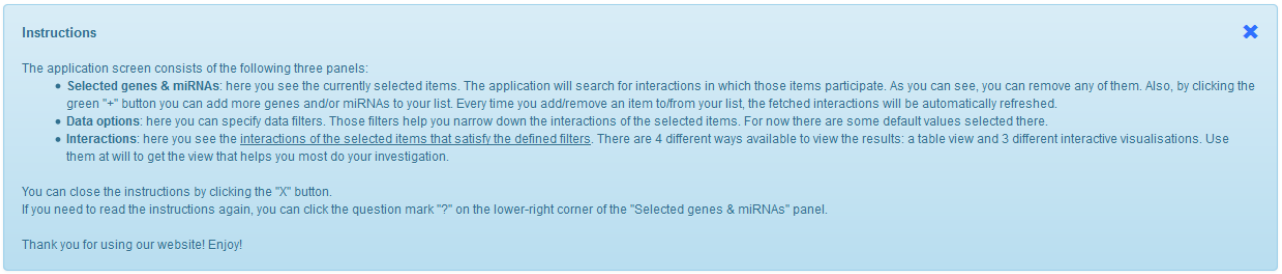
Το κουμπί "+" που βρίσκεται στο δεξί μέρος του πλαισίου μπορεί να χρησιμοποιηθεί για να προστεθούν επιπλέον αντικείμενα στη λίστα εργασίας. Η διαδικασία είναι επακριβώς η ίδια όπως περιγράφηκε στην ενότητα 10.2 με εξαίρεση την επιλογή οργανισμού (βήμα 2) που δεν θα είναι διαθέσιμη. Υπενθυμίζουμε πως αυτό συμβαίνει διότι κάθε στιγμή μπορούμε να μελετάμε αλληλεπιδράσεις μόνο από έναν οργανισμό.

10.3.1.3 Νέα αναζήτηση & αναζήτηση για διαφορετικό οργανισμό

Κάνοντας κλικ επάνω στο κουμπί "New Search" η λίστα εργασίας αρχικοποιείται (αδειάζει) και μπορούμε να ξεκινήσουμε μία νέα αναζήτηση από την αρχή. Η διαδικασία στη συνέχεια είναι επακριβώς η ίδια όπως περιγράφηκε στην ενότητα 10.2. Σε αυτή την περίπτωση η επιλογή οργανισμού είναι και πάλι διαθέσιμη.

10.3.1.4 Εμφάνιση οδηγιών

Κάνοντας κλικ επάνω στο κίτρινο ερωτηματικό "?" στην κάτω δεξιά γωνία του πλαισίου, εμφανίζεται ένα πλαίσιο με οδηγίες για τη χρήση της εφαρμογής.



Εικόνα 10.6: οι οδηγίες χρήσης της εφαρμογής όπως εμφανίζονται σε πλαίσιο

10.3.2 Επιλογές δεδομένων & φίλτρα (data options)

Στο πλαίσιο "Data options" βρίσκονται όλα τα διαθέσιμα κριτήρια αναζήτησης και ταξινόμησης των αποτελεσμάτων. Την πρώτη φορά που φτάνουμε στην κεντρική οθόνη της εφαρμογής είναι προεπιλεγμένες κάποιες αρχικές τιμές των κριτηρίων αναζήτησης έτσι ώστε να γίνεται ένας αρχικός περιορισμός των αποτελεσμάτων.

10.3.2.1 Επιλογή αλγορίθμων

Εδώ μπορούμε να επιλέξουμε από ποιους αλγορίθμους επιθυμούμε να προσκομίσουμε αποτελέσματα. Αυτή η επιλογή είναι σημαντική δεδομένου ότι δεν προβλέπονται όλες οι αλληλεπιδράσεις από όλους τους

αλγορίθμους. Επομένως, στα αποτελέσματα θα εμφανιστούν μόνο οι αλληλεπιδράσεις που προβλέπονται από τους επιλεγμένους αλγορίθμους. Κατά τον ίδιο τρόπο, στα αποτελέσματα θα εμφανιστούν βαθμολογίες μόνο για τους επιλεγμένους αλγορίθμους.

10.3.2.2 Βαθμολογία κατωφλίου ανά αλγόριθμο (thresholds)

Εδώ μπορούμε να επιλέξουμε μία βαθμολογία κατωφλίου (threshold) την οποία επιθυμούμε να επιτυγχάνουν οι αλληλεπιδράσεις. Έτσι στα αποτελέσματα θα εμφανιστούν μόνον εκείνες οι αλληλεπιδράσεις που επιτυγχάνουν το κατώφλι για τον εκάστοτε αλγόριθμο. Μπορούμε, φυσικά, να ορίσουμε κατώφλι για κάθε αλγόριθμο ξεχωριστά.

Η επιλογή γίνεται με δύο τρόπους: είτε με τον κυλιόμενο επιλογέα είτε με το πεδίο κειμένου.

Για να απενεργοποιήσουμε το κατώφλι για κάποιον αλγόριθμο αρκεί να μετακινήσουμε τον κυλιόμενο επιλογέα στο αριστερό άκρο του. Με άλλα λόγια, αυτό που κάνουμε είναι να θέσουμε ως κατώφλι την ελάχιστη βαθμολογία του εν λόγω αλγορίθμου, την οποία, όμως, εξ ορισμού την υπερβαίνουν όλες οι αλληλεπιδράσεις που προβλέπονται από τον αλγόριθμο αυτόν.

Προσοχή: **δεν** έχουν όλοι οι αλγόριθμοι ως ελάχιστη βαθμολογία το 0.

Παράδειγμα: για να απενεργοποιήσουμε τη βαθμολογία κατωφλίου για τον MirTarget, θέτουμε το κατώφλι του στο 50.

10.3.2.3 Συναλήθευση αλληλεπιδράσεων μεταξύ αλγορίθμων

Στο κριτήριο "Show interactions predicted by" οι διαθέσιμες επιλογές είναι οι εξής δύο:

- **All selected algorithms**: με αυτή την επιλογή εμφανίζονται στα αποτελέσματα αλληλεπιδράσεις που προβλέπονται από όλους τους επιλεγμένους αλγορίθμους (λογικό «ΚΑΙ» – τομή)
- **Any of the selected algorithms**: με αυτή την επιλογή εμφανίζονται στα αποτελέσματα αλληλεπιδράσεις που προβλέπονται από έστω έναν, οποιονδήποτε αλγόριθμο (λογικό «Η» – ένωση).

Επισημαίνεται ότι αυτές οι επιλογές λαμβάνουν υπ' όψιν τους και τα κατώφλια που έχουν, ενδεχομένως, ορισθεί.

Παράδειγμα: έστω ότι είναι επιλεγμένοι οι αλγόριθμοι DIANA-microT, TargetScan και Combo και έχουμε ορίσει κατώφλια 0, -0.5 και 0.75 αντιστοίχως. Τα κριτήρια που διαμορφώνονται, δηλαδή, είναι τα εξής:

1. Η αλληλεπίδραση προβλέπεται από τον DIANA-microT.
2. Η αλληλεπίδραση προβλέπεται από τον TargetScan με βαθμολογία τουλάχιστον -0.5.
3. Η αλληλεπίδραση έχει συνδυαστική βαθμολογία Combo τουλάχιστον 0.75.

Η επιλογή "All selected algorithms" θα εμφανίσει στα αποτελέσματα τις αλληλεπιδράσεις εκείνες που πληρούν **και** τα τρία κριτήρια. Η επιλογή "Any of the selected algorithms" θα εμφανίσει στα αποτελέσματα τις αλληλεπιδράσεις εκείνες που πληρούν **έστω ένα** από τα τρία κριτήρια.

10.3.2.4 Εντοπισμός κοινών αλληλεπιδράσεων μεταξύ αντικειμένων

Η λίστα εργασίας κάθε στιγμή μπορεί να περιέχει ένα αντικείμενο ή περισσότερα. Επίσης μπορεί να περιέχει μόνο γονίδια, μόνο miRNAs ή και τα δύο. Στο κριτήριο "Show interactions that are:" οι επιλογές είναι οι εξής:

- **From any working set item**: αυτή η επιλογή εμφανίζει στα αποτελέσματα όλες τις αλληλεπιδράσεις από όλα τα αντικείμενα της λίστας εργασίας.
- **Common among working set items**: αυτή η επιλογή εμφανίζει στα αποτελέσματα μόνο τις κοινές αλληλεπιδράσεις μεταξύ των αντικειμένων της λίστας εργασίας. Αν η λίστα εργασίας περιέχει:
    i. **κ γονίδια** τότε η εφαρμογή θα εντοπίσει αν υπάρχουν miRNAs (έστω ν σε πλήθος) που να αλληλεπιδρούν και με τα κ γονίδια. Έτσι, στα αποτελέσματα θα δούμε κ*ν αλληλεπιδράσεις και μάλιστα κάθε miRNA θα εμφανιστεί ακριβώς κ φορές, δηλαδή σε μία αλληλεπίδραση για καθένα από τα κ γονίδια. (*Σημαντική σημείωση*: αν θέσουμε ως επιπλέον κριτήριο κάποιο κατώφλι στις βαθμολογίες, ενδεχομένως να εμφανιστούν λιγότερες από κ*ν αλληλεπιδράσεις στα αποτελέσματα)
    ii. **κ miRNAs** τότε η εφαρμογή θα εντοπίσει αν υπάρχουν γονίδια που να αλληλεπιδρούν και με τα κ miRNAs. Η αντιμετώπιση είναι ακριβώς αντίστοιχη με το προηγούμενο.
    iii. **x γονίδια και y miRNAs**. Μεταξύ αυτών των αντικειμένων έχουμε x*y πιθανές αλληλεπιδράσεις. Στα αποτελέσματα θα εμφανιστούν μόνον όσες από αυτές τις x*y αλληλεπιδράσεις πράγματι υπάρχουν.

Για την καλύτερη επεξήγηση των περιπτώσεων (i) και (ii) παραθέτουμε δύο παραδείγματα.

Παράδειγμα 1: έστω ότι στη λίστα εργασίας περιέχονται τα γονίδια TNMD και DPM1. ΤοTNMD έχει συνολικά 694 αλληλεπιδράσεις και το DPM1 848 (κάθε αλληλεπίδραση είναι, προφανώς, με διαφορετικό miRNA). Η εφαρμογή σε αυτή την περίπτωση θα εντοπίσει ποια miRNAs αλληλεπιδρούν **και με τα δύο γονίδια** και θα βρει 294 miRNAs. Έτσι, στα αποτελέσματα θα εμφανιστούν 588 αλληλεπιδράσεις συνολικά (294 miRNAs * 2 γονίδια = 588 αλληλεπιδράσεις).

Υπενθυμίζουμε πως αν θέσουμε ως επιπλέον κριτήριο κάποιο κατώφλι στις βαθμολογίες, ενδεχομένως να εμφανιστούν λιγότερες από 588 αλληλεπιδράσεις στα αποτελέσματα.

Παράδειγμα 2: δεν υπάρχει κανένα miRNA που να αλληλεπιδρά και με το γονίδιο TNMD αλλά και με το FIGNL2. Έτσι, αν βάλουμε στη λίστα εργασίας αυτά τα δύο γονίδια και ενεργοποιήσουμε την επιλογή "Common among working set items" δεν θα δούμε κανένα αποτέλεσμα.

10.3.2.5 Ταξινόμηση αποτελεσμάτων

Η επιλογή αυτή καθορίζει με ποιο τρόπο θα ταξινομηθούν οι αλληλεπιδράσεις που θα εμφανιστούν στα αποτελέσματα. Οι πιθανές ταξινομήσεις είναι με βάση το όνομα (γονιδίου/miRNA), το μοναδικό αναγνωριστικό (γονιδίου/miRNA) ή τη βαθμολογία κάποιου εκ των αλγορίθμων.

Η ταξινόμηση των αλληλεπιδράσεων φαίνεται ξεκάθαρα όταν τα αποτελέσματα προβάλλονται σε μορφή πίνακα. Η επιλογή ταξινόμησης είναι ιδιαίτερα σημαντική διότι τα γραφήματα απεικονίζουν τα εκάστοτε κορυφαία αποτελέσματα με βάση την ταξινόμηση που έχει επιλεγεί.

10.3.2.6 Πλήκτρο "Apply"

Κάθε φορά, αφού προβούμε στις επιθυμητές αλλαγές στα κριτήρια αναζήτησης, πρέπει να πατήσουμε το πλήκτρο "Apply" έτσι ώστε να ενημερωθεί η τρέχουσα προβολή των αποτελεσμάτων (πίνακας ή κάποιο απ' τα γραφήματα). Προφανώς μπορούμε να αλλάξουμε όσα κριτήρια θέλουμε κάθε φορά.

10.3.2.7 Πλήκτρο "Reset default filters"

Πατώντας αυτό το πλήκτρο, τα κριτήρια αναζήτησης επανέρχονται στις προεπιλεγμένες τιμές και επιλογές.

10.4 Προβολή αποτελεσμάτων και γραφήματα

Για την παρουσίαση των αποτελεσμάτων υπάρχουν 4 διαφορετικές προβολές: η εμφάνιση σε πίνακα και 3 είδη γραφημάτων.

10.4.1 Προβολή πίνακα

Η κλασσική προβολή σε πίνακα είναι η προεπιλεγμένη μορφή παρουσίασης των αλληλεπιδράσεων. Ο πίνακας υπόκειται σε σελιδοποίηση και κάθε σελίδα του περιέχει 20 αλληλεπιδράσεις. Κάθε γραμμή του πίνακα αντιστοιχεί σε μια αλληλεπίδραση και περιέχει τις εξής στήλες:

- όνομα γονιδίου
- μοναδικό αναγνωριστικό γονιδίου κατά Ensembl (Ensembl ID)
- όνομα miRNA
- μοναδικό αναγνωριστικό miRNA κατά miRBase (miRBase accession number). Για συντομία στον πίνακα αναφέρεται ως "miRNA ID".
- βαθμολογία της αλληλεπίδρασης για κάθε επιλεγμένο αλγόριθμο (1 στήλη / αλγόριθμο)

Ο πίνακας είναι διαδραστικός και παρέχει τις εξής δυνατότητες:

- Πλήκτρα "Previous" και "Next" για περιήγηση στις σελίδες των αποτελεσμάτων.
- Αναδυόμενη λίστα όλων των σελίδων των αποτελεσμάτων για δυνατότητα άμεσης μετάβασης σε συγκεκριμένη σελίδα.
- Κάνοντας κλικ επάνω στον τίτλο οποιασδήποτε στήλης του πίνακα, τα αποτελέσματα ταξινομούνται με βάση αυτή τη στήλη.
- Το όνομα του γονιδίου και του miRNA αποτελούν υπερσυνδέσμους προς τη σελίδα λεπτομερειών του εκάστοτε αντικειμένου.

- Η βαθμολογία του DIANA-microT (όπου υπάρχει) είναι υπερσύνδεσμος που ανακατευθύνει το χρήστη προς την ιστοσελίδα του αλγορίθμου αυτού, με προβολή αναλυτικών λεπτομερειών για τη συγκεκριμένη αλληλεπίδραση.

Σημείωση: δυστυχώς για τους αλγορίθμους TargetScan και MirTarget δεν κατέστη δυνατό να δημιουργήσουμε υπερσυνδέσμους προς τα αντίστοιχα αποτελέσματα καθώς οι ιστοσελίδες τους δεν παρέχουν τρόπο να προβάλλουμε προγραμματιστικά μία συγκεκριμένη αλληλεπίδραση, δυνατότητα που παρέχει ο DIANA–microT.

10.4.2 Κατέβασμα αποτελεσμάτων (download) σε μορφή πίνακα

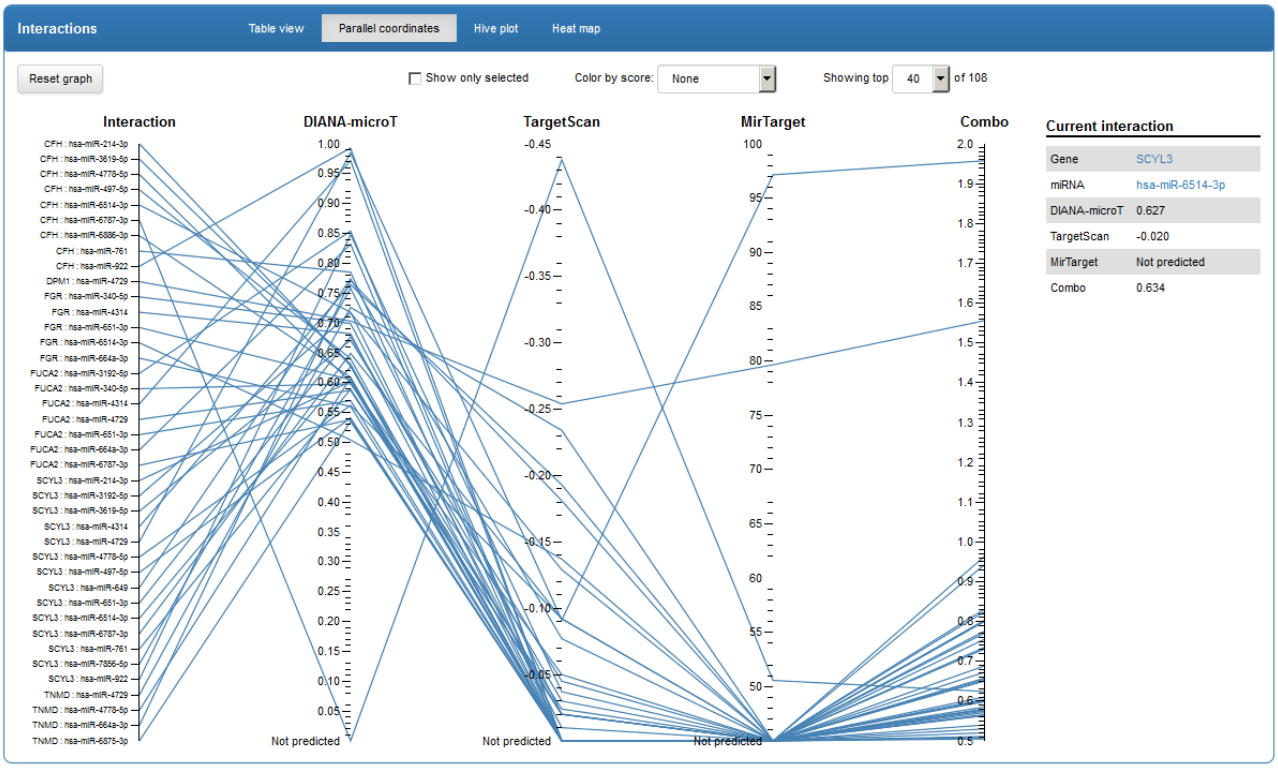
Στην άνω δεξιά γωνία της προβολής πίνακα υπάρχει κουμπί για το κατέβασμα των αποτελεσμάτων. Πατώντας το κουμπί αυτό ο χρήστης μπορεί να κατεβάσει τα τρέχοντα αποτελέσματα σε μορφή πίνακα. Το αρχείο που δημιουργείται είναι μορφής TSV (tab separated values). Η δομή των αποτελεσμάτων μέσα στο αρχείο είναι ακριβώς η ίδια όπως παρουσιάζονται και στην προβολή πίνακα.

10.4.3 Γράφημα: Parallel coordinates

10.4.3.1 Περιγραφή του γραφήματος

Το γράφημα αποτελείται από 2 μέχρι 5 κάθετους, παράλληλους άξονες, ανάλογα με το πόσοι αλγόριθμοι είναι επιλεγμένοι.

- Ο πρώτος άξονας αριστερά (“Interaction”) αναγράφει με αλφαβητική σειρά τις αλληλεπιδράσεις που προβάλλονται στο γράφημα. Η κάθε αλληλεπίδραση αναγράφεται ως «γονίδιο : miRNA».
- Καθένας από τους επόμενους άξονες αντιπροσωπεύει έναν αλγόριθμο πρόβλεψης. Στο γράφημα εμφανίζονται μόνο οι επιλεγμένοι αλγόριθμοι. Οι άξονες των αλγορίθμων δεν αναγράφουν την πλήρη βαθμολογική κλίμακα του εκάστοτε αλγορίθμου. Αντίθετα, ο κάθε άξονας εστιάζει γύρω από τις τιμές στις οποίες κυμαίνονται οι προβαλλόμενες αλληλεπιδράσεις, ούτως ώστε να επιτυγχάνεται η καλύτερη δυνατή ευκρίνεια στο γράφημα.
- Κάθε γραμμή αντιπροσωπεύει μια αλληλεπίδραση. Οι γραμμές ξεκινούν από τον άξονα των ονομάτων και διασχίζουν όλο το γράφημα, τέμνοντας κάθε άξονα. Το σημείο τομής μιας γραμμής με έναν άξονα είναι το σημείο που αντιστοιχεί στη βαθμολογία της αλληλεπίδρασης αυτής για αυτό τον αλγόριθμο.
- Όταν υπάρχουν αλληλεπιδράσεις που δεν προβλέπονται από κάποιον αλγόριθμο, τότε στη βάση του αντίστοιχου άξονα εμφανίζεται η τιμή “Not predicted”. Όσες αλληλεπιδράσεις δεν προβλέπονται από αυτόν τον αλγόριθμο, θα τέμνουν τον αντίστοιχο άξονα στο σημείο “Not predicted”.



Εικόνα 10.7: το γράφημα Parallel coordinates

10.4.3.2 Δυνατότητες του γραφήματος

Το γράφημα είναι διαδραστικό και παρέχει μία σειρά από δυνατότητες στο χρήστη έτσι ώστε να του επιτρέπει να προσαρμόζει την εμφάνιση του γραφήματος ανάλογα με την ανάλυση που επιθυμεί να διεξάγει.

- Περνώντας το ποντίκι (mouseover) πάνω από τα ονόματα των αλληλεπιδράσεων («ετικέτες») ή πάνω από μια γραμμή (σε οποιοδήποτε σημείο της γραμμής), τονίζεται η αντίστοιχη γραμμή και ετικέτα στο γράφημα ενώ όλες οι υπόλοιπες γραμμές και ετικέτες πηγαίνουν στο παρασκήνιο. Στον πίνακα "Current interaction" εμφανίζονται τα στοιχεία αυτής της αλληλεπίδρασης.
- Κάνοντας κλικ επάνω στην ετικέτα ή τη γραμμή μιας αλληλεπίδρασης, μπορούμε να επιλέξουμε την αλληλεπίδραση αυτή. Τότε, για να είναι αυτή η αλληλεπίδραση συνεχώς διακριτή, η γραμμή αποκτά μεγαλύτερο πάχος και χρωματίζεται με ένα τυχαίο χρώμα. Η αντίστοιχη ετικέτα λαμβάνει το ίδιο χρώμα με τη γραμμή και γίνεται έντονη (bold).

  Επίσης, τα δεδομένα της αλληλεπίδρασης αυτής παραμένουν σταθερά στον πίνακα "Current interaction" για 3 δευτερόλεπτα έτσι ώστε να μπορέσουμε να «βγούμε» από το χώρο του γραφήματος (αν θέλουμε να πάμε σε κάποιο άλλο σημείο της σελίδας) χωρίς να επηρεαστούν τα δεδομένα του πίνακα καθώς περνάμε πάνω από άλλες γραμμές.
- Για να αποεπιλέξουμε μία γραμμή αρκεί να ξανακάνουμε κλικ επάνω της ή επάνω στην ετικέτα της.
- Όσες γραμμές είναι επιλεγμένες παραμένουν εμφανείς και τονισμένες και δεν πηγαίνουν στο παρασκήνιο όταν περνάμε πάνω από μια άλλη γραμμή/ετικέτα.
- Όταν βρισκόμαστε πάνω σε έναν άξονα μπορούμε να κάνουμε «κλικ και σύρσιμο» (click and drag) ώστε να ορίσουμε ένα εύρος επάνω στον άξονα αυτό. Τότε όσες γραμμές δεν περνάνε μέσα από το

εύρος που ορίσαμε απενεργοποιούνται και δεν είναι πλέον ορατές, ούτε καν στο παρασκήνιο. Έτσι μπορούμε να περιορίσουμε την ανάλυσή μας σε συγκριμένες αλληλεπιδράσεις.

- Αφότου έχουμε ορίσει ένα εύρος, μπορούμε κατόπιν να μετακινήσουμε το εύρος επάνω στον άξονα με «σύρσιμο και εναπόθεση» (drag and drop). Για να ακυρώσουμε ένα εύρος αρκεί να κάνουμε απλό κλικ επάνω σε οποιοδήποτε σημείο του άξονα έξω από το εύρος.
- Μπορούμε να ορίσουμε ταυτόχρονα εύρη επάνω σε διαφορετικούς άξονες. Όμως, επάνω σε ένα συγκεκριμένο άξονα, μπορεί να ορισθεί μόνο ένα εύρος κάθε στιγμή. Επίσης, δε γίνεται να ορίσουμε εύρος επάνω στον άξονα των ονομάτων.
- Μπορούμε να κάνουμε «σύρσιμο και εναπόθεση» (drag and drop) επάνω στα ονόματα των αξόνων και, έτσι, να τους αναδιατάξουμε. Αυτό μας βοηθά να συγκρίνουμε άμεσα δύο συγκεκριμένους αλγόριθμους φέρνοντάς τους δίπλα-δίπλα. Δεν μπορούμε να αλλάξουμε θέση στον άξονα των ονομάτων ο οποίος παραμένει πάντα αριστερότερα από τους υπόλοιπους άξονες.

Οι πληροφορίες του πίνακα "Current interaction" είναι οι εξής:

- Αναγράφονται το γονίδιο και το miRNA της τρέχουσας αλληλεπίδρασης. Τα ονόματα γονιδίου και miRNA είναι υπερσύνδεσμοι που οδηγούν στην αντίστοιχη σελίδα λεπτομερειών.
- Αναγράφονται οι επιμέρους βαθμολογίες της αλληλεπίδρασης για τον κάθε αλγόριθμο. Αν κάποιος αλγόριθμος δεν έχει επιλεγεί από το χρήστη, θα εμφανιστεί μία παύλα «–» στη θέση της βαθμολογίας του. Με άλλα λόγια, η παύλα σημαίνει πως τα δεδομένα δεν είναι διαθέσιμα οπότε δεν είναι γνωστό αν ο αλγόριθμος προβλέπει ή όχι την εν λόγω αλληλεπίδραση. Αν τα δεδομένα είναι διαθέσιμα και η εν λόγω αλληλεπίδραση πράγματι δεν προβλέπεται από τον αλγόριθμο αυτό, στη θέση της βαθμολογίας θα εμφανιστεί "Not predicted".

Η γραμμή εργαλείων πάνω από το γράφημα παρέχει τις εξής επιπλέον επιλογές:

- "Reset graph": επαναφέρει το γράφημα στην αρχική του εμφάνιση (χωρίς επιλεγμένες γραμμές, απενεργοποιημένο "Show only selected" και "Color by score: None").
- "Show only selected": αφήνει εμφανείς στο γράφημα μόνο τυχόν επιλεγμένες γραμμές και στέλνει όλες τις μη επιλεγμένες γραμμές στο παρασκήνιο. Μπορούμε να επιλέξουμε αυτή την επιλογή ακόμη κι αν δεν έχουμε καμία γραμμή επιλεγμένη (οπότε όλες οι γραμμές θα βρεθούν στο παρασκήνιο).
- "Color by": χρωματισμός των γραμμών ανάλογα με τη βαθμολογία των αλληλεπιδράσεων στον συγκεκριμένο αλγόριθμο. Ο χρωματισμός γίνεται από κόκκινο μέχρι μπλε όπου κόκκινο σημαίνει «καλύτερη βαθμολογία» και μπλε «χειρότερη βαθμολογία». Οι ενδιάμεσες βαθμολογίες λαμβάνουν ενδιάμεσες χρωματικές διαβαθμίσεις. Αν υπάρχουν τυχόν τονισμένες γραμμές, τότε αυτές διατηρούνται όταν αλλάζουμε επιλογή στο "Color by score".

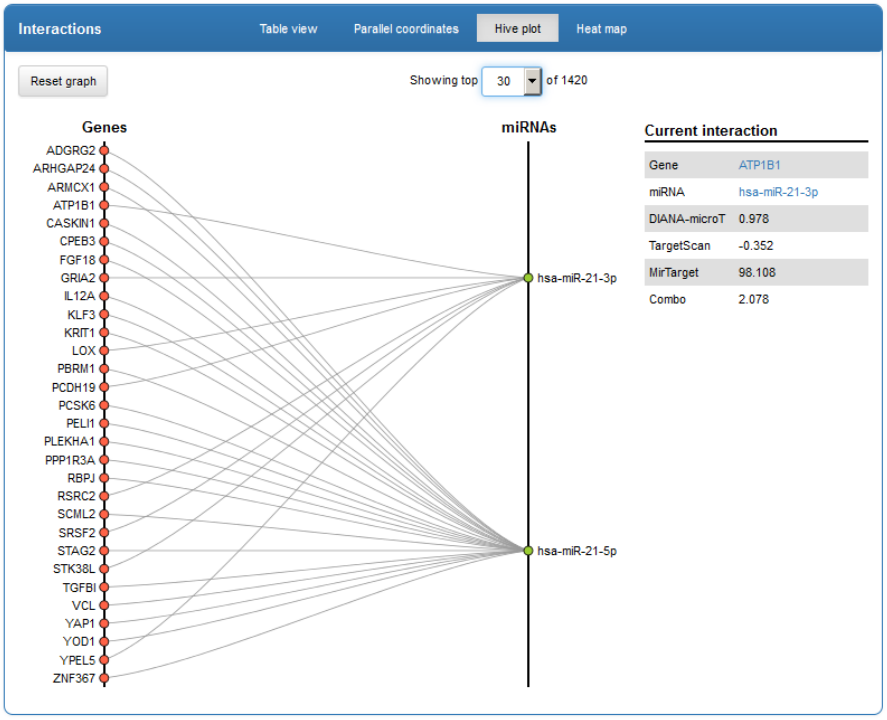
  Παράδειγμα: μια κόκκινη γραμμή θα τέμνει τον αντίστοιχο άξονα σε υψηλότερο σημείο από μια μπλε γραμμή.
- "Show top X of Y": αναδυόμενη λίστα με επιλογή του πλήθους των αλληλεπιδράσεων που θέλουμε να προβάλουμε στο γράφημα. Τα αποτελέσματα που εμφανίζονται είναι τα X κορυφαία

αποτελέσματα, βάσει του τρόπου εμφάνισής τους στον πίνακα. Ο αριθμός Υ αντιπροσωπεύει το σύνολο των αλληλεπιδράσεων που έχουν βρεθεί.

10.4.4 Γράφημα: Hive plot

10.4.4.1 Περιγραφή του γραφήματος

Το γράφημα αποτελείται από δύο άξονες, έναν για τα γονίδια και έναν για τα miRNAs. Επάνω στους άξονες είναι διατεταγμένοι αλφαβητικά οι κόμβοι που αντιπροσωπεύουν τα αντικείμενα της κάθε κατηγορίας. Η κάθε ακμή αντιπροσωπεύει μία αλληλεπίδραση μεταξύ ενός γονιδίου και ενός miRNA.



Εικόνα 10.8: το γράφημα Hive plot

10.4.4.2 Δυνατότητες του γραφήματος

Το γράφημα είναι διαδραστικό και παρέχεις τις εξής δυνατότητες:

- Όταν περνάμε με το ποντίκι πάνω από μία ακμή, η ακμή αυτή τονίζεται. Στον πίνακα "Current interaction" εμφανίζονται τα στοιχεία αυτής της αλληλεπίδρασης.
- Όταν περνάμε πάνω από έναν κόμβο (ή την ετικέτα του) τότε τονίζονται όλες οι ακμές που ακουμπούν σε αυτόν τον κόμβο.
- Κάνοντας κλικ επάνω σε μία ακμή, τότε αυτή αποκτά μεγαλύτερο πάχος και χρωματίζεται με ένα τυχαίο χρώμα. Επίσης, τα δεδομένα της ακμής αυτής παραμένουν σταθερά στον πίνακα "Current interaction" για 3 δευτερόλεπτα έτσι ώστε να μπορέσουμε να «βγούμε» από το χώρο του γραφήματος (αν θέλουμε να πάμε σε κάποιο άλλο σημείο της σελίδας) χωρίς να επηρεαστούν τα δεδομένα του πίνακα καθώς περνάμε πάνω από άλλες ακμές/κόμβους.

- Κάνοντας κλικ επάνω σε έναν κόμβο τονίζονται όλες οι ακμές που ακουμπούν σε αυτόν.
- Για να αποεπιλέξουμε μία ακμή αρκεί να ξανακάνουμε κλικ επάνω της. Για να αποεπιλέξουμε όλες τις ακμές που ακουμπούν σε έναν κόμβο, αρκεί να ξανακάνουμε κλικ επάνω στον κόμβο. Αν έχουμε επιλέξει όλες τις ακμές ενός κόμβου, είναι δυνατόν να αποεπιλέξουμε μεμονωμένα κάποιες από αυτές.
- Όσες ακμές είναι επιλεγμένες παραμένουν τονισμένες όταν περνάμε πάνω από μια άλλη ακμή ή κόμβο.

Η μπάρα εργαλείων πάνω από το γράφημα παρέχει τις εξής επιπλέον επιλογές:

- **Reset graph**: επαναφέρει το γράφημα στην αρχική του εμφάνιση, χωρίς καμία επιλεγμένη ακμή.
- **Show top X of Y**: αναδυόμενη λίστα με επιλογή του πλήθους των αλληλεπιδράσεων που θέλουμε να προβάλουμε στο γράφημα. Τα αποτελέσματα που εμφανίζονται είναι τα X κορυφαία αποτελέσματα, βάσει του τρόπου εμφάνισής τους στον πίνακα. Ο αριθμός Y αντιπροσωπεύει το σύνολο των αλληλεπιδράσεων που έχουν βρεθεί.

Οι πληροφορίες του πίνακα "Current interaction" είναι οι ίδιες όπως και στο γράφημα Parallel coordinates.

10.4.5 Γράφημα: Heat map

10.4.5.1 Περιγραφή του γραφήματος

Σε κάθε χάρτη τα γονίδια και τα miRNAs κατανέμονται στον οριζόντιο και τον κάθετο άξονα. Τα αντικείμενα που είναι περισσότερα θα τοποθετηθούν στον οριζόντιο άξονα για να γίνει καλύτερη αξιοποίηση του πλάτους της οθόνης. Ο κεντρικός χώρος του χάρτη χωρίζεται σε γραμμές και στήλες σύμφωνα με τα αντικείμενα των δύο αξόνων. Έτσι δημιουργείται ένα πλέγμα κελιών όπου κάθε κελί ανήκει σε μία γραμμή και μία στήλη. Επομένως, κάθε κελί αντιστοιχεί σε ένα ζεύγος γονιδίου–miRNA για το οποίο μπορεί να υπάρχει ή όχι η αντίστοιχη αλληλεπίδραση.
Τα κελιά χρωματίζονται με βάση τη βαθμολογία των αλληλεπιδράσεων από κόκκινο (καλή βαθμολογία) μέχρι μπλε (κακή βαθμολογία). Ένα κελί χρωματίζεται γκρι στις εξής περιπτώσεις:

- Αν αντιστοιχεί σε αλληλεπίδραση που δεν προβλέπεται από τον αλγόριθμο με βάση τον οποίο έχει χρωματιστεί το πλέγμα. Προσοχή: η αλληλεπίδραση μπορεί, όμως, να προβλέπεται από κάποιον άλλο αλγόριθμο.
- Αν η αλληλεπίδραση αυτή δεν βρίσκεται μέσα στα κορυφαία X αποτελέσματα που προβάλλονται στο γράφημα αυτή τη στιγμή.

10.4.5.2 Δυνατότητες του γραφήματος

Το γράφημα είναι διαδραστικό και παρέχει τις εξής δυνατότητες:

- Αρχικά υπάρχει η επιλογή για το πόσους ξεχωριστούς χάρτες επιθυμούμε να βλέπουμε ταυτόχρονα, από έναν μέχρι τρεις. Ο πρώτος χάρτης είναι πάντα ορατός.

- Η επιλογή χρωματισμού (Color by) πάνω από τον κάθε χάρτη, επιλέγει τον αλγόριθμο με βάση τον οποίο θα χρωματιστούν τα κελιά του χάρτη αυτού. Ο κάθε χάρτης μπορεί να χρωματίζεται με βάση διαφορετικό αλγόριθμο γι' αυτό και υπάρχει ανεξάρτητη επιλογή χρωματισμού σε κάθε χάρτη. Τα κελιά χρωματίζονται ανάλογα με τη βαθμολογία που επιτυγχάνουν στον επιλεγμένο αλγόριθμο. Η χειρότερη δυνατή βαθμολογία του αλγορίθμου χρωματίζεται μπλε, η καλύτερη δυνατή βαθμολογία χρωματίζεται κόκκινο και οι ενδιάμεσες βαθμολογίες λαμβάνουν ανάλογες, ενδιάμεσες χρωματικές διαβαθμίσεις.
- Περνώντας με το ποντίκι πάνω από ένα κελί, το κελί αυτό μαζί με τα αντίστοιχά του στους άλλους χάρτες τονίζονται ενώ τα υπόλοιπα κελιά πηγαίνουν στο παρασκήνιο. Επίσης, στον πίνακα "Current interaction" εμφανίζονται τα στοιχεία αυτού του κελιού.
- Κάνοντας κλικ επάνω σε ένα κελί, τότε το κελί αυτό μαζί με τα αντίστοιχά του στους άλλους χάρτες παραμένουν τονισμένα ενώ τα υπόλοιπα πηγαίνουν στο παρασκήνιο για 3 δευτερόλεπτα. Για αυτό το χρονικό διάστημα, τα δεδομένα του κελιού παραμένουν σταθερά στον πίνακα "Current interaction".

Εικόνα 10.9: το γράφημα Heat map με δύο ορατούς χάρτες και διαφορετική επιλογή χρωματισμού σε κάθε χάρτη

Η μπάρα εργαλείων πάνω από τον πρώτο χάρτη, στο δεξί μέρος της παρέχει ορισμένες επιπλέον επιλογές οι οποίες επηρεάζουν από κοινού όλους τους χάρτες του γραφήματος. Αυτές οι επιλογές είναι:

- **Order axis by**: ορίζει την ταξινόμηση των αντικειμένων στους άξονες. Η μία επιλογή είναι η αλφαβητική ταξινόμηση (Name). Οι άλλες επιλογές ταξινομούν με βάση τις βαθμολογίες των αλγορίθμων.

  Παράδειγμα: έστω ότι επιλέγουμε ταξινόμηση με βάση το Combo score. Τότε, για κάθε γονίδιο θα

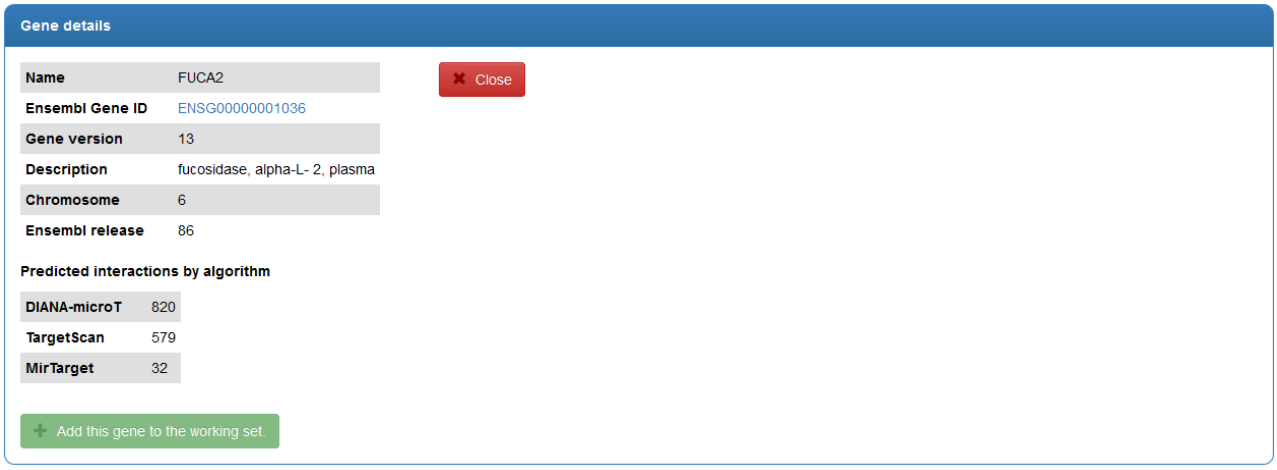
υπολογιστεί η μέση βαθμολογία Combo score που λαμβάνουν όλα τα κελιά του γονιδίου αυτού. Κατόπιν τα γονίδια θα ταξινομηθούν στον άξονά τους με βάση τη μέση βαθμολογία που μόλις υπολογίστηκε.

Η ταξινόμηση με βάση κάποιον απ' τους αλγορίθμους έχει ως αποτέλεσμα οι βαθμολογίες να ελαττώνονται «κατά μέσο όρο» όσο πηγαίνουμε από πάνω αριστερά προς κάτω δεξιά του γραφήματος, ασχέτως με το ποιος άξονας περιέχει τα γονίδια και ποιος τα miRNA. Η διατύπωση «κατά μέσο όρο» γίνεται αντιληπτή ως εξής: όσο πηγαίνουμε προς τα κάτω δεξιά τα περισσότερα κελιά αποκτούν όλο και «χειρότερο» χρωματισμό ενώ, αν υπάρχουν γκρι κελιά, θα αυξάνεται η συγκέντρωσή τους προς κάτω δεξιά. Αντίθετα, όσο πηγαίνουμε προς τα πάνω αριστερά, τα περισσότερα κελιά αποκτούν «καλύτερο» χρωματισμό. Αυτό, βεβαίως, δεν αποκλείει στις παραπάνω περιπτώσεις μεμονωμένα κελιά με καλό ή κακό χρωματισμό, αντίστοιχα.

- **Enable highlighting**: ενεργοποιεί/απενεργοποιεί τον τονισμό των κελιών όταν βρισκόμαστε πάνω από αυτά. Ασχέτως με την επιλογή αυτή, οι πληροφορίες του πίνακα "Current interaction" ενημερώνονται πάντα καθώς και η καθυστέρηση των 5 δευτερολέπτων μετά από ένα κλικ ισχύει πάντα.
- **Show top X of Y**: αναδυόμενη λίστα με επιλογή του πλήθους των αλληλεπιδράσεων που θέλουμε να προβάλουμε στο γράφημα. Τα αποτελέσματα που εμφανίζονται είναι τα X κορυφαία αποτελέσματα, βάσει του τρόπου εμφάνισής τους στον πίνακα. Ο αριθμός Y αντιπροσωπεύει το σύνολο των αλληλεπιδράσεων που έχουν βρεθεί.

Οι πληροφορίες του πίνακα "Current interaction" είναι οι ίδιες όπως και στα άλλα γραφήματα.

10.5 Σελίδα λεπτομερειών γονιδίων/miRNAs



Εικόνα 10.10: η σελίδα λεπτομερειών ενός γονιδίου

Σε οποιοδήποτε σημείο της εφαρμογής συναντάται ένα όνομα γονιδίου ή miRNA, το όνομα αποτελεί υπερσύνδεσμο που ανακατευθύνει το χρήστη προς μία σελίδα λεπτομερειών αφιερωμένη στο εκάστοτε αντικείμενο. Η σελίδα λεπτομερειών ανοίγει σε ξεχωριστή καρτέλα του φυλλομετρητή έτσι ώστε να μην επηρεάζεται η περιήγηση του χρήστη – π.χ. η προβολή πίνακα ή το γράφημα που βλέπει ο χρήστης παραμείνει ανέπαφο στην προηγούμενη καρτέλα.

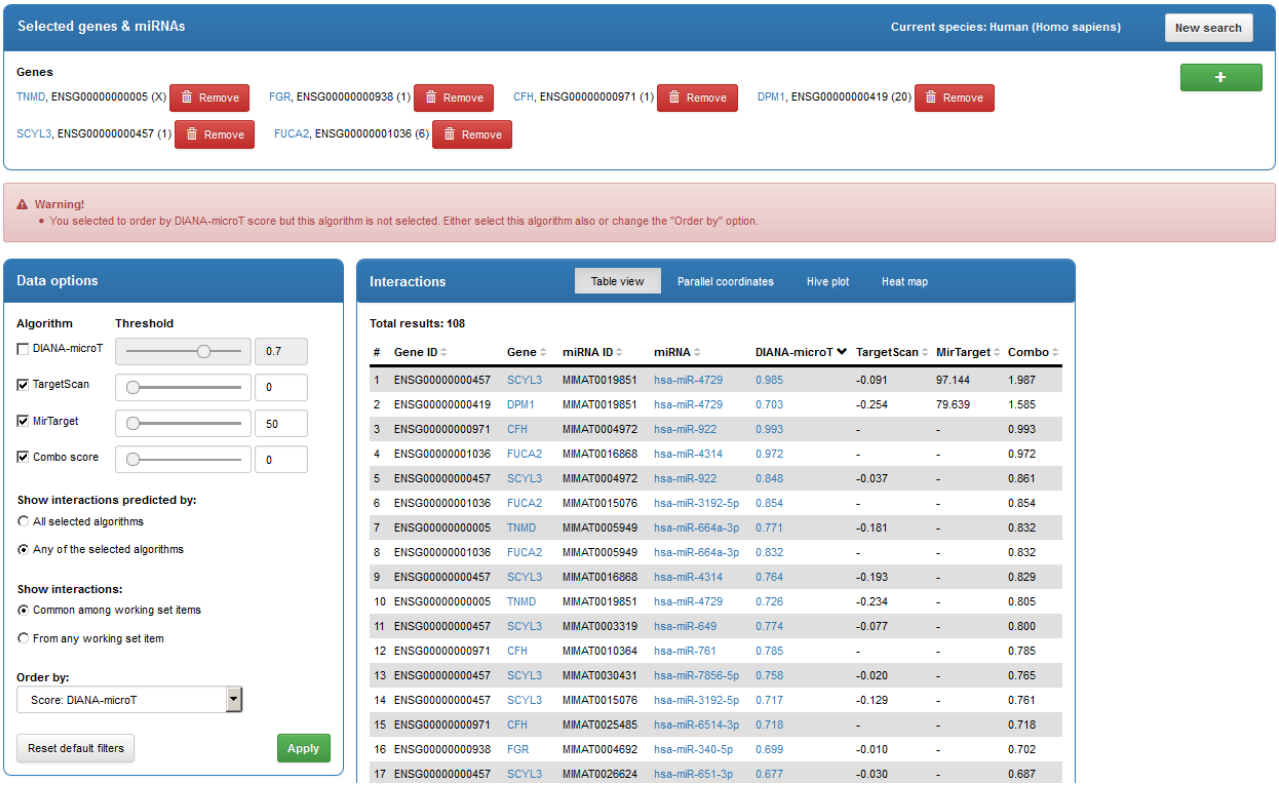
Τα στοιχεία που περιέχονται στη σελίδα λεπτομερειών είναι:

- Όνομα αντικειμένου και μοναδικό αναγνωριστικό. Το μοναδικό αναγνωριστικό είναι υπερσύνδεσμος προς την εξωτερική βάση δεδομένων που αποτελεί πηγή πληροφοριών για το αντικείμενο αυτό, δηλαδή προς την Ensembl για τα γονίδια και προς τη miRBase για τα miRNAs.
- Ειδικά στοιχεία γονιδίων: η έκδοση του γονιδίου (gene version), η αναλυτική περιγραφή του, το χρωμόσωμα στο οποίο βρίσκεται καθώς και η πιο πρόσφατη έκδοση της Ensembl στην οποία συναντάται το γονίδιο.
- Ειδικά στοιχεία miRNAs: η ακολουθία του και το μήκος της.
- Πόσες αλληλεπιδράσεις προβλέπει ο κάθε αλγόριθμος για το εν λόγω αντικείμενο.
- Πλήκτρο για την προσθήκη του αντικειμένου στη λίστα εργασίας. Αν χρησιμοποιηθεί το πλήκτρο αυτό συνιστάται να κλείσουμε την προηγούμενη καρτέλα, δηλαδή την καρτέλα στην οποία βρισκόμασταν πριν ανοίξουμε τη σελίδα λεπτομερειών.

Σημείωση για τα γονίδια: αν η έκδοση Ensembl που αναγράφεται στα στοιχεία του γονιδίου είναι παλαιότερη από την τρέχουσα έκδοση της Ensembl, τότε αυτό σημαίνει πως το εν λόγω γονίδιο έχει πλέον καταργηθεί και δεν περιλαμβάνεται στην τρέχουσα έκδοση. Σε αυτές τις περιπτώσεις ο υπερσύνδεσμος προς την Ensembl είναι προς την αρχειοθετημένη έκδοση που περιέχει το γονίδιο (και όχι προς την τρέχουσα). Σε αυτές τις περιπτώσεις εμφανίζεται και κατάλληλο ενημερωτικό μήνυμα.

10.6 Ανάνηψη από σφάλμα

10.6.1 Ενημερωτικά μηνύματα



Εικόνα 10.11: παράδειγμα ενημερωτικού μηνύματος για ασυνεπείς επιλογές στα φίλτρα. Το πλαίσιο "data options" είναι ορατό και φαίνονται οι επιλογές.

Σε περιπτώσεις κάποιας απλής εσφαλμένης επιλογής του χρήστη (π.χ. ασυνεπείς επιλογές στα φίλτρα) τότε η εφαρμογή συνεχίζει να εκτελείται κανονικά. Σε αυτές τις περιπτώσεις απλώς εμφανίζονται κατάλληλα ενημερωτικά μηνύματα προς το χρήστη είτε σε πλαίσιο στο άνω μέρος της σελίδας είτε σε αναδυόμενο παράθυρο, ανάλογα με την περίπτωση. Τα μηνύματα αυτά ενημερώνουν το χρήστη πώς μπορεί να διορθώσει τις επιλογές του και να συνεχίσει απρόσκοπτα τη χρήση της εφαρμογής. Σημειώνεται, όμως, πως αυτά είναι απλώς ενημερωτικά μηνύματα και δεν υποδεικνύουν σφάλμα στην εκτέλεση της εφαρμογής.

10.6.2 Σελίδα σφάλματος

Σε περίπτωση που η εφαρμογή βρεθεί σε ασυνεπή κατάσταση, ο χρήστης θα δει σελίδα σφάλματος. Τονίζουμε πως η οθόνη της εφαρμογής θα αντικατασταθεί εξ ολοκλήρου από μία σελίδα σφάλματος. Δεν πρόκειται, δηλαδή, για απλό ενημερωτικό μήνυμα. Σε αυτή την περίπτωση ο χρήστης πρέπει να πληκτρολογήσει τη διεύθυνση της αρχικής σελίδας της εφαρμογής. Έτσι θα επανέλθει στην προβολή πίνακα και μπορεί να συνεχίσει να χρησιμοποιεί την εφαρμογή κανονικά. Τόσο η λίστα εργασίας όσο και οι επιλογές των φίλτρων που έχει κάνει ο χρήστης θα διατηρηθούν ανεπηρέαστες.

Στη σπάνια περίπτωση που η αρχική σελίδα δεν εμφανιστεί αλλά προκύψει εκ νέου σελίδα σφάλματος ή αν

η σελίδα σφάλματος προκύπτει επαναλαμβανόμενα μετά από κάποια συγκεκριμένη ενέργεια του χρήστη, τότε η προτεινόμενη ενέργεια είναι η εξής: ο χρήστης πρέπει να επισκεφθεί τη σελίδα http://snf-723479.vm.okeanos.grnet.gr/ws/clear. Αυτή η ενέργεια θα διαγράψει τη λίστα εργασίας και θα αρχικοποιήσει την κατάσταση της εφαρμογής. Κατόπιν ο χρήστης θα ανακατευθυνθεί στην αρχική σελίδα της εφαρμογής η οποία, βέβαια, θα είναι πλέον κενή και θα μπορεί να ξεκινήσει μια νέα αναζήτηση.
Πιθανές περιπτώσεις που μπορεί να προκαλέσουν αυτό το ακραίο σφάλμα είναι:

- να συμβεί κάποιο εσωτερικό σφάλμα στον εξυπηρετητή,
- να αποσταλούν αιτήματα HTTP προς την εφαρμογή που σκοπίμως δεν είναι ορθώς διαμορφωμένα,
- να χρησιμοποιηθούν τα πλήκτρα "Back/Forward" του φυλλομετρητή,
- να προσπαθήσει ο χρήστης να προσπελάσει ιστοσελίδες της εφαρμογής πληκτρολογώντας διεύθυνση αντί να χρησιμοποιήσει τη διεπαφή της εφαρμογής.

10.6.3 Πιθανότητα σοβαρού σφάλματος

Σημειώνεται, ωστόσο, πως οι δοκιμές που πραγματοποιήθηκαν για την επαλήθευση της λειτουργίας της εφαρμογής και την αποσφαλμάτωσή της ήταν εξαντλητικές. Η εφαρμογή δεν παρουσίασε σφάλμα σε καμία περίπτωση ορθής χρήσης της. Ακόμη κι όταν επιχειρήθηκε να προκληθεί σφάλμα εσκεμμένα, η ανάνηψη ήταν πάντοτε δυνατή με την απλή εναλλακτική της επίσκεψης της αρχικής σελίδας ενώ ουδέποτε απαιτήθηκε να χρησιμοποιηθεί η ιστοσελίδα που αρχικοποιεί πλήρως την εφαρμογή.

10.7 Προειδοποιήσεις

Για την ορθή λειτουργία της εφαρμογής απαιτείται ο χρήστης να έχει ενεργοποιημένη την JavaScript στον φυλλομετρητή.

Συνιστάται να μην χρησιμοποιούνται τα πλήκτρα "Back" και "Forward" του φυλλομετρητή. Ακόμη, συνιστάται να μην διατηρείτε ανοιχτές διαφορετικές καρτέλες στις οποίες εκτελείτε διαφορετικά μέρη της εφαρμογής – αν και, τις περισσότερες φορές, ακόμη και κάτι τέτοιο πιθανότατα θα λειτουργούσε σωστά.

Η περίοδος χρήσης της εφαρμογής (session) λήγει αν ο χρήστης παραμείνει ανενεργός για 45 λεπτά.

Η εφαρμογή έχει δοκιμαστεί και είναι συμβατή με τους εξής φυλλομετρητές: Mozilla Firefox (v47 και μετά), Google Chrome (v55 και μετά).

Τη στιγμή της συγγραφής της διπλωματικής εργασίας, η εφαρμογή φιλοξενείται στη διεύθυνση:
http://snf-723479.vm.okeanos.grnet.gr/
Λόγω παρέλευσης της φοιτητικής ιδιότητας του συγγραφέα, η εν λόγω διεύθυνση θα πάψει να λειτουργεί μετά από κάποιο χρονικό διάστημα αφότου ολοκληρωθεί η διπλωματική εργασία.

11

Παράρτημα ΙΙ: Ευρετήριο πηγών & εργαλείων

Βιολογικές βάσεις δεδομένων

Ensembl	http://www.ensembl.org/
miRBase	http://mirbase.org/

Αλγόριθμοι πρόβλεψης αλληλεπιδράσεων

DIANA-microT-CDS	http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=microT_CDS/index
TargetScan	http://www.targetscan.org/
MirTarget	http://mirdb.org/

Προγραμματιστικά εργαλεία

~Okeanos	https://okeanos.grnet.gr/
Ubuntu	https://www.ubuntu.com/
Apache	https://httpd.apache.org/
MySQL	https://www.mysql.com/
PHP	http://php.net/
GNU AWK	https://www.gnu.org/software/gawk/
Laravel	https://laravel.com/
D3	https://d3js.org/
jQuery	https://jquery.com/
Bootstrap	http://getbootstrap.com/
git	https://git-scm.com/