



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ
ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ

**Ανάπτυξη φιλικής προς τον χρήστη εφαρμογής για την
αυτοματοποιημένη ανάλυση από ανεπεξέργαστες μεταγραφωμικές
αλληλουχίες σε αποτελέσματα έτοιμα προς δημοσίευση**

Λύτρα Μαγδαληνή Ούρσουλα

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Άρτεμις Χατζηγεωργίου
Καθ. Τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική

Αναγνώστης Αργυρίου
Καθ. Πανεπιστημίου Αιγαίου, Αναπ. Διευθ. INEB - ΕΚΕΤΑ

Λαμία, 2023



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ
ΒΙΟΙΑΤΡΙΚΗ**

**Ανάπτυξη φιλικής προς τον χρήστη εφαρμογής για την
αυτοματοποιημένη ανάλυση από ανεπεξέργαστες μεταγραφωμικές
αλληλουχίες σε αποτελέσματα έτοιμα προς δημοσίευση**

Λύτρα Μαγδαληνή Ούρσουλα

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Άρτεμις Χατζηγεωργίου
Καθ. Τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική**

**Αναγνώστης Αργυρίου
Καθ. Πανεπιστημίου Αιγαίου, Αναπ. Διευθ. INEB - ΕΚΕΤΑ**

Λαμία, 2023

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία:/...../20.....

Ο – Η Δηλ.

(Υπογραφή)

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.

**Ανάπτυξη φιλικής προς τον χρήστη εφαρμογής για την
αυτοματοποιημένη ανάλυση από ανεπεξέργαστες μεταγραφωμικές
αλληλουχίες σε αποτελέσματα έτοιμα προς δημοσίευση**

Λύτρα Μαγδαληνή Ούρσουλα

Τριμελής Επιτροπή:

Ονοματεπώνυμο, Βαθμίδα(επιβλέπων/σα)

Ονοματεπώνυμο, Βαθμίδα.....

Ονοματεπώνυμο, Βαθμίδα.....

Ευχαριστίες

Η παρούσα πτυχιακή εργασία, εκπονήθηκε από το Φεβρουάριο του 2023 έως τον Ιούνιο του ίδιου έτους στο ερευνητικό ινστιτούτο INEB - ΕΚΕΤΑ, στα πλαίσια της απόκτησης του βασικού πτυχίου από το τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική, του Πανεπιστημίου Θεσσαλίας. Κατά τη διάρκεια του κύκλου των σπουδών μου, ανακάλυψα το πραγματικό ερευνητικό μου ενδιαφέρον, διευρύνθηκαν οι ορίζοντες μου και εξελίχθηκα ως άνθρωπος.

Η παρούσα πτυχιακή εργασία υλοποιήθηκε με την υποστήριξη ορισμένων ανθρώπων, για τους οποίους θα ήταν παράλειψη να μην εκφράσω τις θερμότερες ευχαριστίες μου. Αρχικά θα ήθελα να ευχαριστήσω θερμά την επιβλέπουσα καθηγήτρια μου κυρία Χατζηγεωργίου Άρτεμις, για την ευκαιρία που μου έδωσε να βρεθώ ένα βήμα πιο κοντά στον στόχο μου αλλά και για την συνεργασία που είχαμε κατά τη διάρκεια εκπόνησης της πτυχιακής μου εργασίας.

Επιπλέον, ιδιαίτερα οφείλω να ευχαριστήσω τον κύριο Αργυρίου Αναγνώστη, καθηγητή Πανεπιστημίου Αιγαίου και αναπληρωτή διευθυντή του ινστιτούτου INEB - ΕΚΕΤΑ, για την εξαιρετική μας συνεργασία και την υποστήριξή του καθ' όλη τη διάρκεια διεξαγωγής της εργασίας αυτής. Ακόμη, θα ήθελα να ευχαριστήσω θερμά τον άμεσο επιβλέποντα της πτυχιακής μου εργασίας, κύριο Τσιόλα Γεώργιο, βοηθό έρευνας στο ινστιτούτο INEB – ΕΚΕΤΑ, για την καθοδήγηση και τη βοήθεια που μου πρόσφερε σε όλη τη διάρκεια της εκπόνησης της εργασίας μου καθώς αποτέλεσε και αποτελεί για εμένα πηγή έμπνευσης και γνώσης. Η καθοδήγηση που μου παρείχε και η εμπιστοσύνη που μου έδειξε, ήταν άκρως σημαντικές για την ομαλή διεξαγωγή της εργασίας αυτής. Τέλος, θα ήθελα να ευχαριστήσω τα μέλη της ερευνητικής ομάδας του κυρίου Αργυρίου καθώς και άλλα μέλη του ινστιτούτου, τα οποία μέσω εύστοχων παρατηρήσεων και σχολίων με βοήθησαν στην βελτιστοποίηση της εργασίας αυτής.

Περίληψη

Η εφαρμογή που αναπτύχθηκε στο πλαίσιο της παρούσας εργασίας, στοχεύει στη διευκόλυνση βιολόγων που διαθέτουν το επιστημονικό υπόβαθρο για αναλύσεις “RNASeq”, αλλά έχουν μηδαμινές ή ελάχιστες προγραμματιστικές γνώσεις. Η μείωση του κόστους αλληλούχησης έχει φέρει αυτού του είδους τις μεθοδολογίες στην ερευνητική καθημερινότητα, ωστόσο αρκετοί βιολόγοι είτε δε διαθέτουν τον απαραίτητο χρόνο είτε το κατάλληλο βιοπληροφορικό υπόβαθρο, με αποτέλεσμα να μην είναι σε θέση να αξιοποιήσουν τις νέες αυτές μεθοδολογίες. Έτσι, σε πολλές ερευνητικές ομάδες όπου υπάρχει έλλειψη αυτών των γνώσεων, είναι πολύτιμη η πρόσβαση σε ένα εύχρηστο εργαλείο που να πραγματοποιεί εύκολα και γρηγορά τις απαραίτητες αναλύσεις.

Η εν λόγω εφαρμογή προσφέρει ένα γραφικό περιβάλλον αλληλεπίδρασης φιλικό προς τους χρήστες. Στο περιβάλλον αυτό γίνεται η μεταφόρτωση των αρχείων και στη συνέχεια πραγματοποιούνται διάφορες αναλύσεις δεδομένων από πειράματα “RNASeq”. Τα στάδια της ανάλυσης που λαμβάνουν χώρα είναι το φιλτράρισμα των ακατέργαστων αλληλουχιών από πειράματα “RNASeq” και η εσωτερική αξιολόγηση της ποιότητάς τους, η στοίχιση στο γονιδίωμα αναφοράς, η ποσοτικοποίηση των αλληλουχιών και η μετατροπή τους σε επίπεδα γονιδιακής έκφρασης, η ανάλυση διαφορικής έκφρασης και η ανάλυση εμπλουτισμού γονιδιακών συνόλων. Τα αποτελέσματα των παραπάνω αναλύσεων είναι διαθέσιμα στους χρήστες μέσω του γραφικού περιβάλλοντος και έχουν την επιλογή για τη λήψη τους σε διάφορες μορφές αρχείων (“.csv”, “.pdf”, “.png” κ.α.). Όλες οι παραπάνω διεργασίες και αναλύσεις εκτελούνται στην προγραμματιστική γλώσσα R αξιοποιώντας υπάρχοντα πακέτα που είναι ευρέως αποδεκτά από την ερευνητική κοινότητα.

Μελλοντικά, έπειτα από μικρές τροποποιήσεις, η εφαρμογή θα μπορούσε να χρησιμοποιηθεί στην κλινική διάγνωση και στο πλαίσιο της εξατομικευμένης ιατρικής, έτσι ώστε να διερευνάται η επιλογή της κατάλληλης θεραπείας από γιατρούς εξετάζοντας την απόκριση των ασθενών σε αυτές.

Περιεχόμενα

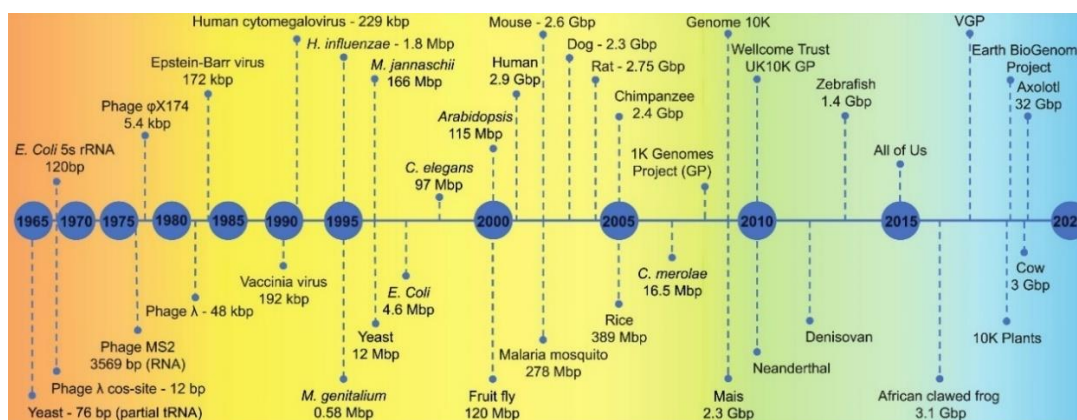
Ευχαριστίες.....	5
Περίληψη.....	6
1. Εισαγωγή	9
1.1. Ιστορία της αλληλούχησης	9
1.1.1. Αλληλούχηση πρώτης γενιάς.....	10
1.1.2. Αλληλούχηση επόμενης γενιάς (NGS)	11
1.1.3. Illumina	12
1.1.4. Αλληλούχηση τρίτης γενιάς (TGS).....	13
1.1.5. Αλληλούχηση RNA	13
1.1.6. Προετοιμασία βιβλιοθηκών RNA-seq	14
1.2. Βιοπληροφορική	16
1.2.1. Σύγχρονες εφαρμογές της βιοπληροφορικής	18
1.2.2. Μελλοντικές εφαρμογές και εξελίξεις της βιοπληροφορικής.....	19
1.2.3. Βιοπληροφορική και RNA-seq	20
1.2.4. Γλώσσα προγραμματισμού R και περιβάλλον RStudio	21
1.2.5. Shiny	22
1.2.6. Simplar.....	23
2. Μεθοδολογία.....	24
2.1. Αρχιτεκτονική εφαρμογής	24
2.1.1. Αρχιτεκτονική αντικειμένου διεπαφής χρήστη UI	24
2.1.2. Ειδική μορφοποίηση του UI	26
2.1.3. Αρχιτεκτονική συνάρτησης διακομιστή server.....	27
2.2. Μεθοδολογία διεργασιών εφαρμογής.....	30
2.2.1. Αντιγραφή αρχείων	30
2.2.2. Φιλτράρισμα (Trimming)	32
2.2.3. Κατασκευή Ευρετηρίου “Index”	35
2.2.4. Στοίχιση (Alignment)	36
2.2.5. Ποσοτικοποίηση (Quantification)	38
2.2.6. Ανάλυση διαφορικής έκφρασης (Differential expression)	41
2.2.7. Ανάκτηση πληροφοριών για την ανάλυση εμπλουτισμού γονιδιακών συνόλων 44	
2.2.8. Ανάλυση εμπλουτισμού γονιδιακών συνόλων (Gene set enrichment)	45

2.3.	Συνολική αξιολόγηση της εφαρμογής από χρήστες.....	49
2.4.	Οδηγίες χρήσης της εφαρμογής.....	60
3.	Αποτελέσματα.....	64
3.1.	Εικόνες από το γραφικό περιβάλλον της εφαρμογής	64
3.2.	Αρχεία VCF.....	68
3.1.	Στατιστικά αποτελέσματα.....	69
3.2.	Οπτικοποίηση αποτελεσμάτων.....	72
3.2.1.	Διαγράμματα της ανάλυσης διαφορικής έκφρασης.....	72
3.2.2.	Διαγράμματα της ανάλυσης εμπλουτισμού γονιδιακών συνόλων	79
4.	Συγκριτική Αξιολόγηση	84
4.1.	Χρονικές διαφορές στην ανάλυση βάσει οργανισμού και μεγέθους αρχείων	85
4.1.1.	Δεδομένα από τον οργανισμό <i>Vitis vinifera</i> (αμπέλι)	85
4.1.2.	Δεδομένα από τον μοντέλο οργανισμό <i>Homo sapiens</i> (άνθρωπος)	86
5.	Συμπεράσματα- Συζήτηση	87
6.	Παράρτημα Κώδικα	90
6.1.	Κώδικας από το “script” με όνομα 'global2v.r'	90
6.2.	Κώδικας από το “script” με όνομα 'shiny_universal.r'	91
7.	Βιβλιογραφία	122

1. Εισαγωγή

1.1. Ιστορία της αλληλούχησης

Η δομή της διπλής έλικας του DNA ανακαλύφθηκε το 1953 από τους James Watson και Francis Crick. Η Rosalind Franklin ήταν μια Αγγλίδα κρυσταλλογράφος ακτίνων-X, η οποία επίσης συνέβαλε μέγιστα και είχε καθοριστική σημασία για την κατανόηση των μοριακών δομών του DNA. Οι Watson και Crick έλαβαν το βραβείο Νόμπελ το 1962 για την εκπληκτική συμβολή τους στην επιστήμη, ενώ λίγο αργότερα ακολούθησε ο Robert Holley το 1968, επειδή ήταν ο πρώτος άνθρωπος που κατάφερε να αλληλουχήσει ένα μόριο RNA. Ο συνδυασμός αυτών των πρωτοποριακών ανακαλύψεων άνοιξε το δρόμο για την αλληλούχηση του DNA. Ακολουθούν μερικές από τις πιο καθοριστικές στιγμές για την αλληλούχηση του DNA, η οποία αποτελεί ένα νέο ερευνητικό πεδίο με συνεχή ανάπτυξη, από τη δημιουργία του έως σήμερα, τα τελευταία 50 χρόνια (Εικόνα 1.).

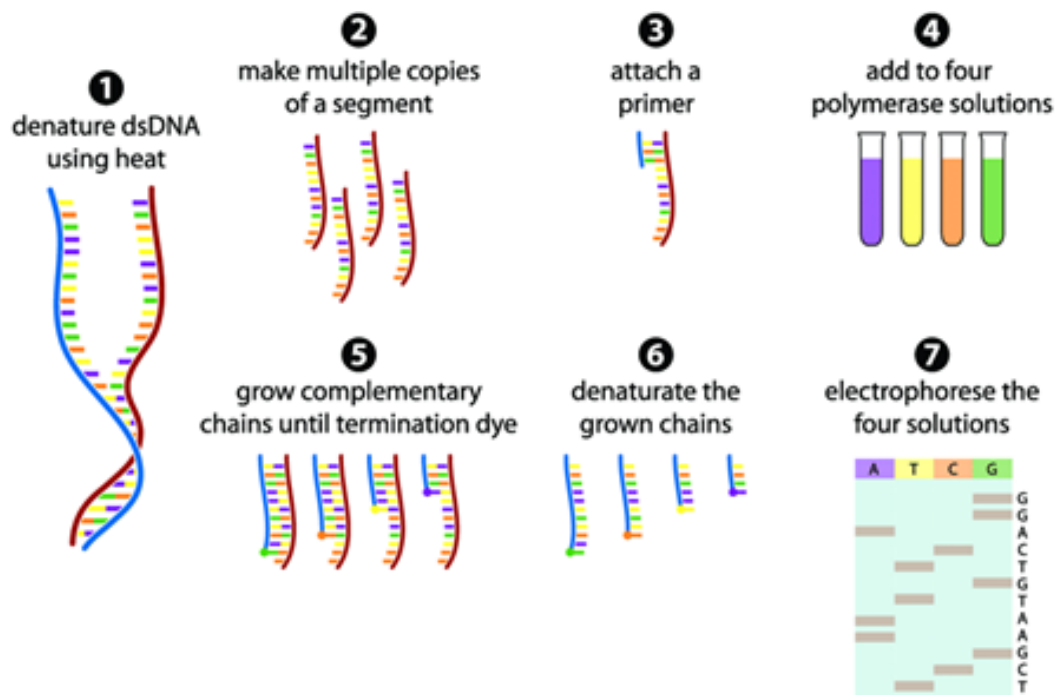


Εικόνα 1: Χρονολόγιο που απεικονίζει πολλά από τα σημαντικότερα επιτεύγματα της συναρμολόγησης γονιδιωμάτων από την αρχή της αλληλούχησης μέχρι τα έργα γονιδιωμάτων μεγάλης κλίμακας που βρίσκονται σε εξέλιξη. Κάθε γονιδίωμα ή έργο γονιδιώματος (GP) τοποθετείται κάτω από ένα χρωματικά κωδικοποιημένο φόντο ανάλογα με την προσέγγιση αλληλούχησης που υιοθετήθηκε. Ανοιχτό κόκκινο: πρώιμες μέθοδοι αλληλούχησης, κίτρινο: Αλληλούχηση shotgun βάσει Sanger, πράσινο: NGS (αλληλούχηση επόμενης γενιάς), γαλάζιο: TGS (αλληλούχηση τρίτης γενιάς) ¹

1.1.1. Αλληλούχηση πρώτης γενιάς

Η αλληλούχηση πρώτης γενιάς (FGS) αναφέρεται στις αρχικές τεχνικές αλληλούχησης DNA που αναπτύχθηκαν στις δεκαετίες του 1970 και 1980, οι οποίες περιλαμβάνουν τη μέθοδο αλληλούχησης κατά Sanger και την αλληλούχηση κατά Maxam-Gilbert (frontlinegenomics, history of sequencing). Η πρώτη σημαντική ανακάλυψη στην τεχνολογία της αλληλούχησης έγινε από τον Fredrick Sanger το 1977. Η μέθοδος αυτή χρησιμοποιήθηκε αργότερα από τον Sanger και τους συνεργάτες του για την αλληλούχηση του ανθρώπινου μιτοχονδριακού DNA (16.569 ζεύγη βάσεων) και του βακτηριοφάγου λ (48.502 ζεύγη βάσεων) - το πρώτο πλήρες γονιδίωμα. Η αλληλούχηση κατά Sanger ήταν η πιο ευρέως χρησιμοποιούμενη μορφή αλληλούχησης για πάνω από 30 χρόνια.

Τα στάδια της αλληλούχησης κατά Sanger είναι τα εξής, αρχικά γίνεται η προετοιμασία του δείγματος DNA, δηλαδή το δείγμα καθαρίζεται και διαχωρίζεται από δίκλωνο DNA σε μονόκλωνα θραύσματα DNA. Στη συνέχεια μικροί ολιγονουκλεοτιδικοί εκκινητές συνδέονται με το μονόκλωνο DNA σε συγκεκριμένες θέσεις που πλαισιώνουν την περιοχή ενδιαφέροντος. Η DNA πολυμεράση χρησιμοποιείται για τη σύνθεση νέων αλυσίδων DNA συμπληρωματικών προς την πρότυπη αλυσίδα, ξεκινώντας από τις θέσεις των εκκινητών. Το επόμενο βήμα περιλαμβάνει την ενσωμάτωση διδεοξυνουκλεοτιδίων, εκτός από τα κανονικά δεοξυνουκλεοτίδια (dATP, dCTP, dGTP, dTTP) που προστίθενται για την επιμήκυνση της αλυσίδας του DNA, προστίθεται και μια μικρή ποσότητα τροποποιημένων τριφωσφορικών δι-δεοξυνουκλεοτιδίων (ddATP, ddCTP, ddGTP, ddTTP). Αυτά τα διδεοξυνουκλεοτίδια στερούνται της ομάδας 3'-OH που απαιτείται για την προσθήκη του επόμενου νουκλεοτιδίου, τερματίζοντας έτσι τη διαδικασία επιμήκυνσης της αλυσίδας. Έπειτα γίνεται ο διαχωρισμός τμημάτων DNA. Οι νεοσυντιθέμενες αλυσίδες DNA διαφορετικού μήκους, που τερματίζονται με ένα από τα τέσσερα διδεοξυνουκλεοτίδια, διαχωρίζονται με βάση το μέγεθος με ηλεκτροφόρηση σε «gel». Τέλος η οπτικοποίηση και η ερμηνεία των δεδομένων αλληλούχησης γίνεται με τη χρήση φθορισμού ή ραδιενεργών ετικετών και η αλληλουχία προσδιορίζεται διαβάζοντας τη σειρά των τερματικών νουκλεοτιδίων σε πηκτή αγαρόζης ή σε άλλα πολυμερή, όπως παρουσιάζεται στην Εικόνα 2.



Εικόνα 2: Αλληλούχηση κατά Sanger (frontlinegenomics, history of sequencing)

Ωστόσο, η αλληλούχηση κατά Sanger δεν ήταν αυτοματοποιημένη και ήταν εξαιρετικά χρονοβόρα. Λόγω των δυνατοτήτων της αλληλούχησης κατά Sanger, έγιναν πολλές προσπάθειες για την αυτοματοποίησή της. Το 1987, ο Leroy Hood και ο Michael Hunkapiller πέτυχαν την αυτοματοποίηση της διαδικασίας και έφεραν δύο σημαντικές βελτιώσεις στη μέθοδο. Πρώτον, τα θραύσματα DNA επισημάνθηκαν με φθορίζουσες χρωστικές ουσίες αντί για ραδιενεργά μόρια και δεύτερον, κατέστη δυνατή η απόκτηση και ανάλυση δεδομένων στον υπολογιστή. Μειονεκτήματα των τεχνικών FGS είναι ότι είναι αργές, ακριβές και απαιτούν σχετικά μεγάλες ποσότητες DNA, περιορίζοντας τη χρήση τους σε έργα αλληλούχησης μικρής κλίμακας.

1.1.2. Αλληλούχηση επόμενης γενιάς (NGS)

Η αυτοματοποιημένη μέθοδος Sanger είχε οδηγήσει σε μια σειρά σημαντικών επιτευγμάτων, συμπεριλαμβανομένης της ολοκλήρωσης της αλληλούχησης του ανθρώπινου γονιδιώματος και άλλων επιλεγμένων γονιδιωμάτων ζώων και φυτών. Ωστόσο, οι περιορισμοί της μεθόδου ανέδειξαν την ανάγκη για νέες και βελτιωμένες τεχνολογίες για την αλληλούχηση μεγάλου αριθμού ανθρώπινων και άλλων γονιδιωμάτων. Στα τέλη του 20ού και στις αρχές του 21ου αιώνα, καταβλήθηκαν προσπάθειες για την ανάπτυξη νέων μεθόδων που θα

αντικαθιστούσαν την αυτοματοποιημένη μέθοδο Sanger. Οι νεότερες μέθοδοι αναφέρονται ως αλληλούχιση επόμενης γενιάς (NGS) και η χρήση τους έχει αλλάξει τις επιστημονικές προσεγγίσεις τόσο στη βασική όσο και στην εφαρμοσμένη έρευνα σε πολλούς επιστημονικούς κλάδους, ιδίως σε πολλούς κλάδους του βιολογικού πεδίου.

Η αλληλούχιση επόμενης γενιάς δίνει τη δυνατότητα παραγωγής τεράστιου όγκου δεδομένων, καθώς και η ικανότητά της να παρέχει γρήγορα και φθηνά, αξιόπιστη γονιδιωματική πληροφορία. Αυτές οι τεχνολογίες, οι οποίες περιλαμβάνουν την αλληλούχιση Illumina, Roche 454 και Ion Torrent, χρησιμοποιούν μαζικά παράλληλη αλληλούχιση για τη δημιουργία εκατομμυρίων αναγνώσεων (reads) αλληλουχίας κάθε φορά, επιτρέποντας ταχύτερη και οικονομικότερη αλληλούχιση από τις μεθόδους FGS. Οι τεχνικές NGS βασίζονται συνήθως στην ενίσχυση τμημάτων DNA μέσω PCR, η οποία μπορεί να εισάγει σφάλματα στα δεδομένα αλληλούχισης. Παρά τους περιορισμούς αυτούς, η NGS έχει φέρει επανάσταση στη γονιδιωματική και έχει καταστήσει εφικτά έργα αλληλούχισης μεγάλης κλίμακας, όπως η αλληλούχιση ολόκληρου του γονιδιώματος και η σκιαγράφηση του μεταγραφικού προφίλ. Η NGS βασίζεται σε πλατφόρμες που παράγουν μεγάλες ποσότητες αναγνώσεων DNA, με μήκος μεταξύ 25 και 400 bp. Αυτές οι αναγνώσεις είναι μικρότερες από τις αναγνώσεις της αλληλουχίας Sanger (300 έως 750 bp). Πρόσφατα, ωστόσο, οι τεχνολογίες NGS έχουν εξελιχθεί ώστε να παράγουν αναγνώσεις DNA μεγαλύτερες από 750 bp. Προς το παρόν, όλες οι τεχνικές NGS απαιτούν προετοιμασία βιβλιοθήκης. Το πρωτόκολλο αυτό λαμβάνει χώρα μετά τον κατακερματισμό του DNA/RNA, όπου προσδένονται προσαρμογείς στα άκρα κάθε θραύσματος. Συνήθως ακολουθείται από ένα στάδιο ενίσχυσης του DNA/RNA για να προκύψει μια βιβλιοθήκη που μπορεί στη συνέχεια να αλληλουχηθεί από την πλατφόρμα NGS. Η ποιότητα των δεδομένων που προσφέρει η NGS σε συνδυασμό με το προσιτό κόστος, τις βελτιωμένες δυνατότητες χειρισμού δεδομένων, την αυξημένη υπολογιστική ισχύ και τα αποτελεσματικά εργαλεία βιοπληροφορικής ανάλυσης, έχουν διευκολύνει σημαντικά την ενσωμάτωση στρατηγικών γενετικής ανάλυσης με βάση την αλληλούχιση επόμενης γενιάς στην κλινική διάγνωση και τη γενετική ιατρική.

1.1.3. Illumina

Η Illumina, είναι ίσως η πιο δημοφιλής από τις διαθέσιμες σημερινές πλατφόρμες NGS, προσφέροντας διάφορες κλιμακούμενες επιλογές που συμπληρώνουν τις απαιτήσεις διαφορετικών σχεδίων μελέτης, το κόστος της αλληλούχισης και την προβλεπόμενη χρήση

της αλληλούχησης. Αυτές οι ιδιότητες παρέχουν στους πελάτες προσιτές επιλογές και ευελιξία κατά το σχεδιασμό των μελετών τους. Η Illumina προσφέρει μια μέθοδο για την επιλογή της βέλτιστης πλατφόρμας αλληλούχησης μέσω του εργαλείου σύγκρισης πλατφορμών αλληλούχησης. Οι διάφορες πλατφόρμες παράγουν διαφορετική ποσότητα αναγνώσεων αλληλούχησης σε διαφορετικούς χρόνους εκτέλεσης.

1.1.4. Αλληλούχηση τρίτης γενιάς (TGS)

Μια νέα γενιά τεχνολογιών, που ονομάζεται αλληλούχηση τρίτης γενιάς (TGS), βρίσκεται υπό ενεργή χρήση και περεταίρω ανάπτυξη. Οι τεχνικές TGS, όπως η χημεία της PacBio και της Oxford Nanopore, προσφέρουν αρκετά πλεονεκτήματα σε σχέση με την NGS, όπως μεγαλύτερα μήκη ανάγνωσης, αλληλούχηση σε πραγματικό χρόνο και δυνατότητα αλληλούχησης εγγενούς DNA χωρίς ενίσχυση. Η TGS έχει επίσης τη δυνατότητα να αντιμετωπίσει ορισμένους από τους περιορισμούς της NGS, όπως οι δυσκολίες στην αλληλούχηση επαναλαμβανόμενων περιοχών του γονιδιώματος και στην ανίχνευση δομικών παραλλαγών. Ωστόσο, οι τεχνολογίες TGS είναι επί του παρόντος πιο ακριβές και έχουν υψηλότερα ποσοστά σφάλματος από την NGS, με αποτέλεσμα να χρησιμοποιούνται λιγότερο ευρέως για εφαρμογές αλληλούχησης. Η ακρίβεια και η μείωση του κόστους θα συνεχίσουν να εξελίσσονται, επιτρέποντας την εφαρμογή της TGS για ένα ευρύ φάσμα εφαρμογών στη γονιδιωματική.

1.1.5. Αλληλούχηση RNA

Η αλληλούχηση των ριβονουκλεϊκών οξέων (RNA sequencing ή RNA-seq), είναι ένα ισχυρό εργαλείο για τη μελέτη της γονιδιακής έκφρασης και της μεταγραφωμικής. Η μεθοδολογία RNA-seq επιτρέπει στους ερευνητές να αναλύσουν ολόκληρο το μεταγράφημα ενός κυττάρου ή ιστού, το οποίο περιλαμβάνει όλα τα μόρια RNA που εκφράζονται, συμπεριλαμβανομένου του αγγελιοφόρου RNA (mRNA), του μη κωδικού RNA (non-coding RNA, ncRNA) και του ρυθμιστικού RNA (rRNA, tRNA, scRNA). Η τεχνολογία αυτή έδωσε τη δυνατότητα ποσοτικοποίησης του επιπέδου έκφρασης πολλών γονιδίων ταυτόχρονα, επιτρέποντας τη μελέτη των λειτουργιών ενός οργανισμού σε βάθος. Η συνεχής μείωση του κόστους και η ανεξαρτησία των πρωτοκόλλων προετοιμασίας βιβλιοθηκών, ενίσχυσαν αυτή την τεχνολογία, παράγοντας μεγάλα σύνολα δεδομένων για συγκεκριμένες ασθένειες, τα

οποία, με τη σειρά τους, μετέτρεψαν τη μεταγραφωμική έρευνα σε ευρέως χρησιμοποιούμενη μεθοδολογία.

Η RNA-seq περιλαμβάνει διάφορα βασικά βήματα τα οποία είναι η εκχύλιση RNA, η προετοιμασία βιβλιοθήκης, η αλληλούχηση και η ανάλυση δεδομένων. Αρχικά, το ολικό RNA εξάγεται από το δείγμα υπό μελέτη, το οποίο μπορεί να είναι κύτταρα, ιστοί ή άλλο βιολογικό υλικό. Το RNA μετατρέπεται στη συνέχεια σε βιβλιοθήκη cDNA, η οποία περιλαμβάνει διάφορα στάδια, όπως ο κατακερματισμός του RNA, η αντίστροφη μεταγραφή και η προσθήκη προσαρμογέων. Η βιβλιοθήκη cDNA αλληλουχείται στη συνέχεια με τη χρήση τεχνολογιών αλληλούχησης υψηλής απόδοσης, όπως η Illumina, Ion Torrent ή η Oxford Nanopore. Οι παραγόμενες αναγνώσεις αποτελούν μια ψηφιακή ανάγνωση του μεταγραφώματος του δείγματος. Οι αναγνώσεις αναλύονται στη συνέχεια με τη χρήση εργαλείων βιοπληροφορικής για τον εντοπισμό και τον ποσοτικό προσδιορισμό των μεταγράφων που υπάρχουν στο δείγμα. Αυτό συνήθως περιλαμβάνει την στοίχιση των αναγνώσεων σε ένα γονιδίωμα ή ένα μεταγράφημα αναφοράς και, στη συνέχεια, την ποσοτικοποίηση των επιπέδων έκφρασης με τη χρήση ειδικών εργαλείων.

Η RNA-seq έχει αρκετά πλεονεκτήματα έναντι των παραδοσιακών μεθόδων ανάλυσης της γονιδιακής έκφρασης, όπως οι μικροσυστοιχίες (Microarrays) ή η αλυσιδωτή αντίδραση της πολυμεράσης πραγματικού χρόνου (Real-Time PCR, RT-PCR ή qPCR). Μπορεί να ανιχνεύσει και εναλλακτικά μετάγραφα, καθώς παρέχει και μια πιο ακριβή και ποσοτική μέτρηση της γονιδιακής έκφρασης. Η RNA-seq μπορεί επίσης να χρησιμοποιηθεί για τη μελέτη του εναλλακτικού ματίσματος, των γεγονότων γονιδιακής σύντηξης και της επεξεργασίας RNA. Επιπλέον, έχει πολλές εφαρμογές στη βασική και μεταφραστική έρευνα, όπως ο εντοπισμός βιοδεικτών για τη διάγνωση και την πρόγνωση ασθενειών, η κατανόηση των μοριακών μηχανισμών που διέπουν τις ασθένειες και η ανάπτυξη νέων θεραπευτικών στόχων.

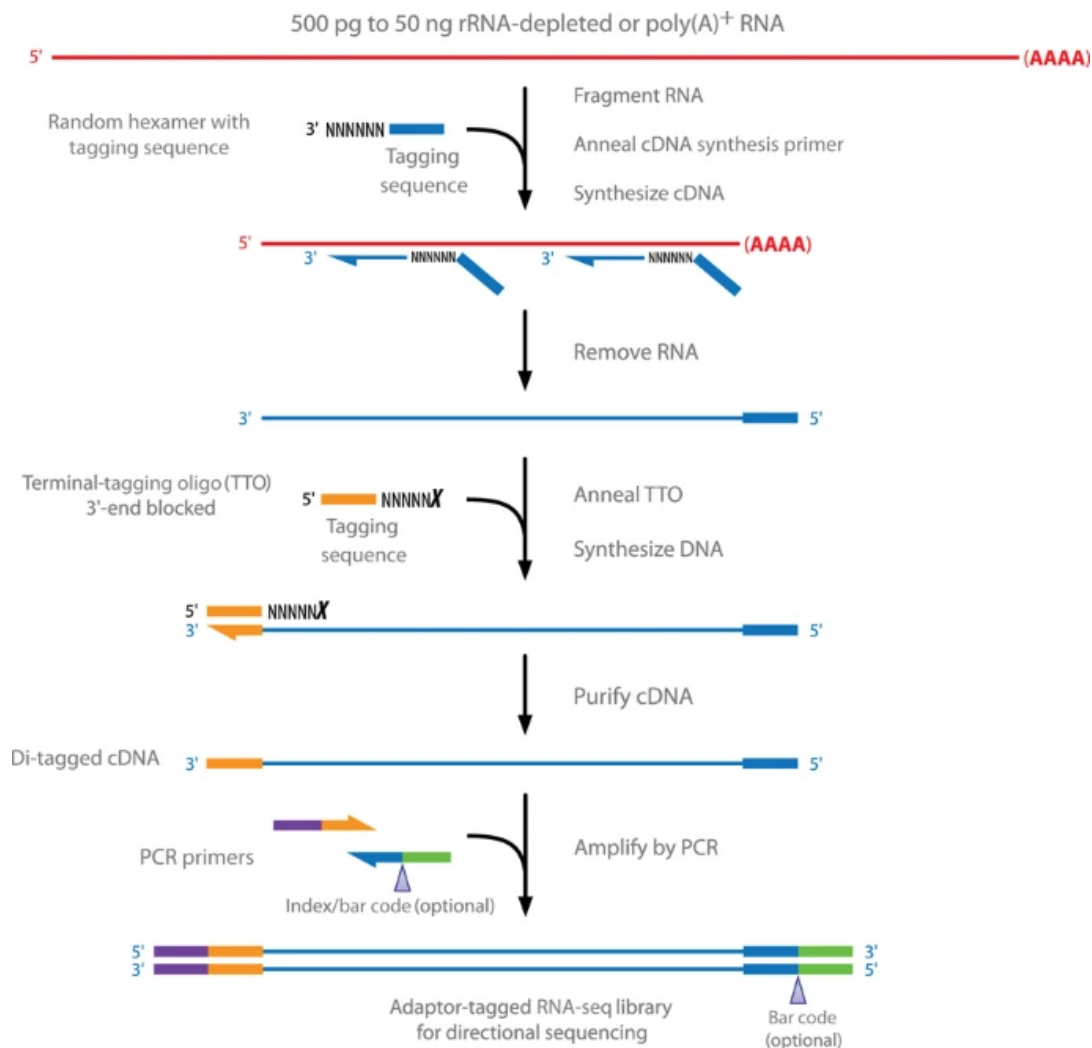
1.1.6. Προετοιμασία βιβλιοθηκών RNA-seq

Η προετοιμασία των βιβλιοθηκών RNA περιλαμβάνει διάφορα στάδια που μετατρέπουν το RNA σε βιβλιοθήκη cDNA κατάλληλη για αλληλούχηση υψηλής απόδοσης. Τα βήματα προετοιμασίας της βιβλιοθήκης μπορεί να ποικίλλουν ανάλογα με το συγκεκριμένο πρωτόκολλο RNA-seq και τον τύπο της πλατφόρμας αλληλούχησης που χρησιμοποιείται.

Το πρώτο βήμα στην προετοιμασία της βιβλιοθήκης RNA-seq είναι η απομόνωση ολικού RNA υψηλής ποιότητας από το δείγμα ενδιαφέροντος. Αυτό μπορεί να γίνει με τη χρήση

εμπορικών κιτ ή άλλων μεθόδων, όπως η εκχύλιση με φαινόλη-χλωροφόρμιο ή ο καθαρισμός με στήλη πυριτίου. Το απομονωμένο RNA κατακερματίζεται στη συνέχεια σε μικρότερα κομμάτια με τη χρήση ενζύμων ή φυσικών μεθόδων. Το στάδιο του κατακερματισμού είναι σημαντικό επειδή επιτρέπει στα θραύσματα RNA να χωρέσουν στο εργαλείο αλληλούχησης. Σε επόμενο βήμα το κατακερματισμένο RNA μεταγράφεται αντίστροφα σε cDNA με χρήση αντίστροφης μεταγραφάσης και τυχαίων εξαμερών εκκινητών. Αυτό μετατρέπει τα θραύσματα RNA σε δίκλωνο cDNA. Μετά τα θραύσματα cDNA υποβάλλονται σε επιδιόρθωση άκρων, η οποία περιλαμβάνει την πλήρωση τυχόν μονόκλωνων προεξοχών και την προσθήκη φωσφορικών ομάδων στα 5' άκρα των θραυσμάτων. Το βήμα αυτό είναι απαραίτητο για την προετοιμασία των θραυσμάτων για τη σύνδεση προσαρμογέα. Οι προσαρμογείς είναι σύντομες αλληλουχίες DNA που περιέχουν θέσεις πρόσδεσης για την πλατφόρμα αλληλούχησης και προστίθενται στα θραύσματα cDNA για να επιτραπεί η προσκόλληση στη ροή αλληλούχησης (Εικόνα 3). Τα συνδεδεμένα με προσαρμογέα θραύσματα cDNA επιλέγονται ως προς το μέγεθος και καθαρίζονται με ηλεκτροφόρηση σε πηκτή αγαρόζης ή με μεθόδους που βασίζονται σε σφαιρίδια. Το στάδιο αυτό απομακρύνει τυχόν ανεπιθύμητα θραύσματα, όπως διμερή του προσαρμογέα ή θραύσματα που είναι πολύ μικρά ή πολύ μεγάλα για την αλληλούχηση. Τα καθαρισμένα θραύσματα cDNA ενισχύονται στη συνέχεια με PCR χρησιμοποιώντας εκκινητές που συνδέονται με τις αλληλουχίες προσαρμογέα. Με τον τρόπο αυτό ενισχύονται τα θραύσματα ώστε να παραχθεί αρκετό υλικό για την αλληλούχηση. Η τελική βιβλιοθήκη υποβάλλεται σε ποιοτικό έλεγχο χρησιμοποιώντας μεθόδους όπως ηλεκτροφόρηση σε πηκτή αγαρόζης, ή βιοαναλυτή. Αυτό εξασφαλίζει ότι η βιβλιοθήκη περιέχει την κατάλληλη κατανομή μεγέθους θραυσμάτων και συγκέντρωση για την αλληλούχηση.

Η προετοιμασία της βιβλιοθήκης RNA-seq επηρεάζει άμεσα την ποιότητα και την ακρίβεια της ανάλυσης που ακολουθεί για αυτό πρέπει να δίνεται ιδιαίτερη προσοχή σε κάθε βήμα ώστε να διασφαλίζεται ότι η βιβλιοθήκη που προκύπτει είναι αντιπροσωπευτική του αρχικού δείγματος RNA και κατάλληλη για αλληλούχηση υψηλής ποιότητας.



Εικόνα 3: Προετοιμασία βιβλιοθήκης RNA ²

1.2. Βιοπληροφορική

Η βιοπληροφορική είναι ένας τομέας που συνδυάζει την επιστήμη των υπολογιστών, τη στατιστική, τα μαθηματικά και τη βιολογία για την ανάλυση και την ερμηνεία βιολογικών δεδομένων. Παρέχει επίσης τη δυνατότητα εξόρυξης δεδομένων δημιουργώντας διάφορες αξιόπιστες υποθέσεις σχετικά με τη δομή ή τη λειτουργία ενός γονιδίου ή μιας πρωτεΐνης. Έχει την ικανότητα να προσδιορίζει γενετικές παραλλαγές και ταυτόχρονα μπορεί να επιτελέσει το απίστευτο έργο της αποκρυπτογράφησης πρωτεϊνών. Παράλληλα διευκολύνει την αποκάλυψη κρυμμένων μοτίβων ή σχέσεων μεταξύ γονιδίων και μοριακών μηχανισμών μέσω της εξόρυξης δεδομένων. Η βιοπληροφορική βοηθά επίσης στην ανάλυση της κατάστασης της νόσου και στην κατανόηση γονιδιώματος οποιουδήποτε οργανισμού. Τα

τρία βασικότερα πράγματα που επιτυγχάνονται από τη βιοπληροφορική είναι πρώτον η ανάπτυξη νέων αλγορίθμων και στατιστικών, δεύτερον, η ανάπτυξη και η εφαρμογή εργαλείων και τρίτον, η ανάλυση και η ερμηνεία. Γενικότερα χρησιμοποιείται στη δημιουργία και αποθήκευση βιολογικών βάσεων δεδομένων, στην πρόβλεψη άγνωστων γονιδίων/πρωτεϊνών, στην αναζήτηση βιβλιογραφίας, στη σύγκριση των δεδομένων με άλλα, στη σύνθεση και διανομή δεδομένων και άλλα.

Η ιστορία της βιοπληροφορικής μπορεί να αναχθεί στις αρχές της μοριακής βιολογίας και στην ανακάλυψη της δομής του DNA τη δεκαετία του 1950. Ακολουθεί μια σύντομη αναδρομή στην ιστορία της βιοπληροφορικής:

Στις δεκαετίες του 1960 και του 1970, οι επιστήμονες πληροφορικής άρχισαν να αναπτύσσουν αλγόριθμους και προγράμματα λογισμικού για την επεξεργασία και ανάλυση αλληλουχιών DNA και πρωτεϊνών. Στα πρώτα προγράμματα περιλαμβάνονταν τα EMBOSS και FASTA, τα οποία χρησιμοποιούσαν τεχνικές στοίχισης ακολουθιών για τη σύγκριση ακολουθιών και τον εντοπισμό ομοιοτήτων.

Στη δεκαετία του 1980, η ανάπτυξη των τεχνολογιών αλληλούχισης του DNA οδήγησε σε πληθώρα δεδομένων αλληλουχιών, δημιουργώντας την ανάγκη για νέους αλγορίθμους και προγράμματα για τον χειρισμό των μεγάλων ποσοτήτων δεδομένων. Αυτό οδήγησε στην ανάπτυξη βάσεων δεδομένων ακολουθιών, όπως η GenBank (<https://www.ncbi.nlm.nih.gov/genbank/>) και η EMBL (<https://www.embl.org/>), οι οποίες επέτρεψαν στους ερευνητές να αποθηκεύουν και να έχουν πρόσβαση σε δεδομένα ακολουθιών.

Στη δεκαετία του 1990, η έκρηξη των γονιδιωματικών δεδομένων που δημιουργήθηκαν από το Πρόγραμμα Ανθρώπινου Γονιδιώματος και άλλα προγράμματα αλληλούχισης μεγάλης κλίμακας οδήγησε στην ανάπτυξη νέων αλγορίθμων και εργαλείων για την ανάλυση και την ερμηνεία των δεδομένων. Αυτό περιελάμβανε την ανάπτυξη αλγορίθμων πρόβλεψης γονιδίων, εργαλείων σχολιασμού ακολουθιών και εργαλείων για τον εντοπισμό γενετικών παραλλαγών.

Στη δεκαετία του 2000, η εμφάνιση τεχνολογιών αλληλούχισης υψηλής απόδοσης, όπως η Illumina και η Roche 454, έφερε επανάσταση στη γονιδιωματική και οδήγησε σε μια νέα εποχή της βιοπληροφορικής έντασης δεδομένων. Αυτό οδήγησε στην ανάπτυξη νέων εργαλείων και αλγορίθμων για τον χειρισμό δεδομένων αλληλούχισης μεγάλης κλίμακας,

συμπεριλαμβανομένων εργαλείων για de novo συναρμολόγηση γονιδιωμάτων και μεταγραφωμάτων και εργαλείων για την ανάλυση μεταγονιδιωματικής.

Στη δεκαετία του 2010, ο τομέας της βιοπληροφορικής επεκτάθηκε πέρα από τη γονιδιωματική και συμπεριέλαβε άλλους τομείς της βιολογίας, όπως η πρωτεομική και η μεταβολομική βιολογία. Αυτό οδήγησε στην ανάπτυξη νέων αλγορίθμων και εργαλείων για τον χειρισμό πολύπλοκων και ετερογενών δεδομένων, συμπεριλαμβανομένων αλγορίθμων μηχανικής μάθησης, εργαλείων ανάλυσης δικτύων και εργαλείων οπτικοποίησης.

1.2.1. Σύγχρονες εφαρμογές της βιοπληροφορικής

Σήμερα, η βιοπληροφορική είναι ένας ταχέως εξελισσόμενος τομέας με ποικίλες εφαρμογές, όπως η ανακάλυψη φαρμάκων και η εξατομικευμένη ιατρική. Η ανάπτυξη νέων αλγορίθμων και προγραμμάτων συνεχίζει να οδηγεί τον τομέα προς τα εμπρός και να επιτρέπει νέες ανακαλύψεις στη βιολογία και την ιατρική. Ορισμένες από τις πιο συχνά χρησιμοποιούμενες μεθόδους ανάλυσης στη βιοπληροφορική είναι η ανάλυση αλληλουχιών, η δομική ανάλυση, η συγκριτική γονιδιωματική, η λειτουργική ανάλυση, η μηχανική μάθηση, μεταγονιδιωματική και επιγονιδιωματική.

Η ανάλυση αλληλουχίας περιλαμβάνει την ανάλυση αλληλουχιών DNA, RNA ή πρωτεϊνών για τον εντοπισμό μοτίβων και μεταλλάξεων. Οι συνήθεις μέθοδοι περιλαμβάνουν στοίχιση ακολουθιών, συναρμολόγηση ακολουθιών, πρόβλεψη γονιδίων και ανακάλυψη μοτίβων. Η δομική ανάλυση πρόκειται για την ανάλυση της τρισδιάστατης δομής των πρωτεϊνών, των νουκλεϊκών οξέων και άλλων μακρομορίων, όπως πρωτεϊνών. Οι συνήθεις μέθοδοι περιλαμβάνουν μοντελοποίηση ομολογίας, προσομοιώσεις μοριακής δυναμικής και πρόσδεση πρωτεΐνης-δέσμου. Η συγκριτική γονιδιωματική, περιέχει τη σύγκριση γονιδιωμάτων από διαφορετικά είδη για τον εντοπισμό συντηρημένων περιοχών, οικογενειών γονιδίων και εξελικτικών σχέσεων. Οι μέθοδοι της συγκριτικής γονιδιωματικής αποτελούνται από την ευθυγράμμιση γονιδιωμάτων και τη φυλογενετική ανάλυση. Η λειτουργική ανάλυση, περιλαμβάνει την ανάλυση της γονιδιακής έκφρασης, της πρωτεϊνικής λειτουργίας και των μεταβολικών μονοπατιών για την κατανόηση των μοριακών μηχανισμών που διέπουν τις βιολογικές διεργασίες. Σημαντικές μέθοδοι λειτουργικής ανάλυσης αποτελούν η ανάλυση γονιδιακής οντολογίας, η ανάλυση μονοπατιών και η ανάλυση δικτύων. Με βάση τη μηχανική μάθηση, γίνεται χρήση αλγορίθμων και στατιστικών μοντέλων για την ανάλυση και ερμηνεία βιολογικών δεδομένων μεγάλης κλίμακας. Οι κυριότερες μέθοδοι περιλαμβάνουν την ταξινόμηση, την ομαδοποίηση, την παλινδρόμηση και τη βαθιά μάθηση. Η μεταγονιδιωματική πρόκειται για την ανάλυση μικροβιακών

κοινοτήτων σε περιβαλλοντικά δείγματα, όπως το έδαφος, το νερό και το ανθρώπινο έντερο. Ειδικότερα πραγματοποιείται ταξινόμηση, λειτουργικός σχολιασμός και συναρμολόγηση μεταγονιδιωματικών δεδομένων. Η επιγονιδιωματική, αφορά την ανάλυση επιγενετικών τροποποιήσεων, όπως η μεθυλίωση του DNA και οι τροποποιήσεις των ιστονών, για την κατανόηση του ρόλου τους στη γονιδιακή ρύθμιση και τις ασθένειες και περιλαμβάνονται μέθοδοι όπως ChIP-seq και ATAC-seq.

Η βιοπληροφορική και η γονιδιωματική διαμορφώνουν συνδυαστικά την εξατομικευμένη ιατρική. Η αντίδραση του οργανισμού σε περιβαλλοντικό στρες προκαλεί μεταβολές στο γονιδίωμα (π.χ. καρκίνοι, καρδιακές παθήσεις και διαβήτης). Χάρη στην ολοκλήρωση της αλληλούχησης του ανθρώπινου γονιδιώματος, στην οποία συνέβαλλε άμεσα η βιοπληροφορική, μπορούμε να αναζητήσουμε τα γονίδια που σχετίζονται άμεσα με διάφορες ασθένειες και να αρχίσουμε να κατανοούμε τις μοριακές βάσεις αυτών των ασθενειών με μεγαλύτερη σαφήνεια. Επιπλέον, η κλινική ιατρική χάρη στη βιοπληροφορική έγινε πιο εξατομικευμένη. Πρώτον, οι υπολογιστές μεταφράζουν το γονιδίωμα ενός συγκεκριμένου ασθενούς σε ψηφιακά δεδομένα τα οποία στη συνέχεια επεξεργάζονται, αποθηκεύονται και ανακτώνται όπως και όταν απαιτείται. Οι γιατροί χρησιμοποιούν αυτά τα γονιδιωματικά δεδομένα για να βρουν το καλύτερο κατάλληλο φάρμακο για τη θεραπεία ενός συγκεκριμένου ασθενούς.

Αυτές οι μέθοδοι ανάλυσης εξελίσσονται και βελτιώνονται διαρκώς, ενώ νέες μέθοδοι και τεχνολογίες αναδεικνύονται συνεχώς. Η σημερινή κατάσταση της βιοπληροφορικής χαρακτηρίζεται από την ενσωμάτωση αυτών των μεθόδων και την ανάπτυξη νέων προσεγγίσεων για την αντιμετώπιση σύνθετων βιολογικών ερωτημάτων.³

1.2.2. Μελλοντικές εφαρμογές και εξελίξεις της βιοπληροφορικής

Το μέλλον της βιοπληροφορικής και των αλγορίθμων αναμένεται να διαμορφωθεί από διάφορες τάσεις και εξελίξεις, όπως τα μεγάλα δεδομένα, η ενσωμάτωση, η τεχνητή νοημοσύνη, η εξατομικευμένη ιατρική και η πολυομική.

Η ποσότητα των βιολογικών δεδομένων που παράγονται αυξάνεται ραγδαία, με τη δυνατότητα να δημιουργηθούν νέες γνώσεις στη βιολογία και την ιατρική. Ωστόσο, αυτό παρουσιάζει επίσης προκλήσεις για την αποθήκευση, την ανάλυση και την ερμηνεία των δεδομένων. Στο μέλλον, οι αλγόριθμοι βιοπληροφορικής θα πρέπει να είναι σε θέση να χειρίζονται μεγαλύτερα και πιο σύνθετα σύνολα δεδομένων. Η βιοπληροφορική ενσωματώνεται όλο και περισσότερο με άλλους τομείς, όπως η επιστήμη των υπολογιστών, η στατιστική και η μηχανική. Η ενσωμάτωση αυτή αναμένεται να οδηγήσει στην ανάπτυξη

νέων αλγορίθμων και μεθόδων που θα μπορούν να χειριστούν τις μοναδικές προκλήσεις των βιολογικών δεδομένων. Επιπλέον, η χρήση της τεχνητής νοημοσύνης και των αλγορίθμων μηχανικής μάθησης αναμένεται να διαδοθεί περισσότερο στη βιοπληροφορική. Αυτοί οι αλγόριθμοι μπορούν να βοηθήσουν στον εντοπισμό μοτίβων σε μεγάλα σύνολα δεδομένων και να κάνουν προβλέψεις για βιολογικά φαινόμενα. Ακόμη, ο τομέας της εξατομικευμένης ιατρικής αναμένεται να οδηγήσει στην ανάπτυξη νέων αλγορίθμων και εργαλείων βιοπληροφορικής. Η εξατομικευμένη ιατρική περιλαμβάνει την προσαρμογή των ιατρικών θεραπειών σε μεμονωμένους ασθενείς με βάση τις γενετικές και άλλες βιολογικές πληροφορίες τους. Τέλος η ενσωμάτωση πολλαπλών τύπων βιολογικών δεδομένων, όπως γονιδιωματικών, πρωτεομικών και μεταβολωμικών, αναμένεται να γίνει πιο διαδεδομένη στη βιοπληροφορική. Αυτό θα απαιτήσει την ανάπτυξη νέων αλγορίθμων και μεθόδων για την ενσωμάτωση και την ανάλυση αυτών των διαφορετικών συνόλων δεδομένων.

Συνολικά, το μέλλον της βιοπληροφορικής και των αλγορίθμων αναμένεται να διαμορφωθεί από την ανάγκη χειρισμού μεγαλύτερων και πιο σύνθετων συνόλων δεδομένων, την σύνδεση με άλλα πεδία, τη χρήση της τεχνητής νοημοσύνης και της μηχανικής μάθησης, καθώς και την αυξανόμενη σημασία της εξατομικευμένης ιατρικής και των προσεγγίσεων πολυομικής. Η ανάπτυξη νέων αλγορίθμων και μεθόδων θα είναι ζωτικής σημασίας για την πρόοδο της κατανόησης της βιολογίας και τη βελτίωση της ανθρώπινης υγείας.

1.2.3. Βιοπληροφορική και RNA-seq

Η ανάλυση δεδομένων RNA-seq περιλαμβάνει συνήθως τα εξής στάδια: φιλτράρισμα των αναγνώσεων (trimming), στοίχιση στο γονιδίωμα αναφοράς (alignment), ποσοτικοποίηση των αναγνώσεων (quantification), ανάλυση διαφορικής έκφρασης (differential expression) και πολύ συχνά ανάλυση εμπλουτισμού γονιδιακών συνόλων (gene set enrichment), ανάλυση γονιδιακής οντολογίας (gene ontology), ανάλυση μονοπατιών (pathway enrichment) και ανάλυση δικτύων.

Το πρώτο βήμα στην ανάλυση δεδομένων RNA-seq είναι η αξιολόγηση της ποιότητας των αναγνώσεων αλληλούχισης με τη χρήση εργαλείων όπως το FastQC. Το βήμα αυτό διασφαλίζει ότι οι αναγνώσεις είναι επαρκούς ποιότητας για την ανάλυση σε επόμενο στάδιο. Το φιλτράρισμα χρησιμοποιείται για την αύξηση του βαθμού και του ρυθμού χαρτογράφησης των αναγνώσεων μέσω της εξάλειψης των αλληλουχιών προσαρμογέα και της αφαίρεσης νουκλεοτιδίων κακής ποιότητας. Αυτό το βήμα πρέπει να γίνεται προσεκτικά με επιλεγμένο μήκος ανάγνωσης, για να αποφεύγονται απρόβλεπτες αλλαγές στη γονιδιακή

έκφραση. Η στοίχιση σε ένα γονιδίωμα αναφοράς ή σε ένα μεταγράφημα είναι το δεύτερο βήμα στη ροή της RNA-seq. Στη συνέχεια ακολουθεί η διαδικασία της ποσοτικοποίησης για την εκτίμηση των επιπέδων έκφρασης των γονιδίων ή των μεταγράφων. Σε αυτό το βήμα γίνεται υπολογισμός του αριθμού των αναγνώσεων που αντιστοιχούν σε ένα συγκεκριμένο γονίδιο ή μεταγράφημα. Το τελικό βήμα σε πολλές μελέτες RNA-seq είναι η ανάλυση διαφορικής έκφρασης. Το κύριο μέλημα σε αυτό το σημείο είναι πώς να ανιχνεύονται με ακρίβεια τα διαφορικά εκφραζόμενα γονίδια (DEGs) μεταξύ δύο ή περισσότερων συνθηκών. Αυτό το βήμα επιτρέπει στους ερευνητές να εντοπίσουν γονίδια που εκφράζονται διαφορετικά μεταξύ των δειγμάτων, τα οποία μπορούν να παράσχουν πληροφορίες για βιολογικές διεργασίες και μονοπάτια που επηρεάζονται από τις πειραματικές συνθήκες. Δεδομένου ότι το βήμα αυτό αποτελεί κρίσιμο στοιχείο της ανάλυσης δεδομένων RNA-seq, έχουν αναπτυχθεί διάφορες μέθοδοι και έχουν δημοσιευθεί πολλές συγκριτικές μελέτες που αξιολογούν την καταλληλότητά τους. Μετά την ανάλυση διαφορικής έκφρασης το επόμενο βήμα είναι η λειτουργική ανάλυση όπως η ανάλυση εμπλουτισμού γονιδιακών συνόλων. Μια μέθοδος για τον εντοπισμό κατηγοριών γονιδίων ή πρωτεϊνών που υπερεκφράζονται σε ένα μεγάλο σύνολο γονιδίων ή πρωτεϊνών και ενδέχεται να έχουν συσχέτιση με φαινότυπους ασθενειών. Η μέθοδος χρησιμοποιεί στατιστικές προσεγγίσεις για τον εντοπισμό σημαντικά υπερεκφραζόμενων ή υποεκφραζόμενων ομάδων γονιδίων. Αυτό το βήμα επιτρέπει στους ερευνητές να αποκτήσουν γνώσεις σχετικά με τις βιολογικές διεργασίες που επηρεάζονται από τις αλλαγές στη γονιδιακή έκφραση. Τέλος, τα αποτελέσματα της ανάλυσης μπορούν να οπτικοποιηθούν με τη χρήση διαφόρων διαγραμμάτων, όπως «heatmaps», διαγράμματα διασποράς και διαγράμματα ηφαιστείου και πολλά άλλα. Έτσι επιτρέπεται στους ερευνητές να διερευνήσουν τα μοτίβα και τις τάσεις στα δεδομένα και να εντοπίσουν πιθανές ακραίες τιμές.

Η ανάλυση δεδομένων RNA-seq είναι ένα κρίσιμο βήμα στη ροή εργασίας RNA-seq, καθώς παρέχει πληροφορίες σχετικά με τα πρότυπα γονιδιακής έκφρασης και τις βιολογικές διεργασίες που επηρεάζονται από τις πειραματικές συνθήκες. Σε κάθε βήμα της ανάλυσης πρέπει να δίνεται ιδιαίτερη προσοχή, ώστε να διασφαλίζεται ότι τα αποτελέσματα είναι ακριβή και αξιόπιστα.

1.2.4. Γλώσσα προγραμματισμού R και περιβάλλον RStudio

Η R⁴ δεν είναι μια γλώσσα προγραμματισμού όπως η C ή η Java και δεν δημιουργήθηκε από μηχανικούς λογισμικού για την ανάπτυξη λογισμικού. Αντίθετα, αναπτύχθηκε από

στατιστικολόγους ως ένα διαδραστικό περιβάλλον για την ανάλυση δεδομένων. Η διαδραστικότητα είναι ένα απαραίτητο χαρακτηριστικό στην επιστήμη των δεδομένων (data science), διότι η ικανότητα γρήγορης ανάλυσης δεδομένων είναι απαραίτητη προϋπόθεση για την επιτυχία σε αυτόν τον τομέα. Ωστόσο, όπως και σε άλλες γλώσσες προγραμματισμού, είναι δυνατή η αποθήκευση της δουλειάς ως «scripts» που μπορούν εύκολα να εκτελεστούν ανά πάσα στιγμή. Αυτά τα «scripts» χρησιμεύουν ως αρχείο της ανάλυσης που έχει εκτελεστεί. Βασικό χαρακτηριστικό της R είναι η ανάλυση δεδομένων και, συγκεκριμένα, η οπτικοποίηση δεδομένων. Άλλα ελκυστικά χαρακτηριστικά της R είναι ότι είναι δωρεάν και ανοικτού κώδικα και ότι λειτουργεί σε όλες τις μεγάλες πλατφόρμες: Windows, Mac Os, UNIX/Linux. Ακόμη τα «scripts» και τα αντικείμενα δεδομένων μπορούν να διαμοιραστούν μεταξύ των πλατφορμών. Επιπλέον υπάρχει μια μεγάλη, αναπτυσσόμενη και ενεργή κοινότητα χρηστών της R και, ως εκ τούτου, υπάρχουν πολυάριθμες πηγές για μάθηση και υποβολή ερωτήσεων, γεγονός που επιτρέπει στους προγραμματιστές να μοιράζονται υλοποιήσεις λογισμικού νέων μεθοδολογιών της επιστήμης δεδομένων. Αυτό δίνει στους χρήστες της R έγκαιρη πρόσβαση στις πιο πρόσφατες μεθόδους και σε εργαλεία που αναπτύσσονται για ένα ευρύ φάσμα επιστημονικών κλάδων, όπως η οικολογία, η μοριακή βιολογία, οι κοινωνικές επιστήμες και η γεωγραφία.

Το RStudio⁵ αποτελεί πολύ σημαντικό εργαλείο για έργα επιστήμης δεδομένων. Δεν παρέχει μόνο έναν κειμενογράφο για τη δημιουργία και την επεξεργασία «scripts», αλλά και πολλά ακόμη χρήσιμα εργαλεία. Ανοίγοντας το RStudio για πρώτη φορά εμφανίζονται τρία παράθυρα. Στο αριστερό παράθυρο εμφανίζεται η κονσόλα R. Στα δεξιά, το επάνω παράθυρο περιλαμβάνει καρτέλες όπως το Περιβάλλον και το Ιστορικό, ενώ το κάτω παράθυρο εμφανίζει πέντε καρτέλες: File, Plots, Packages, Help και Viewer (αυτές οι καρτέλες ενδέχεται να αλλάξουν σε νέες εκδόσεις).

1.2.5. Shiny

Το Shiny είναι ένα πακέτο της R που διευκολύνει τη δημιουργία διαδραστικών εφαρμογών ιστού (apps) απευθείας από την R. Με το πακέτο Shiny μπορείτε να μοιραστείτε τα μοντέλα, τις γραφικές παραστάσεις ή τους πίνακές σας, ώστε να μπορούν να τα χρησιμοποιήσουν και άλλοι με διαδραστικό τρόπο. Οποιοσδήποτε έχει πρόσβαση στο διαδίκτυο μπορεί να αναζητήσει μία εφαρμογή Shiny στο τηλέφωνό του ή στον υπολογιστή του, να εισάγει μερικές τιμές για τις μεταβλητές πρόβλεψης και να λάβει μια εκτίμηση για ένα αποτέλεσμα, χωρίς να κάνει υπολογισμούς μόνος του. Οι εφαρμογές shiny έχουν τρία συστατικά: ένα αντικείμενο διεπαφής χρήστη, μια συνάρτηση διακομιστή και μια κλήση στη συνάρτηση shinyApp. Το αντικείμενο διεπαφής χρήστη (user interface, UI) ελέγχει τη διάταξη και την

εμφάνιση της εφαρμογής. Η συνάρτηση διακομιστή (server) περιέχει τις οδηγίες που χρειάζεται ο υπολογιστής για να κατασκευάσει την εφαρμογή δηλαδή επεξεργάζεται δεδομένα, αποδίδει τις τιμές εξόδου σε HTML και τις επιστρέφει στο αντικείμενο διεπαφής (UI). Τέλος, η συνάρτηση shinyApp δημιουργεί αντικείμενα από το ζεύγος UI /server. Ένα αξιοσημείωτο χαρακτηριστικό των εφαρμογών είναι ότι είναι δυνατή η αντιγραφή και η επικόλληση ολόκληρης της εφαρμογής στην κονσόλα R, γεγονός που καθιστά εύκολη τη γρήγορη κοινή χρήση κώδικα για να πειραματιστούν άλλοι.

1.2.6. Simplar

Το Simplar είναι μια εφαρμογή shiny που περιέχει ένα γραφικό περιβάλλον αλληλεπίδρασης με τους χρήστες. Στο περιβάλλον αυτό γίνεται η μεταφόρτωση αρχείων. Τα στάδια της ανάλυσης που λαμβάνουν χώρα είναι το φιλτράρισμα των ακατέργαστων αλληλουχιών από πειράματα RNA-seq και η εσωτερική αξιολόγηση της ποιότητάς τους, η στοίχιση στο γονιδίωμα αναφοράς, η ποσοτικοποίηση των αλληλουχιών και η μετατροπή τους σε επίπεδα γονιδιακής έκφρασης, η ανάλυση διαφορικής έκφρασης και η ανάλυση εμπλουτισμού γονιδιακών συνόλων. Τα αποτελέσματα των παραπάνω αναλύσεων είναι διαθέσιμα στους χρήστες μέσω του γραφικού περιβάλλοντος και έχουν την επιλογή για τη λήψη τους σε διάφορες μορφές αρχείων (.csv, , .pdf, .png κ.α.). Επίσης όλα τα αποτελέσματα που παρέχονται από την εφαρμογή είναι άμεσα δημοσιεύσιμα. Ακόμη δίνεται η δυνατότητα στον χρήστη να επεξεργαστεί περεταίρω τα διαγράμματα που προκύπτουν βάσει του προσωπικού του γούστου, καθώς και να εφαρμόσει διάφορα φίλτρα που παρέχονται σε κάθε διεργασία ώστε να λάβει τα επιθυμητά στατιστικά αποτελέσματα βάσει των αναγκών της μελέτης του.

Η εφαρμογή στοχεύει στη διευκόλυνση βιολόγων που διαθέτουν το επιστημονικό υπόβαθρο για αναλύσεις RNA-seq , αλλά έχουν μηδαμινές ή ελάχιστες προγραμματιστικές γνώσεις. Η μείωση του κόστους αλληλούχησης έχει φέρει αυτού του είδους τις μεθοδολογίες στην ερευνητική καθημερινότητα, ωστόσο αρκετοί βιολόγοι είτε δε διαθέτουν τον απαραίτητο χρόνο είτε το κατάλληλο βιοπληροφορικό υπόβαθρο, με αποτέλεσμα να μην είναι σε θέση να αξιοποιήσουν τις νέες αυτές μεθοδολογίες. Έτσι, σε πολλές ερευνητικές ομάδες όπου υπάρχει έλλειψη αυτών των γνώσεων, είναι πολύτιμη η πρόσβαση σε ένα εύχρηστο εργαλείο που να πραγματοποιεί εύκολα και γρήγορά τις απαραίτητες αναλύσεις.

2. Μεθοδολογία

2.1. Αρχιτεκτονική εφαρμογής

2.1.1. Αρχιτεκτονική αντικειμένου διεπαφής χρήστη UI

Το αντικείμενο διεπαφής χρήστη αποτελείται από ένα ταμπλό (dashboard) το οποίο διαθέτει ένα μενού πλοήγησης στα αριστερά, μέσω του οποίου ο χρήστης μπορεί να μεταβεί σε κάθε καρτέλα για κάθε διεργασία.

Το πακέτο “shinydashboard” παρέχει ένα σύνολο συναρτήσεων που έχουν σχεδιαστεί για τη δημιουργία HTML, το οποίο δημιουργεί ένα ταμπλό (dashboard). Το “dashboardPage” διαθέτει τρία βασικά στοιχεία: το “dashboardHeader”, το “dashboardSidebar” και το “dashboardBody”. Οι περίπλοκες εφαρμογές χωρίζονται για κάθε χαρακτηριστικό του “dashboardPage” έτσι ώστε ο κώδικας να είναι πιο ευανάγνωστος. Έτσι λοιπόν έχουμε τις εξής συναρτήσεις για κάθε στοιχείο:

```
sidebar <- dashboardSidebar()

body <- dashboardBody()

ui <- dashboardPage(

  dashboardHeader(title = "Simplar"),

  sidebar,

  body

)
```

Το “dashboardHeader” περιέχει το όνομα της εφαρμογής και εμφανίζεται πάνω αριστερά. Το “dashboardSidebar” περιέχει το μενού (“sidebarMenu”) με τις καρτέλες της εφαρμογής και κάνοντας κλικ σε κάθε καρτέλα εμφανίζεται διαφορετικό περιεχόμενο στο σώμα του ταμπλό. Τα στοιχεία του μενού (“menuItem”) τοποθετούνται στην συνάρτηση “sidebarMenu()”. Κάθε στοιχείο του μενού συνδέεται με την αντίστοιχη καρτέλα (“tabItem”). Για να αντιστοιχίζεται ένα “menuItem” με ένα “tabItem”, πρέπει να έχουν ίδιες τιμές με το “tabName”. Τα “menuItems” έχουν ένα εικονίδιο επιλογής, το οποίο δημιουργείται από τη συνάρτηση “icon()” από το Shiny. Το σώμα “dashboardBody()” μιας σελίδας ταμπλό μπορεί να περιέχει οποιοδήποτε περιεχόμενο Shiny.

Το βασικό δομικό στοιχείο των περισσότερων ταμπλό είναι τα πλαίσια/κουτιά. Τα πλαίσια με τη σειρά τους μπορούν να περιέχουν οποιοδήποτε περιεχόμενο και δημιουργούνται με τη συνάρτηση `box()`. Επίσης, μπορούν να έχουν τίτλους, γραμμές τίτλου (title) και κατάστασης (status) με την ανάλογη μορφοποίηση, να έχουν συγκεκριμένο μέγεθος με το (width), καθώς και να μπορούν να συμπυκνωθούν με ένα κουμπί στο πάνω δεξί μέρος τους (collapsible=TRUE).

Όλα τα πλαίσια τοποθετούνται μέσα σε συναρτήσεις `fluidRow()` και συναρτήσεις `column()`. Αυτές οι δύο συναρτήσεις δημιουργούν μια διάταξη πλέγματος στο ταμπλό. Οι γραμμές δημιουργούνται από τη συνάρτηση `fluidRow()` και περιλαμβάνουν στήλες που ορίζονται από τη συνάρτηση `column()`. Η συνάρτηση `fluidRow()` χρησιμοποιείται συχνά για τη δημιουργία μιας οριζόντιας ομαδοποίησης στοιχείων, όπως κείμενο, στοιχεία ελέγχου εισόδου ή άλλα στοιχεία UI. Η `fluidRow()` προσαρμόζει αυτόματα το πλάτος των στοιχείων που περιέχει, καθιστώντας την χρήσιμη για τη δημιουργία διατάξεων που προσαρμόζονται σε διαφορετικά μεγέθη οθόνης. Η συνάρτηση `column()` χρησιμοποιείται για τη δημιουργία μιας στήλης μέσα σε μια `fluidRow()`. Επιτρέπει τον καθορισμό του πλάτους της στήλης, το οποίο μπορεί να εκφραστεί ως κλάσμα του πλάτους της `fluidRow()`. Η `column()` χρησιμοποιείται συχνά για τη δημιουργία ενός πλέγματος στοιχείων, όπως κείμενο, στοιχεία ελέγχου εισόδου ή διαγράμματα, όπου κάθε στήλη μπορεί να περιέχει ένα διαφορετικό στοιχείο. Η συνάρτηση `wellPanel()`, χρησιμοποιείται πολλές φορές μέσα στα πλαίσια για τη δημιουργία ενός πίνακα με περίγραμμα και ανοιχτό γκρι φόντο, χρησιμοποιείται συχνά για την οπτική ομαδοποίηση σχετικών στοιχείων, διευκολύνοντας τους χρήστες να κατανοήσουν τη σχέση μεταξύ των διαφόρων τμημάτων του UI. Τα περιεχόμενα της `wellPanel()` μπορούν να περιλαμβάνουν κείμενο, στοιχεία ελέγχου εισόδου ή άλλα στοιχεία του UI. Συνοπτικά, η `fluidRow()` και η `column()` χρησιμοποιούνται για τη δημιουργία ευέλικτων διατάξεων που προσαρμόζονται σε διαφορετικά μεγέθη οθόνης, ενώ η `wellPanel()` χρησιμοποιείται για την οπτική ομαδοποίηση σχετικών στοιχείων.

2.1.1.1. Βασικές Συναρτήσεις δια δραστικών εισόδων

`fileInput()`: Η συνάρτηση `fileInput()` επιτρέπει στους χρήστες να ανεβάζουν αρχεία από το τοπικό τους μηχάνημα στην εφαρμογή Shiny. Επιστρέφει μια λίστα με πληροφορίες σχετικά με το αρχείο που μεταφορτώθηκε, συμπεριλαμβανομένου του ονόματος του αρχείου, του τύπου περιεχομένου και του μεγέθους. Το ανεβασμένο αρχείο μπορεί να χρησιμοποιηθεί σε περαιτέρω επεξεργασία ή ανάλυση εντός της εφαρμογής Shiny.

`selectizeInput()`: Η συνάρτηση `selectizeInput()` δημιουργεί ένα αναπτυσσόμενο μενού με μια γραμμή αναζήτησης που επιτρέπει στους χρήστες να επιλέγουν εύκολα μία ή περισσότερες

επιλογές από μια λίστα. Οι επιλογές μπορούν να είναι προκαθορισμένες ή μπορούν να δημιουργηθούν δυναμικά με βάση τα δεδομένα της εφαρμογής. Η συνάρτηση `selectizeInput()` είναι ιδιαίτερα χρήσιμη όταν υπάρχει μεγάλος αριθμός επιλογών για επιλογή.

`selectInput()`: Η συνάρτηση `selectInput()` είναι παρόμοια με την `selectizeInput()`, αλλά δημιουργεί ένα απλούστερο αναπτυσσόμενο μενού χωρίς γραμμή αναζήτησης. Είναι χρήσιμη όταν υπάρχουν μόνο λίγες επιλογές για επιλογή.

`numericInput()`: Η συνάρτηση `numericInput()` δημιουργεί ένα πεδίο όπου οι χρήστες μπορούν να εισάγουν αριθμητικές τιμές. Χρησιμοποιείται συχνά για ρύθμιση παραμέτρων στην ανάλυση δεδομένων ή για να επιτρέπεται στους χρήστες να εισάγουν αριθμητικές τιμές για υπολογισμούς ή προσομοιώσεις στην εφαρμογή Shiny.

2.1.1.2. Κουμπιά

`actionButton`: Αποτελεί ένα διαδραστικό κουμπί και με ένα κλικ ο χρήστης μπορεί να ενεργοποιήσει κάποια διεργασία στην εφαρμογή.

`downloadButton`: Αποτελεί ένα διαδραστικό κουμπί και με ένα κλικ ο χρήστης μπορεί να κατέβασει δεδομένα από την εφαρμογή.

`downloadLink`: Δημιουργεί έναν υπερσύνδεσμο από τον οποίο είναι δυνατή η λήψη δεδομένων από τον χρήστη.

Τα `actionButton`, `downloadButton` και `downloadLink` είναι όλα χρήσιμα στοιχεία UI που επιτρέπουν στους χρήστες να ενεργοποιούν ενέργειες από την πλευρά του διακομιστή σε εφαρμογές R Shiny.

Οι παραπάνω συναρτήσεις βρίσκονται σε διάφορα σημεία του κώδικα, στο τμήμα που αντιστοιχεί στο UI. Παραδείγματος χάρη στις γραμμές κώδικα 300-355 και 501-602, ο οποίος παρουσιάζεται αναλυτικά στο παράρτημα.

2.1.2. Ειδική μορφοποίηση του UI

Ακόμη έχει πραγματοποιηθεί ειδική μορφοποίηση του απλού “dashboard” του Shiny. Δηλαδή έχει γίνει προσαρμογή των χρωμάτων και των σκιάσεων του ταμπλό, των πλαισίων των κουμπιών και των πινάκων, καθώς και στο χρώμα του “background”. Επιπλέον έχει προσαρμοστεί το χρώμα και το μέγεθος των ειδοποιήσεων καθώς και έχει προστεθεί ένα

“GIF”, το οποίο εμφανίζεται κάθε φορά που τρέχει μία διεργασία ο διακομιστής της εφαρμογής. Όταν ο διακομιστής είναι απασχολημένος με τη λειτουργία μιας διεργασίας το γραφικό περιβάλλον της εφαρμογής γίνεται πιο σκούρο, εμφανίζεται το συγκεκριμένο “GIF” και «κλειδώνει» στην καρτέλα που βρίσκεται ο χρήστης μέχρι το τέλος της διεργασίας. Επιπλέον έχει προστεθεί ένα ειδικά προσαρμοσμένο “logo” για την εφαρμογή, ως επικεφαλίδα στο “dashboard”. Αυτές οι τροποποιήσεις έχουν γίνει με τη βοήθεια των συναρτήσεων του πακέτου “dashboardthemes⁶” καθώς και με την προσθήκη κώδικα CSS. Κάποιες από αυτές τις τροποποιήσεις παρουσιάζονται στον κώδικα, στις γραμμές 703-825 και στις γραμμές 41-65.

2.1.3. Αρχιτεκτονική συνάρτησης διακομιστή server

Η συνάρτηση διακομιστή χρησιμοποιείται για τον ορισμό της λογικής της εφαρμογής από την πλευρά του διακομιστή. Η συνάρτηση διακομιστή περιέχει συνήθως διαδραστικές εκφράσεις “reactive expressions”, συναρτήσεις δέσμευσης εισόδου/εξόδου και άλλο κώδικα από την πλευρά του διακομιστή που εκτελείται ανταποκρινόμενος στην είσοδο του χρήστη.

server <- function(input, output, session) {}

Το όρισμα “input” είναι ένας κατάλογος τιμών εισόδου που έχει δώσει ο χρήστης, όπως εισόδοι κειμένου μενού, ρυθμιστικά (slider) και άλλα. Αυτές οι τιμές εισόδου προσπελάζονται συνήθως με τη χρήση του τελεστή “\$” (π.χ. “input\$textInput”).

Το όρισμα “output” είναι μια λίστα με αντικείμενα εξόδου που παράγει ο διακομιστής, όπως γραφήματα, πίνακες ή κείμενο. Σε κάθε αντικείμενο εξόδου εκχωρείται ένα όνομα, το οποίο χρησιμοποιείται για την αναφορά του στο περιβάλλον εργασίας. Τα αντικείμενα εξόδου συνήθως παράγονται χρησιμοποιώντας μία από τις συναρτήσεις “render”, όπως η “renderPlot()” ή η “renderTable()”.

Το όρισμα session παρέχει πρόσβαση στην τρέχουσα συνεδρία (session) και επιτρέπει την τροποποίηση της συμπεριφοράς της εφαρμογής με διάφορους τρόπους, όπως η ανακατεύθυνση του χρήστη σε μια διαφορετική διεύθυνση URL ή η αποσύνδεση ενός χρήστη από την εφαρμογή.

Η συνάρτηση διακομιστή είναι ένα βασικό συστατικό των εφαρμογών R Shiny, επιτρέποντας στους προγραμματιστές να δημιουργούν, δυναμικές και διαδραστικές διεπαφές χρήστη που ανταποκρίνονται στην είσοδο του χρήστη σε πραγματικό χρόνο.

2.1.3.1. Βασικές συναρτήσεις για τη κατασκευή διαδραστικών αντικειμένων

Για τη δημιουργία διαδραστικών αποτελεσμάτων σε μία εφαρμογή shiny χρησιμοποιούνται οι συναρτήσεις απόδοσης “render({ })” καθώς και οι συναρτήσεις έκφρασης reactive({ }), που δημιουργούν διαδραστικές εκφράσεις “reactive expressions”. Οι διαδραστικές εκφράσεις είναι λίγο πιο «έξυπνες» από τις κανονικές συναρτήσεις της R. Αποθηκεύουν τις τιμές τους και γνωρίζουν πότε οι τιμές τους έχουν ενημερωθεί. Την πρώτη φορά που εκτελείται μια διαδραστική έκφραση το αποτέλεσμά της αποθηκεύεται στη μνήμη του υπολογιστή. Την επόμενη φορά που καλείται η ίδια έκφραση, ελέγχει αν το αποθηκευμένο αποτέλεσμα είναι ενημερωμένο και στη συνέχεια το επιστρέφει χωρίς να κάνει κανέναν υπολογισμό (πράγμα που κάνει την εφαρμογή πιο γρήγορη). Σε περίπτωση που το αποτέλεσμα είναι παρωχημένο, η έκφραση θα υπολογίσει εκ νέου το αποτέλεσμα. Στη συνέχεια επιστρέφει το νέο αποτέλεσμα και αποθηκεύει ένα νέο αντίγραφο. Με αυτό τον τρόπο αποφεύγεται η άσκοπη επανεκτέλεση κώδικα.

Οι συναρτήσεις απόδοσης “render” λειτουργούν περίπου με τον ίδιο τρόπο. Για τη δημιουργία διαδραστικών αντικειμένων οι render συναρτήσεις χρειάζονται τιμές εισόδου “input” και μια συνάρτηση εξόδου “output” στο αντικείμενο διεπαφής (UI). Ο κώδικας εντός της συνάρτησης απόδοσης είναι εκεί όπου παράγεται η δυναμική έξοδος με βάση την είσοδο του χρήστη. Αυτός ο κώδικας μπορεί να έχει πρόσβαση στις τιμές εισόδου χρησιμοποιώντας το αντικείμενο “input” (π.χ. input\$input_name), το οποίο περιέχει τις τρέχουσες τιμές όλων των στοιχείων εισόδου στην εφαρμογή Shiny.

Πίνακας 1. < Κάθε συνάρτηση render και output δημιουργεί διαφορετικό τύπο αποτελέσματος. Στον παρακάτω πίνακα παρουσιάζονται οι διάφορες τέτοιες συναρτήσεις που χρησιμοποιούνται στον κώδικα, καθώς και τα αντίστοιχα αποτελέσματά τους. >

output function	Αποτέλεσμα	render function	Αποτέλεσμα
dataTableOutput	DataTable	renderDataTable	DataTable
imageOutput	image	renderImage	image
plotOutput	plot	renderPlot	plot
verbatimTextOutput	text	renderPrint	printed output
tableOutput	table	renderTable	data frame, matrix
textOutput	text	renderText	character strings

Η συνάρτηση “observe({})” χρησιμοποιείται για την παρακολούθηση των αλλαγών που πραγματοποιούνται σε μια διαδραστική τιμή ή έκφραση και την ενημέρωση της αντίστοιχης εξόδου ή και εξόδων, σε πραγματικό χρόνο. Η συνάρτηση δέχεται ως όρισμα μια διαδραστική έκφραση ή τιμή και στη συνέχεια εκτελεί τον κώδικα μέσα στις αγκύλες κάθε φορά που η τιμή ή η έκφραση αλλάζει.

Οι συναρτήσεις ενημέρωσης “update functions()”, χρησιμοποιούνται για την ενημέρωση των ιδιοτήτων και των τιμών των στοιχείων διεπαφής χρήστη (UI) ή των εξόδων με βάση ορισμένες συνθήκες ή αλληλεπιδράσεις του χρήστη. Αυτές οι συναρτήσεις είναι απαραίτητες για τη δημιουργία δυναμικών εφαρμογών Shiny. Ακολουθούν ορισμένες βασικές συναρτήσεις ενημέρωσης στο Shiny:

updateTextInput: Αυτή η συνάρτηση χρησιμοποιείται για την ενημέρωση της τιμής ενός στοιχείου εισόδου κειμένου ως απόκριση σε ένα συμβάν χρήστη η συνάρτηση μπορεί να χρησιμοποιηθεί για ενημέρωση της τιμής ενός πεδίου εισαγωγής κειμένου με βάση την επιλογή που γίνεται σε ένα αναπτυσσόμενο μενού.

updateSelectInput: Αυτή η συνάρτηση χρησιμοποιείται για την ενημέρωση των επιλογών και της επιλογής ενός στοιχείου εισόδου «select». Για παράδειγμα, η συνάρτηση μπορεί να χρησιμοποιηθεί για την ενημέρωση των διαθέσιμων επιλογών σε ένα μενού με βάση την τιμή ενός άλλου πεδίου εισόδου.

updateSliderInput: Αυτή η συνάρτηση χρησιμοποιείται για την ενημέρωση της τιμής και των ιδιοτήτων ενός στοιχείου εισόδου «slider». Για παράδειγμα, η συνάρτηση μπορεί να

χρησιμοποιηθεί για να την ενημέρωση των ελάχιστων και μέγιστων τιμών ενός ρυθμιστή με βάση την αλληλεπίδραση του χρήστη.

`updateCheckboxInput`: Αυτή η συνάρτηση χρησιμοποιείται για την ενημέρωση της κατάστασης ενός στοιχείου εισόδου «checkbox». Η συνάρτηση χρησιμοποιείται για να επιλέξει ή να αποεπιλέξει ο χρήστης ένα πλαίσιο ελέγχου με βάση την αλληλεπίδραση του χρήστη.

`updateOutput`: Αυτή η συνάρτηση χρησιμοποιείται για την ενημέρωση της εξόδου μιας εφαρμογής Shiny. Για παράδειγμα, η συνάρτηση μπορεί να χρησιμοποιηθεί για την ενημέρωση των περιεχομένων ενός διαγράμματος ή πίνακα με βάση την αλληλεπίδραση του χρήστη.

2.2. Μεθοδολογία διεργασιών εφαρμογής

2.2.1. Αντιγραφή αρχείων

Αρχικά γίνεται αντιγραφή των ακατέργαστων αρχείων τύπου “.fastq” του χρήστη, στον υπό φάκελο “adrm” του εσωτερικού φακέλου “extdata” του πακέτου “Rbowtie2”⁷. Αυτό το βήμα είναι απαραίτητο καθώς η συνάρτηση “remove_adapters()” του “Rbowtie2”, η οποία αφαιρεί τους προσαρμογείς, διαβάζει μόνο αρχεία που βρίσκονται μέσα στον φάκελο “adrm”. Ο χρήστης για κάθε αρχείο που φορτώνει στην εφαρμογή, μπορεί να δει σε ένα πλαίσιο τις εξής πληροφορίες: το όνομα και το μέγεθός του αρχείου σε Bytes και την τοποθεσία στη οποία αποθηκεύτηκε δηλαδή τη διαδρομή για το “Rbowtie2/extdata/adrm”. Επίσης εμφανίζονται τα ονόματα των αρχείων που βρίσκονται μέσα στον φάκελο “adrm” και κάθε φορά που φορτώνεται κάποιο νέο αρχείο η λίστα ανανεώνεται. Όταν δεν υπάρχει κάποιο αρχείο στον φάκελο “adrm”, εμφανίζεται το εξής μήνυμα στο πλαίσιο: “NOT FILE” και “character (0)”. Ακολουθεί απόσπασμα του κώδικα που αντιστοιχεί στη διαδικασία που μόλις περιεγράφηκε.

```
#path to Rbowtie2/extdata/adrm
```

```
destDir <- system.file("extdata", "adrm", package = "Rbowtie2")
```

```
output$result <- renderPrint({
```

```
  inFile <- input$file1
```

```
  if (is.null(inFile)) {
```

```

cat("NOT FILE\n")

#Update the files inside adrm and render the names in the textoutput

result <- list.files(system.file("extdata", "adrm", package="Rbowtie2"),
                     pattern="*.fastq$")

updateForm()

return(result) }

cat("Reading file:", inFile$name, "\n")

cat("size:", inFile$size, " Bytes, type:", inFile$type, "\n")

if (dir.exists(destDir)){

  cat("Copying file to:", destDir, "\n")

  result <- file.copy( inFile$datapath,
                      file.path(destDir, inFile$name) )

  result<- list.files(system.file("extdata", "adrm", package="Rbowtie2"),
                     pattern="*.fastq$")

} else {

  result <- FALSE }

updateForm()

result })

```

Τα ονόματα των αρχείων ανανεώνονται κάθε φορά στα αντίστοιχα πλαίσια του “UI” για τις πρωτότυπες “forward” και τις συμπληρωματικές “reverse” αλληλουχίες. Από αυτά τα πλαίσια, ο χρήστης μπορεί να επιλέξει τα ζεύγη των αλληλουχιών που θέλει κάθε φορά να φιλτράρει. Ακολουθεί απόσπασμα του κώδικα που αντιστοιχεί στη διαδικασία που μόλις περιεγράφηκε.

#update the file lists (forward and reverse) in the trimming box in ui

```
updateForm = function()
```

```

{ isolate({

  groupv = list.files(system.file(package = "Rbowtie2","extdata","adrm"),pattern =
"R1.*.fastq$" )

  groupv2 = list.files(system.file(package = "Rbowtie2","extdata","adrm"),pattern =
"R2.*.fastq$" )

  updateSelectizeInput(session, "pair1", choices = groupv, selected =
groupv[1],server=TRUE)

  updateSelectizeInput(session, "pair2", choices = groupv2, selected =
groupv2[1],server=TRUE) }) }

```

2.2.2. Φιλτράρισμα (Trimming)

Για την προ-επεξεργασία και τον καθαρισμό των ακατέργαστων RNA-seq δεδομένων χρησιμοποιείται το πακέτο “Rbowtie2”, το οποίο περιέχει το “AdapterRemoval”, ένα εύχρηστο εργαλείο για την προ επεξεργασία ακατέργαστων δεδομένων αλληλούχισης.

Ειδικότερα γίνεται ταχεία αναγνώριση και περικοπή προσαρμογών, χρησιμοποιώντας μια τροποποιημένη έκδοση του αλγορίθμου “Needleman-Wunsch”, προκειμένου να πραγματοποιηθεί στοίχιση των αλληλουχιών ανά ζεύγος αναγνωσμάτων όπως και των αλληλουχιών προσαρμογέα. Η συνάρτηση “remove_adapters()” του “AdapterRemoval” αφαιρεί τους προσαρμογείς. Οι αλληλουχίες των προσαρμογέων που αναγνωρίζονται και αφαιρούνται είναι της βιβλιοθήκης “Truseq” της “Illumina”. Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η διαδικασία του φιλτραρίσματος καθώς και η συνάρτηση “remove_adapters()” και οι παράμετροί της.

```

observe({

  Rbowtie2Reactive()

})

td<-getwd()

Rbowtie2Reactive <- eventReactive(input$initRbowtie,{

  withProgress(message = "Trimming , please wait",{

```

```

pair1<-input$pair1

pair2<-input$pair2

reads_1 <- system.file(package = "Rbowtie2","extdata","adrm" , pair1)

reads_2 <-system.file(package = "Rbowtie2","extdata","adrm" , pair2)

(cmdout<-remove_adapters(

file1=reads_1,

file2=reads_2,

adapter1="TACACTCTTCCCTACACGACGCTCTCCGATCT",

adapter2="GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT",

basename=file.path(td, "trimmed"),

overwrite=TRUE,

"--threads 3"))

```

file1: Αντιστοιχεί στο όνομα του αρχείου με τις πρωτότυπες “forward” αλληλουχίες.

file2: Αντιστοιχεί στο όνομα του αρχείου με τις συμπληρωματικές “reverse” αλληλουχίες.

adapter1: Συμβολοσειρά χαρακτήρων που αντιστοιχεί στη “forward” αλληλουχία προσαρμογέα.

adapter2: Συμβολοσειρά χαρακτήρων που αντιστοιχεί στη “reverse” αλληλουχία προσαρμογέα.

basename: Αντιστοιχεί στο όνομα των παραγόμενων φιλτραρισμένων αρχείων.

overwrite: Λογική ένδειξη για το αν πρέπει να γίνει αντικατάσταση των παραγόμενων αρχείων.

threads: Αντιστοιχεί στον αριθμό των νημάτων “threads” που δεσμεύονται για την διαδικασία του φιλτραρίσματος.

Στη συνέχεια γίνεται μετονομασία των “trimmed” αρχείων που προκύπτουν. Αυτό συμβαίνει γιατί η συνάρτηση ονομάζει αυτόματα τις πρωτότυπες φιλτραρισμένες αλληλουχίες (forward reads) ως “trimmed.pair1.truncated” και αντίστοιχα τις συμπληρωματικές (reverse reads) ως “trimmed.pair2.truncated”. Έτσι τα αρχεία μετονομάζονται με βάση το αρχικό τους

όνομα , έχοντας στο τέλος την χαρακτηριστική κατάληξη “_trimmed.fastq”», για την αποφυγή της εγγραφής των επόμενων αποτελεσμάτων στα ίδια αρχεία. Ακολουθεί απόσπασμα του κώδικα για τη διαδικασία που μόλις περιεγράφηκε.

#path to "trimmed.pair1.truncated" file

```
file1<-file.path(td, "trimmed.pair1.truncated")
```

#rename pair1 from "trimmed.pair1.truncated" to "original_file_name_trimmed.fastq"

```
file.rename(file1,(sub("*.fastq$", "_trimmed.fastq", pair1)))
```

#path to "trimmed.pair2.truncated" file

```
file2<-file.path(td, "trimmed.pair2.truncated")
```

#rename pair2 from "trimmed.pair2.truncated" to "original_file_name_trimmed.fastq"

```
file.rename(file2,(sub("*.fastq$", "_trimmed.fastq", pair2)))
```

#path to "trimmed.settings" file

```
file3<-file.path(td, "trimmed.settings")
```

```
file.rename(file3,(sub("*.fastq$", "_trimmed.settings", pair1)))
```

Επιπλέον, εκτός από τα φιλτραρισμένα τύπου “.fastq” αρχεία προκύπτει και ένα αρχείο “SETTINGS” με τα στατιστικά και τις ρυθμίσεις της διαδικασίας του φιλτραρίσματος για κάθε δείγμα.

Τέλος, δίνεται η επιλογή στον χρήστη να διαγράψει τα αρχεία από τον υπό φάκελο “adrm” μετά το τέλος της διαδικασίας του φιλτραρίσματος, ώστε να μην καταλαμβάνουν αποθηκευτικό χώρο.

```
observe({
```

```
  delete_files() })
```

#delete all the files from extdata\adrm after you are done with the trimming process

```
delete_files <- eventReactive(input$delete,{
```

```
  files_to_delete <- dir(path=destDir ,pattern=".fastq")
```

```

file.remove(file.path(destDir, files_to_delete))

showNotification("Files are deleted!", duration = 5,

               type = ( "message")) })

```

2.2.3. Κατασκευή Ευρετηρίου “Index”

Αρχικά πριν γίνει η στοίχιση πρέπει να κατασκευαστεί ένα ευρετήριο “index” βάσει του γονιδιώματος αναφοράς το οποίο δίνει ο χρήστης, αυτό γίνεται με τη συνάρτηση “buildindex()” του πακέτου “Rsubread”⁸.

Το γονιδίωμα αναφοράς μπορεί να είναι ένα αρχείο μορφής “.fasta”, “.fa” ή “.fna” και να βρίσκεται μέσα στον προσωρινό κατάλογο (working directory). Ο χρήστης μπορεί να επιλέξει από το ανάλογο πλαίσιο στο “UI” το όνομα του αρχείου που θέλει να χρησιμοποιήσει.

Το ευρετήριο αποθηκεύεται στον προσωρινό κατάλογο αποθήκευσης (working directory), με δυνατότητα να χρησιμοποιηθεί και σε επόμενες αναλύσεις του ίδιου οργανισμού. Ένα μοναδικό χαρακτηριστικό του “Rsubread” είναι η δυνατότητα ορισμού/επιλογής του μεγέθους της μνήμης(σε MB) που χρησιμοποιείται στην διαδικασία της κατασκευής ευρετηρίου. Επιπλέον η κατασκευή ενός διακοπτόμενου ευρετηρίου “gapped index” σε συνδυασμό με το “indexSplit” καθιστούν αρκετά πιο γρήγορη τη διαδικασία και συνιστώνται για χρήση σε προσωπικό υπολογιστή. Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η διαδικασία κατασκευής ευρετηρίου καθώς και η συνάρτηση “ buildindex()” και οι παράμετροί της.

```

observe({ BuildindexReactive() })

#list reference genome file (.fastq/.fa/.fna) file from the working directory

REF_GENOME <- list.files(td, pattern = "*.fasta$|*.fa$|*.fna$" )

updateSelectizeInput(session, "reference_fasta", choices = REF_GENOME, selected =
REF_GENOME[1],server=TRUE)

BuildindexReactive <- eventReactive(input$initBuild,{

withProgress(message = "Building index , please wait",{

#define the basename for the index

```

```
RSUBREAD_INDEX_BASE <- "genome"
```

```
setProgress(value = 0.3, detail = "This might take a while, please wait ...")
```

```
#build gapped and splitted index to save storage
```

```
buildindex(
```

```
  basename=file.path(td,RSUBREAD_INDEX_BASE),
```

```
  reference=REF_GENOME,
```

```
  gappedIndex=TRUE,
```

```
  indexSplit=TRUE,
```

```
  memory=4000)
```

basename: Αντιστοιχεί στο όνομα του παραγόμενου "Index".

reference: Αντιστοιχεί στο γονιδίωμα αναφοράς.

gappedIndex: Λογική ένδειξη για το αν πρέπει να δημιουργηθεί ένα ευρετήριο με κενά ή ένα πλήρες ευρετήριο.

indexSplit: Λογική ένδειξη για το αν το ευρετήριο μπορεί να χωριστεί σε πολλαπλά μπλοκ. Το μέγεθος του μπλοκ καθορίζεται από την τιμή του "memory".

memory: Αριθμητική τιμή που καθορίζει το μέγεθος της μνήμης (σε megabytes) που χρησιμοποιείται για την αποθήκευση του ευρετηρίου. Αυτή η επιλογή αγνοείται όταν το "indexSplitis=FALSE".

2.2.4. Στοίχιση (Alignment)

Το πακέτο "Rsubread"⁸ χρησιμοποιείται για διαδικασία της στοίχισης και συγκεκριμένα η συνάρτηση "align()".

Για να γίνει γρηγορότερα η διαδικασία της στοίχισης, προτείνεται η έξοδος να οριστεί ως τύπος αρχείων σε μορφή ".BAM" (output_format= .BAM) , καθώς τα αρχεία ".BAM" είναι μικρότερα σε μέγεθος από τα ".SAM", τα οποία αποτελούν εναλλακτικό τύπο αρχείων εξόδου της διαδικασίας. Επίσης καλό θα ήταν να χρησιμοποιηθεί ο κατάλληλος αριθμός threads για

να επιτευχθεί ο βέλτιστος δυνατός χρόνος υλοποίησης της διαδικασίας με βάση τους επεξεργαστικούς πυρήνες (CPU) του υπολογιστή κάθε χρήστη. Στους περισσότερους επεξεργαστές σε κάθε πυρήνα αντιστοιχούν 2 λογικοί επεξεργαστές, “threads”, οπότε αν μια CPU έχει 16 πυρήνες μπορούν να δεσμευτούν μέχρι και 24 “threads”. Ο μέγιστος αριθμός threads που μπορεί να χρησιμοποιηθεί από τη συνάρτηση είναι 40. Όλα τα αρχεία μορφής “fastq” που βρίσκονται στον προσωρινό κατάλογο (working directory) θα στοιχηθούν αυτόματα ανά ζεύγη, δηλαδή δεν απαιτείται από τον χρήστη να φορτώσει τα αρχεία στην εφαρμογή. Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η διαδικασία της στοίχισης καθώς και η συνάρτηση “align()” και οι παράμετροί της.

```
observe({  
  
  MappingReactive() })  
  
MappingReactive <- eventReactive(input$initAlign,{  
  
  withProgress(message = "Mapping , please wait",{  
  
    # define the basename for the index  
  
    RSUBREAD_INDEX_BASE <- "genome"  
  
    #list trimmed forward reads  
  
    inputfilefwd <- list.files(td, pattern = "_R1_*.fastq$" )  
  
    #list trimmed reverse reads  
  
    inputfilervs <- list.files(td, pattern = "_R2_*.fastq$" )  
  
    setProgress(value = 0.3, detail = "This might take a while, please wait ...")  
  
    align(  
  
index=file.path(td,RSUBREAD_INDEX_BASE),  
  
minFragLength=input$minFrag,  
  
maxFragLength = input$maxFrag,  
  
maxMismatches=input$misMatch,
```

```

nthreads=input$nthreads,

readfile1=inputfilefwd,

readfile2=inputfilervs,

output_format=input$format)

showNotification("Mapping completed!", duration = NULL,

type = ( "message")) }

```

index: Αντιστοιχεί στο αρχείο του ευρετηρίου.

minFragLength: Αριθμητική τιμή που δίνει το ελάχιστο μήκος θραύσματος (50 από προεπιλογή).

maxFragLength: Αριθμητική τιμή που δίνει το μέγιστο μήκος θραύσματος (600 από προεπιλογή).

maxMismatches: Αριθμητική τιμή που δίνει τον μέγιστο αριθμό αταίριαστων βάσεων που επιτρέπονται στη στοίχιση (1 από προεπιλογή). Οι εσφαλμένες αντιστοιχίσεις που εντοπίζονται σε βάσεις δεν υπολογίζονται.

nthreads: Αριθμητική τιμή που δίνει τον αριθμό των «threads» που δεσμεύονται για την διαδικασία της στοίχισης (12 από προεπιλογή).

readfile1: Αντιστοιχεί στα αρχεία με τις “forward” φιλτραρισμένες αλληλουχίες.

readfile2: Αντιστοιχεί στα αρχεία με τις “reverse” φιλτραρισμένες αλληλουχίες.

output_format: Αντιστοιχεί στον τύπο αρχείου του αποτελέσματος “.BAM/.SAM”

Εκτός από τα αρχεία της μορφής “.BAM/.SAM” με τα στοιχισμένα αναγνώσματα, προκύπτει επιπλέον ένα αρχείο “SUMMARY” με τα στατιστικά αποτελέσματα της στοίχισης για κάθε δείγμα, καθώς και ένα αρχείο “VCF” με τις γενετικές παραλλαγές για κάθε δείγμα.

2.2.5. Ποσοτικοποίηση (Quantification)

Το πακέτο Rsubread⁸ χρησιμοποιείται και για διαδικασία της ποσοτικοποίησης, καθώς διαθέτει ένα εύχρηστο εργαλείο, το featureCounts, κατάλληλο για ακριβή καταμέτρηση αναγνωσμάτων.

Το παραπάνω, δέχεται ως είσοδο στοιχισμένες αναγνώσεις σε μορφή αρχείων “.SAM” ή “.BAM”, καθώς και γονιδιώματα αναφοράς (annotated reference genome) σε μορφή “.gtf”. Το γονιδίωμα αναφοράς πρέπει να βρίσκεται μέσα στον προσωρινό κατάλογο (working directory). Ο χρήστης μπορεί να επιλέξει από το ανάλογο πλαίσιο στο “UI” το όνομα του αρχείου που θέλει να χρησιμοποιήσει. Όλα τα αρχεία μορφής “.SAM” ή “.BAM” που βρίσκονται στον προσωρινό κατάλογο (working directory) θα καταμετρηθούν αυτόματα, δηλαδή δεν απαιτείται από τον χρήστη να φορτώσει τα αρχεία στην εφαρμογή. Το “featureCounts” παρέχει πολλαπλές επιλογές για την καταμέτρηση αναγνώσεων πολλαπλής χαρτογράφησης (multi-mapping reads), δηλαδή αναγνώσεις που αντιστοιχίζονται σε περισσότερες από μία θέση στο γονιδίωμα αναφοράς. Επίσης παρέχει επιλογές για την καταμέτρηση αναγνώσεων πολλαπλών επικαλύψεων (allowMultiOverlap), δηλαδή αναγνώσεις που αντιστοιχίζονται σε περισσότερες από μία θέσεις στο γονιδίωμα αναφοράς. Ακόμη όταν ισχύει “isPairedEnd=TRUE”, η συνάρτηση ελέγχει αν οι αναγνώσεις από το ίδιο ζεύγος είναι γειτονικές μεταξύ τους (αυτό μπορεί να συμβεί όταν οι αναγνώσεις ταξινομήθηκαν, για παράδειγμα, με βάση τις θέσεις αντιστοίχισής τους), και αναδιατάσσει αυτόματα τις αναγνώσεις που ανήκουν στο ίδιο ζεύγος αλλά δεν είναι γειτονικές μεταξύ τους. Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η συνάρτηση “featureCounts()” καθώς και οι παράμετροι που χρησιμοποιούνται στον κώδικα της εφαρμογής:

```
gtf <- list.files(td, pattern = "*.gtf$" )

updateSelectizeInput(session, "reference_gtf", choices = gtf, selected = gtf[1],server=TRUE)

QuantifyReactive <- eventReactive(input$initQuantify,{

  withProgress(message = "Counting , please wait",{

    #list reference genome files (.gtf) from working directory

    #genome_name <- list.files(td, pattern = "*.gtf$" )

    #list the .SAM or .BAM files from the working directory

    inputfile <- list.files(td, pattern = "*.SAM$|*.BAM$" )

    mycounts<-featureCounts(

    inputfile,
```

```

annot.ext= genome_name,

isGTFAnnotationFile=TRUE,

isPairedEnd=TRUE,

allowMultiOverlap=input$overlap,

minOverlap=input$minOverlap,

countMultiMappingReads=input$countMultiMappingReads,

minFragLength = input$minFragL,

maxFragLength = input$maxFragL,

nthreads=input$nthreads)

  mycounts$count
  return(list('counts'=mycounts$count))

}) })

```

inputfile: Αντιστοιχεί στα αρχεία με τις στοιχισμένες αναγνώσεις.

annot.ext: Δηλώνει ότι το γονιδίωμα αναφοράς (annotated reference genome) παρέχεται από τον χρήστη.

isGTFAnnotationFile: Λογική ένδειξη, όταν είναι “TRUE” η μορφή του αρχείου (annotation file) είναι “.gtf”

isPairedEnd: Λογική ένδειξη, όταν είναι “TRUE” καταμετρούνται ζεύγη αναγνώσεων (θραύσματα).

allowMultiOverlap: Οι χρήστες μπορούν να επιλέξουν κάθε αλληλεπικαλυπτόμενο χαρακτηριστικό θέτοντας το σε “TRUE” ή να αγνοήσουν τέτοιες αναγνώσεις θέτοντας το σε “FALSE”.

minOverlap: Ακέραιος αριθμός που δίνει τον ελάχιστο αριθμό επικαλυπτόμενων βάσεων που απαιτείται για την ανάθεση μιας ανάγνωσης σε ένα χαρακτηριστικό. Για την ανάθεση ζευγών αναγνώσεων (θραυσμάτων), ο αριθμός των επικαλυπτόμενων βάσεων από κάθε ανάγνωση στο ίδιο ζεύγος θα αθροίζεται.

countMultiMappingReads: Οι χρήστες μπορούν να επιλέξουν να μετρήσουν κάθε στοίχιση που αναφέρεται για ένα “multi-mapping” ή να αγνοήσουν τέτοιες αναγνώσεις θέτοντας το σε FALSE.

minFragLength: Αριθμητική τιμή που δίνει το ελάχιστο μήκος θραύσματος (50 από προεπιλογή).

maxFragLength: Αριθμητική τιμή που δίνει το μέγιστο μήκος θραύσματος (600 από προεπιλογή).

nthreads: Αριθμητική τιμή που δίνει τον αριθμό των “threads” που δεσμεύονται για την διαδικασία της στοίχισης (12 από προεπιλογή). Για τον ορισμό του κατάλληλου αριθμού threads ισχύει ότι και στη διαδικασία της στοίχισης.

Ο πίνακας των αποτελεσμάτων μπορεί να αποθηκευτεί σε ένα αρχείο μορφή “.csv” οπουδήποτε θελήσει ο χρήστης. Επίσης από τα ονόματα των δειγμάτων στις στήλες του πίνακα αφαιρούνται ορισμένοι χαρακτήρες αν εντοπιστούν (“_R1_trimmed.fastq.subread.SAM/ “_R1_trimmed..fastq.subread.BAM”), ώστε να είναι πιο ευανάγνωστος ο πίνακας.

2.2.6. Ανάλυση διαφορικής έκφρασης (Differential expression)

Για την ανάλυση διαφορικής έκφρασης χρησιμοποιείται το πακέτο “DESeq2”⁹. Αρχικά διαβάζεται το αρχείο με τα ποσοτικοποιημένα αναγνώσματα (counts)), το οποίο βρίσκεται σε μορφή “.csv”, και στη συνέχεια μετατρέπεται σε πίνακα (matrix) καθώς αυτή είναι η επιθυμητή μορφή των συγκεκριμένων δεδομένων στη συνάρτηση “DESeqDataSetFromMatrix”. Στη συνέχεια, διαβάζεται το αρχείο μορφής “.csv” που περιγράφει τις συνθήκες των δειγμάτων (phenodata). Το αρχείο με τις συνθήκες των δειγμάτων (phenodata) επεξεργάζεται πριν χρησιμοποιηθεί στη συνάρτηση της “DESeq2”. Αρχικά απομονώνονται τα ονόματα των στηλών του “dataframe” που δημιουργείται από το αρχείο που φορτώνει ο χρήστης και στη συνέχεια απομονώνεται το περιεχόμενο των στηλών. Αυτά στη πορεία αντιστοιχίζονται για τη δημιουργία ενός πίνακα που περιέχει τις κατάλληλες πληροφορίες για τη συνάρτηση της “DESeq2”. Το απόσπασμα του κώδικα που αφορά τη συγκεκριμένη διαδικασία βρίσκεται στις γραμμές 1110-1152. Η φόρμουλα σχεδίασης του αντικειμένου της “DESeq2” ανανεώνεται βάσει των μεταβλητών του phenodata ανάλογα με

το αν υπάρχει μία ή περισσότερες συνθήκες. Το απόσπασμα του κώδικα που αφορά τη συγκεκριμένη διαδικασία βρίσκεται στις γραμμές 1155-1178.

Η συνάρτηση “DESeqDataSetFromMatrix” κατασκευάζει ένα αντικείμενο “DESeqDataSet” βάσει των δύο αυτών αρχείων και της φόρμουλας σχεδίασης. Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η συνάρτηση “DESeqDataSetFromMatrix()” καθώς και οι παράμετροι που χρησιμοποιούνται στον κώδικα της εφαρμογής:

```
ResultsReactive <- eventReactive(input$initResults,{  
  
  withProgress(message = "Processing , please wait", {  
  
    df2<-df_products()  
  
    #remove the NA values from the counts dataframe  
  
    df2<-na.omit(df2)  
  
    #convert the counts dataframe to a data matrix  
  
    countDataMatrix <- as.matrix(df2[ , -1])  
  
    rownames(countDataMatrix) <- df2[ , 1]  
  
samples <- myValues$DF  
  
dds <-DESeqDataSetFromMatrix(  
  
countData=countDataMatrix,  
  
colData=samples,  
  
design=as.formula(input$designFormula))
```

countData: Αντιστοιχεί στον πίνακα με τα counts (ο πίνακας δε γίνεται να περιέχει αρνητικές μεταβλητές ούτε κενές μεταβλητές (missing values, NA)).

colData: Αντιστοιχεί στο “dataframe” του “phenodata”.

design: Η φόρμουλα σχεδιασμού εκφράζει τον τρόπο με τον οποίο οι μετρήσεις για κάθε γονίδιο εξαρτώνται από τις μεταβλητές του “colData”. Η φόρμουλα μπορεί να περιέχει μία ή περισσότερες μεταβλητές(συνθήκες).

DESeq(dds)

DESeq(): Αυτή η λειτουργία εκτελεί μια ανάλυση βάσει των εξής βημάτων:

1. Εκτίμηση των παραγόντων μεγέθους: estimateSizeFactors.
2. Εκτίμηση της διασποράς: estimateDispersions.
3. Στατιστικά στοιχεία Wald: nbinomWaldTest.

Η συνάρτηση “DESeq” επιστρέφει ένα αντικείμενο “DESeqDataSet”, οι πίνακες αποτελεσμάτων (log2 fold changes, padjust και p-values) υλοποιούνται χρησιμοποιώντας τη συνάρτηση “results”.

Για να υποδείξουμε στο “DESeq2” τις δύο ομάδες που θέλουμε να συγκρίνουμε, μπορούμε να χρησιμοποιήσουμε τις αντιθέσεις. Οι αντιθέσεις παρέχονται στη συνέχεια στο “DESeq2” για να εκτελέσει έλεγχο διαφορικής έκφρασης με τη χρήση του τεστ Wald. Στη συνάρτηση “results()” καθορίζεται η σύγκριση ενδιαφέροντος (contrast) και τα επίπεδα που θα συγκριθούν. Το επίπεδο που δίνεται τελευταίο είναι το επίπεδο βάσης για τη σύγκριση. Ο χρήστης μπορεί να επιλέξει τη σύγκριση ενδιαφέροντος (contrast) και τα επίπεδα που θα συγκριθούν τα οποία ανανεώνονται βάσει του “rhenodata” αρχείου και της επιλογής του. Το απόσπασμα του κώδικα που αφορά τη συγκεκριμένη διαδικασία βρίσκεται στις γραμμές 1182-1215. Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται ο τρόπος και οι συναρτήσεις με τις οποίες δημιουργούνται οι αντιθέσεις.

```
col<-input$factorNameInput
```

```
contr <- c(col, contrast1,contrast2)
```

```
res1<- results(dds, contrast=contr, alpha= 0.05)
```

alpha: Τιμή του ορίου της στατιστικής σημαντικότητας στην σύγκριση

contrast: Το όρισμα καθορίζει ποια σύγκριση θα εξαχθεί από το αντικείμενο για τη δημιουργία ενός πίνακα αποτελεσμάτων. Είναι ένα διάνυσμα χαρακτήρων με τρία στοιχεία, το όνομα ενός παράγοντα στον τύπο σχεδιασμού, και τα ονόματα των επιπέδων για την επιθυμητή σύγκριση.¹⁰

Με μεγάλους καταλόγους σημαντικών γονιδίων μπορεί να είναι δύσκολο να εξαχθεί ουσιαστική βιολογική συνάφεια. Για να αυξηθεί η αυστηρότητα, μπορεί επίσης να προστεθεί ένα κατώφλι “log fold change” και ένα κατώφλι “adj”.

Μπορούμε εύκολα να διαμορφώσουμε τον πίνακα αποτελεσμάτων έτσι ώστε να συμπεριλάβουμε μόνο αυτά που είναι σημαντικά χρησιμοποιώντας τη συνάρτηση “filter()”.

Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η διαδικασία που μόλις περιεγράφηκε.

```
differ_expr <- results %>%  
  
  filter(padj < padj.cutoff & abs(log2FoldChange) > lfc.cutoff)
```

Ο μετασχηματισμός “vst” (varianceStabilizingTransformation) που υλοποιείται στο “DESeq2”, υπολογίζει έναν μετασχηματισμό σταθεροποίησης της διακύμανσης, ο οποίος είναι περίπου παρόμοιος με την τοποθέτηση των δεδομένων στην κλίμακα “log2”, ενώ παράλληλα αντιμετωπίζει τη δειγματοληπτική μεταβλητότητα των χαμηλών μετρήσεων. Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται ο τρόπος με τον οποίο πραγματοποιείται ο μετασχηματισμός “vst” για τα δεδομένα της “DESeq2”.

```
vsd <- varianceStabilizingTransformation(dds)  
  
myValues$vsMat <- assay(vsd)
```

2.2.7. Ανάκτηση πληροφοριών για την ανάλυση εμπλουτισμού γονιδιακών συνόλων

Για την εύρεση κατάλληλων πληροφοριών για την ανάλυση εμπλουτισμού γονιδιακών συνόλων χρησιμοποιείται το πακέτο “AnnotationHub”¹¹.

Αρχικά το αρχείο με τα διαφορικά εκφρασμένα γονίδια διαβάζεται και αφαιρείται αυτόματα το μοτίβο “LOC”, σε περίπτωση που εντοπιστεί. Σε περίπτωση που υπάρχει επιπλέον μοτίβο δίνεται η δυνατότητα στον χρήστη να το αφαιρέσει. Ακολουθεί το αντίστοιχο κομμάτι κώδικα για αυτή τη διαδικασία. Το απόσπασμα του κώδικα που αφορά τη συγκεκριμένη διαδικασία βρίσκεται στις γραμμές 1297-1331.

Για την εύρεση των κατάλληλων πληροφοριών για τον επιθυμητό οργανισμό, χρησιμοποιείται το πακέτο “AnnotationHub”. Το οποίο δίνει τη δυνατότητα αλληλεπίδρασης του χρήστη με τον διακομιστή “AnnotationHub” του “Bioconductor”. Ο χρήστης έχοντας πρόσβαση στη βάση δεδομένων, μπορεί να ανακαλύψει “annotated” αρχεία για οποιονδήποτε οργανισμό τον ενδιαφέρει και να τα κατεβάσει. Βάσει της αναζήτησης του χρήστη παρέχονται διάφοροι τύποι δεδομένων που υπάρχουν μέσα στη βάση. Επίσης εμφανίζεται από πού προέρχονται τα δεδομένα (dataprovder), καθώς και τι είδους αντικείμενα δεδομένων R θα μπορούσαν να επιστραφούν ως έξοδος (rdataclass). Στην

συγκεκριμένη εφαρμογή επιστρέφονται μόνο αντικείμενα δεδομένων “OrgDb”, καθώς τέτοιου είδους αρχεία είναι συμβατά στην συνάρτηση εμπλουτισμού γονιδιακών συνόλων που χρησιμοποιείται στα παρακάτω βήματα των αναλύσεων. Έτσι διευκολύνεται και ο χρήστης στην αναζήτηση του κατάλληλου τύπου αρχείου. Κάθε αρχείο που ανακτάται από τον διακομιστή του “AnnotationHub” αποθηκεύεται σε τοπική προσωρινή τοποθεσία, έτσι ώστε την επόμενη φορά που θα χρειαστεί να ανακτηθεί, κατεβαίνει πολύ γρηγορότερα. Στο παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η συνάρτηση “AnnotationHub()” καθώς και οι παράμετροι που χρησιμοποιούνται στον κώδικα της εφαρμογής:

```
hub <- AnnotationHub()
```

```
annotation<-query(hub,c("OrgDb", organism))
```

```
annot <- hub[[input$annot]]
```

AnnotationHub(): Με τη συνάρτηση “AnnotationHub()” δημιουργείται ένα αντικείμενο “AnnotationHub” και ανακτώνται πληροφορίες σχετικά με όλες τις εγγραφές στην βάση δεδομένων.

query(hub,c("OrgDb", organism)): Με τη συνάρτηση “query()” γίνεται αναζήτηση στις εγγραφές της βάσης για τον οργανισμό που δόθηκε ως είσοδος/επιλογή από τον χρήστη. Οι εγγραφές που αναζητούνται είναι τις μορφής “OrgDb”.

hub[[input\$annot]]: Ανάκτηση μεμονωμένων εγγραφών γίνεται χρησιμοποιώντας τους χαρακτήρες “[”. Τα επιθυμητά αρχεία, μεταφορτώνονται από το διαδίκτυο σε μια τοπική κρυφή τοποθεσία. Σε όλες τις επόμενες χρήσεις τα αρχεία εισάγονται γρήγορα από την κρυφή τοποθεσία στην περίοδο (session) της R.

2.2.8. Ανάλυση εμπλουτισμού γονιδιακών συνόλων (Gene set enrichment)

Στη συνέχεια ακολουθεί η ανάλυση εμπλουτισμού γονιδιακών συνόλων, με χρήση της συνάρτησης “gseGO()” του πακέτου “clusterProfiler”¹². Η μέθοδος αυτή επιτρέπει τον εντοπισμό κατηγοριών γονιδίων που υπερεκφράζονται σε ένα μεγάλο σύνολο γονιδίων ή και ενδέχεται να έχουν συσχέτιση με φαινότυπους ασθενειών. Η μέθοδος χρησιμοποιεί στατιστικές προσεγγίσεις για τον εντοπισμό σημαντικά υπερεκφραζόμενων ή υποεκφραζόμενων ομάδων γονιδίων.

Αρχικά δημιουργείται μία λίστα γονιδίων η οποία περιέχει τα ονόματα των γονιδίων και τη τιμή της λογαριθμικής μεταβολής που τους αντιστοιχεί ($\log_2\text{FoldChange}$). Για να δημιουργηθεί αυτή η λίστα, απομονώνεται η στήλη που περιέχει τα ονόματα των γονιδίων και η στήλη που περιέχει τις τιμές της λογαριθμικής μεταβολής, από το αρχείο με τα διαφορικά εκφραζόμενα γονίδια που φορτώνει ο χρήστης. Τα ονόματα των γονιδίων εξαρτώνται από το γονιδίωμα αναφοράς που χρησιμοποιήθηκε σε προηγούμενο βήμα και έχουν την κωδικοποίηση που χρησιμοποιεί η βάση δεδομένων από την οποία προήλθε το αρχείο του γονιδιώματος αναφοράς. Για να μην δημιουργηθούν προβλήματα λόγω τις εκάστοτε κωδικοποίησης και για να γίνει σωστή ταυτοποίηση των γονιδίων στην ανάλυση εμπλουτισμού γονιδιακών συνόλων, τα ονόματα των γονιδίων μετατρέπονται σε “ENTREZID”. Αυτό με επιτυγχάνεται με τη βοήθεια της συνάρτησης “transId()” του πακέτου “genekitr”¹³. Ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η διαδικασία κατασκευής λίστας γονιδίων και η μετατροπή των ονομάτων των γονιδίων σε “ENTREZID”, με τον τρόπο που αναφέρθηκε παραπάνω.

```
df1<- df_products_upload3()

#transform gene_names to ENTREZIDs

f<-transId(df1$gene_name,org=input$org, transTo = "entrez")

gene_name<-f$entrezid

original_gene_list <- df1$log2FoldChange

# name the vector

names(original_gene_list) <- gene_name

# omit any NA values

gene_list<-na.omit(original_gene_list)

# sort the list in decreasing order (required for clusterProfiler)

gene_list = sort(gene_list, decreasing = TRUE)
```

transId(): Η συνάρτηση περιέχει τις εξής παραμέτρους: τα ονόματα των γονιδίων από το αρχείο που φορτώθηκε(df1\$gene_name), τον οργανισμό στον οποίο αντιστοιχούν τα γονίδια (org=input\$org,unique) ο οποίος επιλέγεται από τον χρήστη και τη μορφή κωδικοποίησης στην οποία θέλουμε να μετατραπούν τα δεδομένα, στη συγκεκριμένη περίπτωση σε “ENTREZID” (transTo = "entrez").

Τα ονόματα των οργανισμών για τη συνάρτηση “transId()” έχουν μία συγκεκριμένη μορφή, δηλαδή προκύπτουν από τον συνδυασμό του αρχικού της πρώτης λέξης της λατινικής τους ονομασίας με ολόκληρη τη δεύτερη λέξη (π.χ. homo sapiens = hsapiens, mus musculus = mmusculus, vitis vinifera = vviniifera). Ο χρήστης μπορεί να επιλέξει τον επιθυμητό οργανισμό από τη λίστα με τα ονόματα που του παρέχονται μέσω του παρακάτω τρόπου. Ακολουθεί απόσπασμα του κώδικα που αντιστοιχεί στη διαδικασία που περιεγράφηκε.

```
names<- genekitr::ensOrg_name
```

```
latin_short_name<-names$latin_short_name
```

```
#update organism names for the transformation function
```

```
updateSelectizeInput(session, "org", choices = latin_short_name, selected = latin_short_name[1],server=TRUE)
```

genekitr::ensOrg_name : Αυτή η εντολή επιστρέφει ένα “dataframe” δεδομένων με 4 στήλες (common_name, latin_full_name, latin_short_name, bioc_name) από αυτό απομονώνεται η στήλη με όνομα “latin_short_name” (latin_short_name<-names\$latin_short_name), καθώς αυτή περιέχει τις κατάλληλες ονομασίες για τη συνάρτηση transid(). Τελός ενημερώνεται το αντίστοιχο πλαίσιο επιλογής για τον χρήστη στο “UI” (updateSelectizeInput(session, "org", choices = latin_short_name, selected = latin_short_name[1],server=TRUE))

Τα “ENTREZID” των γονιδίων αποθηκεύονται σε μία μεταβλητή (gene_name<-f\$entrezid) και στη συνέχεια απομονώνονται οι τιμές της λογαριθμικής μεταβολής από το αρχείο (original_gene_list <- df1\$log2FoldChange). Τα “ENTREZID” των γονιδίων αντιστοιχούνται με τις τιμές “log2FoldChange” (names(original_gene_list) <- gene_name) δημιουργώντας την επιθυμητή λίστα κα μετά αφαιρούνται οι κενές μεταβλητές από αυτήν (gene_list<-na.omit(original_gene_list)). Τέλος, η λίστα ταξινομείται κατά φθίνουσα σειρά (gene_list = sort(gene_list, decreasing = TRUE)) καθώς αυτή είναι η επιθυμητή η δομή για τη συνάρτηση “gseGO()”.

Εάν ο στόχος του χρήστη είναι να κατατάξει τα σημαντικά γονιδιακά σύνολα, τότε μπορεί να αυξήσει αρκετά την παράμετρο “nperm” και να δει πώς να δει πώς αυτά τα σύνολα διαχωρίζονται, ωστόσο αυτό θα απαιτήσει πολύ περισσότερο υπολογιστικό χρόνο. Στο

παρακάτω απόσπασμα, ακολουθεί τμήμα του κώδικα στο οποίο παρουσιάζεται η συνάρτηση “gseGO()” καθώς και οι παράμετροι που χρησιμοποιούνται στην εφαρμογή:

```
go_gse <- gseGO(geneList=gene_list,  
  
                ont = input$ontology,  
  
                keyType = "ENTREZID",  
  
                nPerm = input$nPerm,  
  
                minGSSize = input$minGSSize,  
  
                maxGSSize = input$maxGSSize,  
  
                pvalueCutoff = input$pvalCutoff,  
  
                verbose = T,  
  
                OrgDb = annotation,  
  
                pAdjustMethod = input$pAdjustMethod)
```

geneList: Αντιστοιχεί σε μία λίστα γονιδίων (με τα ονόματα των γονιδίων και το αντίστοιχο “log2FoldChange”) που βρίσκονται σε σειρά κατάταξης βάσει του “log2FoldChange”.

ont: Αντιστοιχεί σε μία από τις εξής οντολογίες: "BP", "MF", και "CC", ή "ALL" για όλες (προεπιλογή).

keyType: Αντιστοιχεί στον κωδικό των gene ids. Στην εφαρμογή χρησιμοποιούνται μόνο "ENTREZID" κωδικοί.

nPerm: Αυτή η παράμετρος καθορίζει πόσες φορές γίνεται αυτή η τυχαιοποίηση και όσο περισσότερες τυχαιοποιήσεις εκτελούνται, τόσο πιο ακριβής θα είναι η εκτίμηση της τιμής FDR.

minGSSize: Αντιστοιχεί στο ελάχιστο μέγεθος του κάθε γονιδιακού συνόλου, για την ανάλυση (3 από προεπιλογή).

maxGSSize: Αντιστοιχεί στο μέγιστο μέγεθος των “annotated” γονιδίων, για την δοκιμή(testing) (800 από προεπιλογή).

pvalueCutoff: Κατώφλι για το “pvalue” (0.05 από προεπιλογή).

verbose: Τυπώνει μήνυμα του “clusterProfiler” αν είναι “TRUE”.

OrgDb: Αντιστοιχεί στο "annotation file" που επιλέχθηκε στο προηγούμενο βήμα.

pAdjustMethod: Μέθοδος υπολογισμού του "pAdjust" από τις εξής επιλογές: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" ή "none" για καμία (προεπιλογή).

2.3. Συνολική αξιολόγηση της εφαρμογής από χρήστες

Στο πλαίσιο της αξιολόγησης της εφαρμογής, καθώς και για τη συγκέντρωση παρατηρήσεων με στόχο την βελτίωση της απόδοσης της βάσει των αναγκών κάθε χρήστη, δημιουργήθηκε η παρακάτω φόρμα μέσω "Google Forms". Ακολουθούν οι ερωτήσεις και οι απαντήσεις από τους χρήστες.

Οι οδηγίες χρήσης της εφαρμογής ήταν κατατοπιστικές και κατανοητές;

Αν όχι τι θα θέλατε να διευκρινιστεί;

3 απαντήσεις

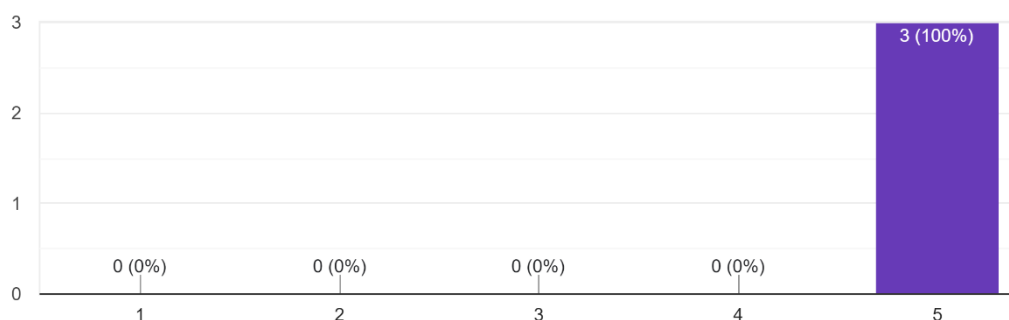
ΝΑΙ

Ναι, ήταν κατατοπιστικές

Οι οδηγίες ήταν ξεκάθαρες και κατανοητές ωστόσο απαιτούν βασικές γνώσεις από τη μεθοδολογία της αναλυτικής διαδικασίας. Θα μπορούσαν να απλοποιηθούν κι άλλο, ώστε να μην απαιτούνται καθόλου τεχνικές, παρά μόνο βιολογικές γνώσεις επι του θέματος.

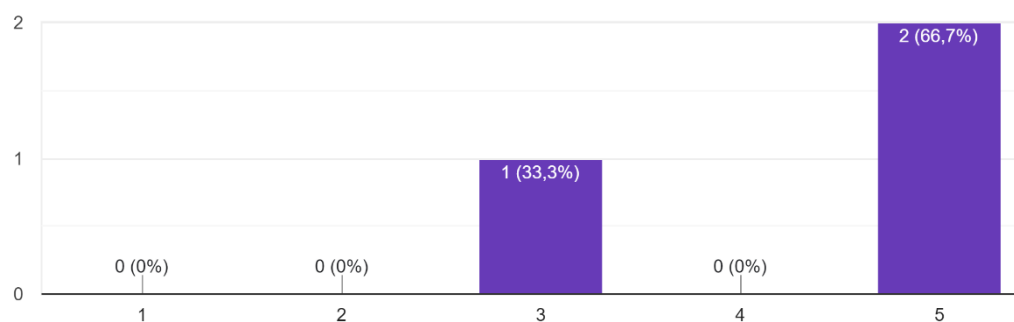
Θα χαρακτηρίζατε εύκολη τη διαδικασία του Trimming; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



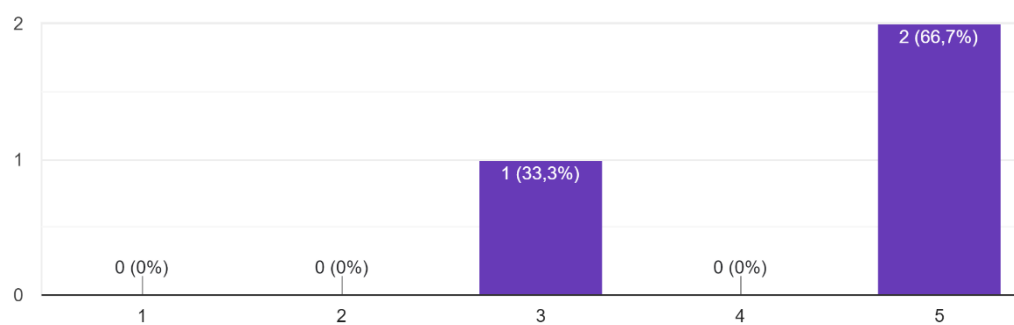
Ήταν εύχρηστο το γραφικό περιβάλλον για τη συγκεκριμένη διαδικασία; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



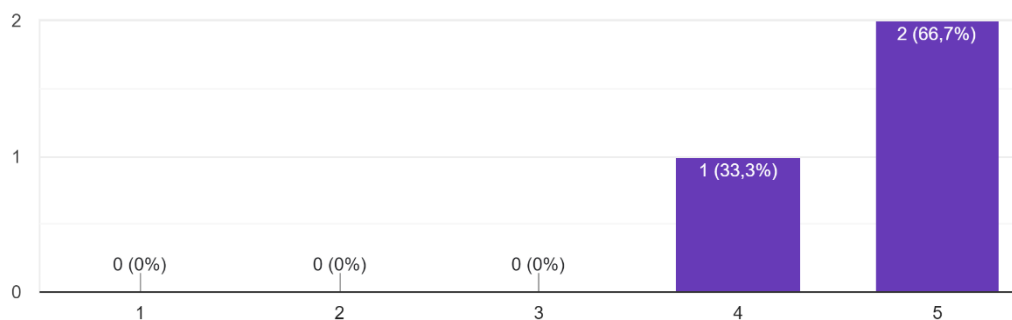
Πως θα αξιολογούσατε τη χρονική διάρκεια της διαδικασίας του Trimming; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



Πως θα αξιολογούσατε τα αποτελέσματα της διαδικασίας του Trimming; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



Θα θέλατε να παρέχονται επιπλέον επιλογές ή φίλτρα στο πλαίσιο της διαδικασίας του φιλτραρίσματος;

3 απαντήσεις

ΝΑΙ

Ναι

Χρήσιμες θα ήταν πληροφορίες σχετικά με τις αρχικές και τις τελικές αλληλουχίες, τι ποσοστό δηλαδή ήταν αυτό που δεν πληρούσε τα κριτήρια για να προχωρήσει σε παρακάτω ανάλυση.

Έχετε να κάνετε κάποιο περαιτέρω σχόλιο για τη συγκεκριμένη διαδικασία;

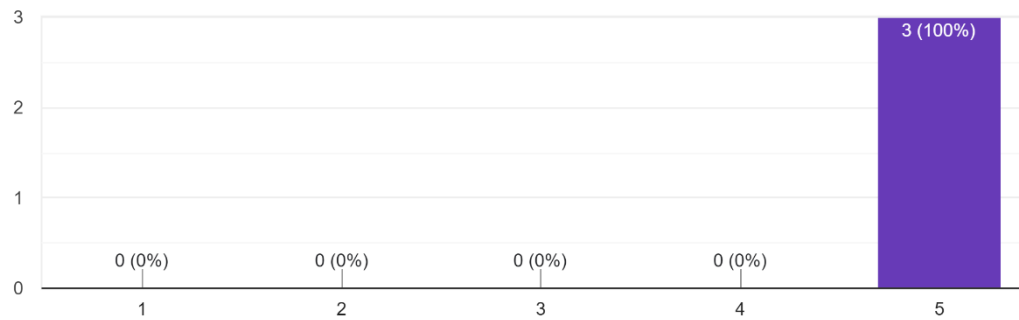
2 απαντήσεις

Trimming log files are overwritten by next sample

Θα βοηθούσε να εμφανίζεται το ποσοστό της προόδου της διαδικασίας, αν είναι εφικτό, για να έχει ο χρήστης μια εικόνα του χρόνου που χρειάζεται. Αυτό θα βοηθούσε στην εκτίμηση του χρόνου που χρειάζεται η διαδικασία ανάλογα με το εκάστοτε σύνολο δεδομένων και τα χαρακτηριστικά του συστήματος που χρησιμοποιούνται (π.χ. αριθμός επεξεργαστών) αλλά και στο να μη χάσει ο χρήστης της υπομονή του και αποφασίσει εν τέλει να μη χρησιμοποιήσει την εφαρμογή.

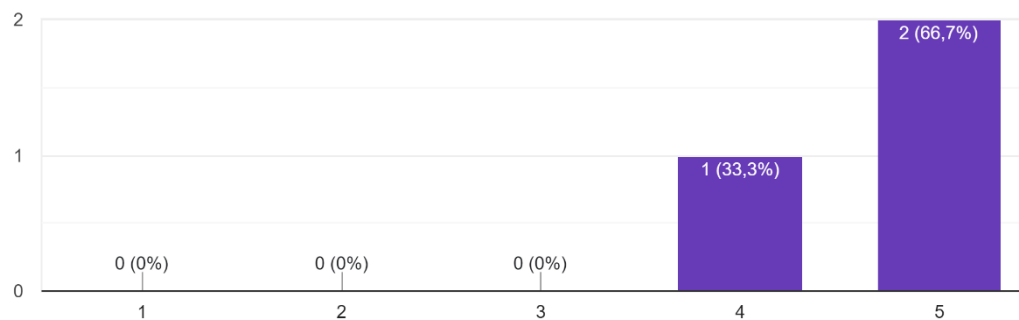
Θα χαρακτηρίζατε εύκολη τη διαδικασία του Mapping; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



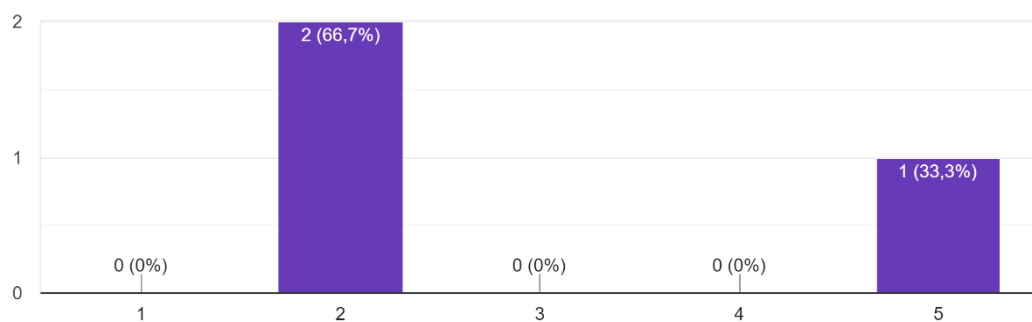
Ήταν εύχρηστο το γραφικό περιβάλλον για τη συγκεκριμένη διαδικασία; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



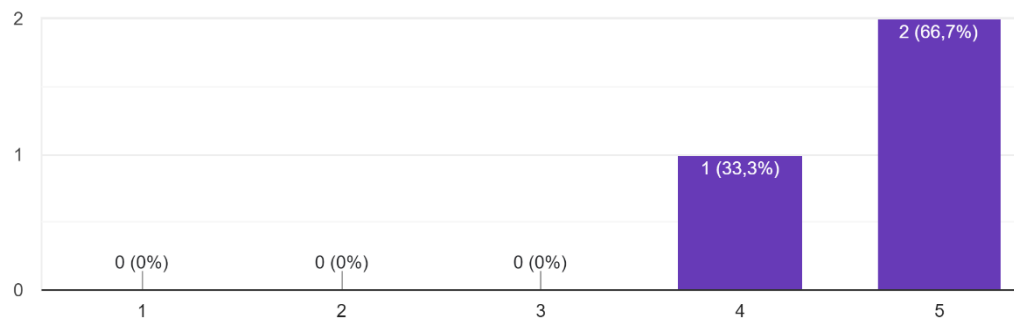
Πως θα αξιολογούσατε τη χρονική διάρκεια της διαδικασίας του Mapping(Index, Alignment); Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



Πως θα αξιολογούσατε τα αποτελέσματα της διαδικασίας του Mapping; Βαθμολογείτε με άριστα το 5:

3 απαντήσεις



Θα θέλατε να παρέχονται επιπλέον επιλογές ή φίλτρα στο πλαίσιο της διαδικασίας της στοίχισης;

3 απαντήσεις

ΝΑΙ

Όχι

Ιδανική θα ήταν η παρουσία προεπιλογών του τύπου "strict", "normal" & "flexible", όπου επιλέγοντάς τις συμπληρώνονται αντίστοιχα οι τιμές στα πεδία.

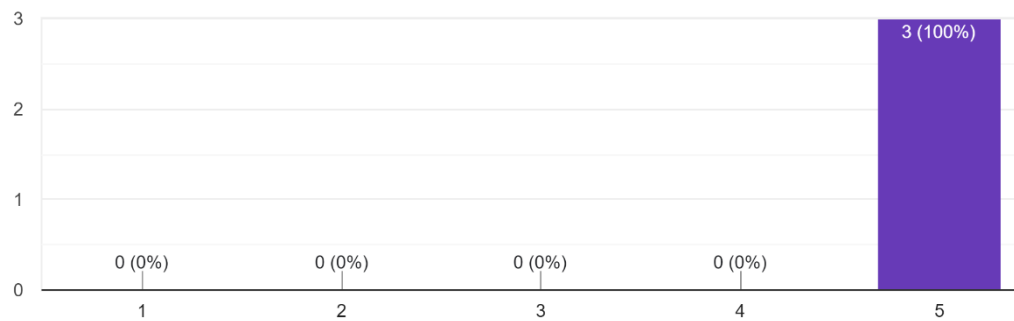
Έχετε να κάνετε κάποιο περαιτέρω σχόλιο για τη συγκεκριμένη διαδικασία;

1 απάντηση

Και πάλι, θα βοηθούσε να εμφανίζεται το ποσοστό ολοκλήρωσης της διαδικασίας.

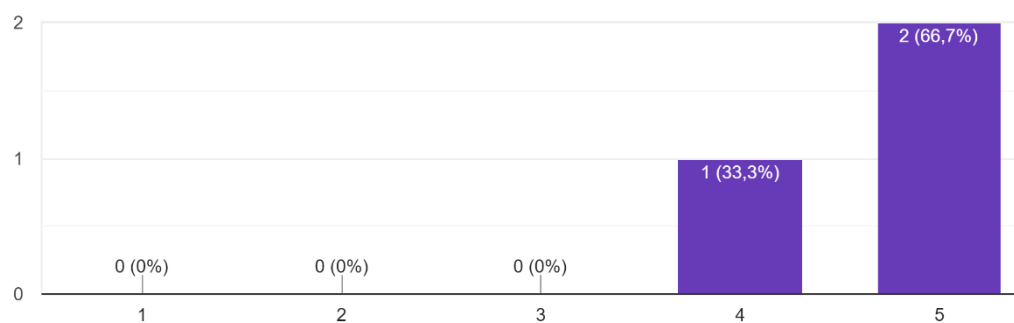
Πως θα αξιολογούσατε τη χρονική διάρκεια της διαδικασίας του Quantification; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



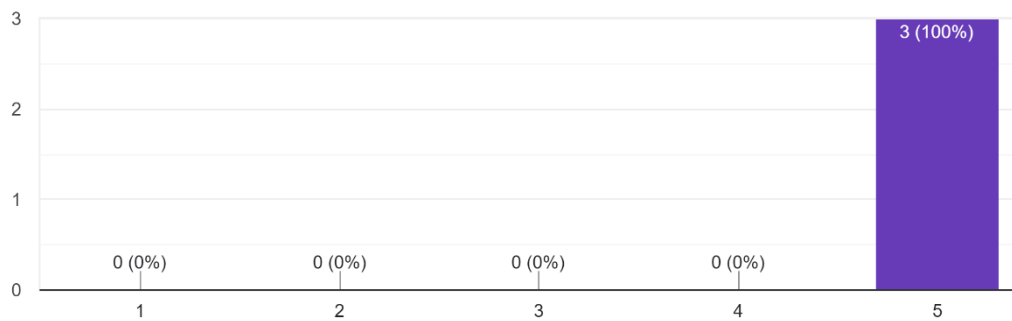
Ήταν εύχρηστο το γραφικό περιβάλλον για τη συγκεκριμένη διαδικασία; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



Πως θα αξιολογούσατε τα αποτελέσματα της διαδικασίας του Quantification; Βαθμολογείτε με άριστα το 5:

3 απαντήσεις



Θα θέλατε να παρέχονται επιπλέον επιλογές ή φίλτρα στο πλαίσιο της διαδικασίας της ποσοτικοποίησης;

2 απαντήσεις

Όχι

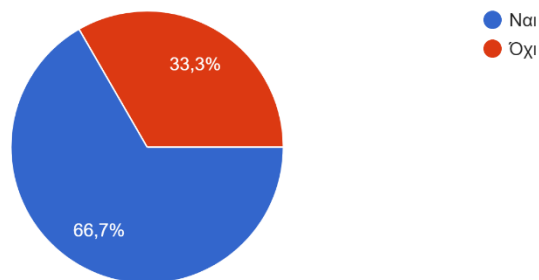
Ομοίως, θα ήταν χρήσιμη η παρουσία προεπιλογών για τις παραμέτρους.

Έχετε να κάνετε κάποιο περαιτέρω σχόλιο για τη συγκεκριμένη διαδικασία

Some global report (e.g. nr genes quantified)

Ομοίως με τις προηγούμενες διαδικασίες, μια εικόνα του χρόνου που χρειάζεται για να ολοκληρωθεί η διαδικασία ή του σταδίου στο οποίο βρίσκεται η διαδικασία (π.χ. για κάθε δείγμα που ολοκληρώνεται η ποσοτικοποίηση να εμφανίζεται ένα μικρό μήνυμα) θα ήταν χρήσιμη.

Το πλαίσιο με τις ρυθμίσεις και τις παραμέτρους της διαφορικής έκφρασης ήταν κατανοητό και εύχρηστο;
3 απαντήσεις



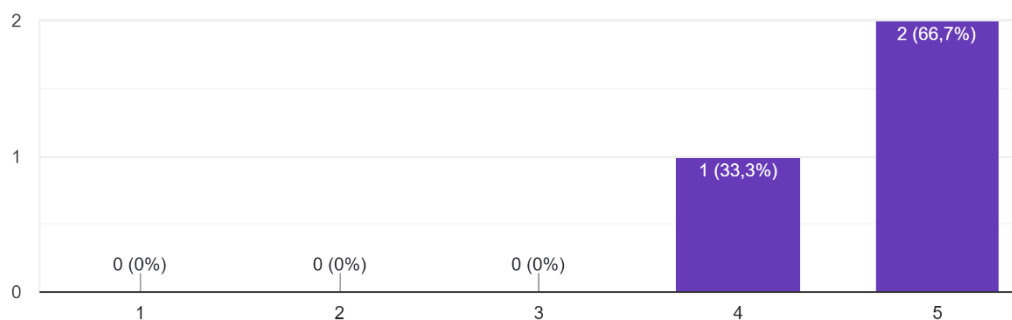
Θα θέλατε να παρέχονται επιπλέον επιλογές ή φίλτρα στο πλαίσιο της διαφορικής έκφρασης;

2 απαντήσεις

Ίσως μια παραπομπή στο documentation του πακέτου DESeq2 να ήταν βοηθητική και αρκετή.

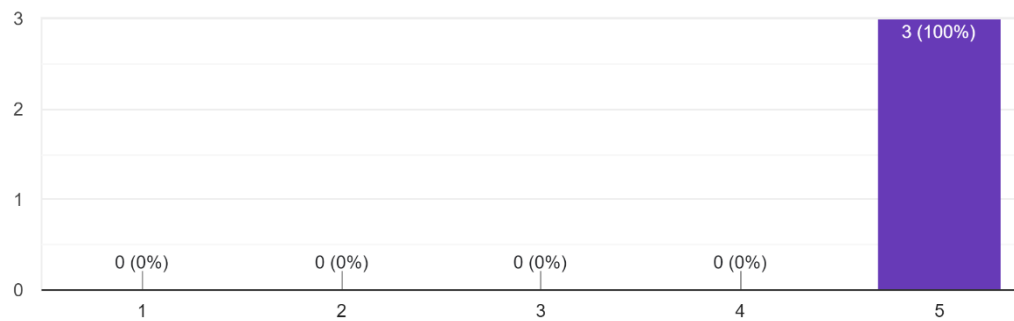
Χρήσιμες θα ήταν οι οδηγίες για τη δημιουργία του αρχείου phenodata.

Πως θα αξιολογούσατε τη χρονική διάρκεια της διαδικασίας της ανάλυσης διαφορικής έκφρασης; Βαθμολογείτε με άριστα το 5:
3 απαντήσεις



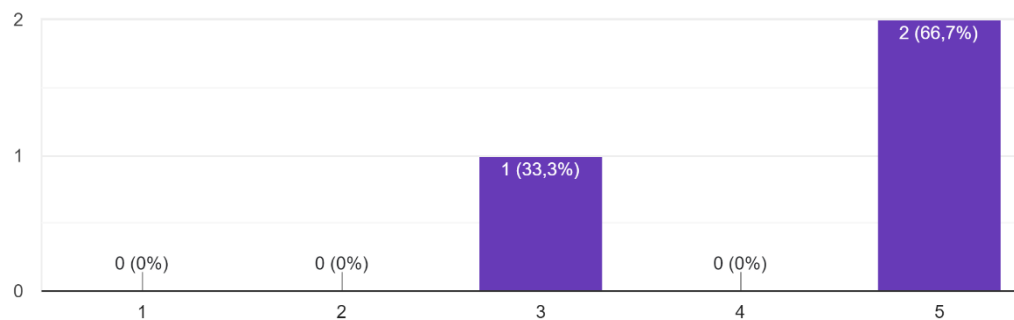
Πως θα αξιολογούσατε τα στατιστικά αποτελέσματα της ανάλυσης της διαφορικής έκφρασης; Βαθμολογείτε με άριστα το 5:

3 απαντήσεις



Πως θα αξιολογούσατε το περιεχόμενο των διαγραμμάτων της ανάλυσης της διαφορικής έκφρασης; Βαθμολογείτε με άριστα το 5:

3 απαντήσεις



Θα θέλατε να παρέχονται επιπλέον επιλογές ή φίλτρα για τα διαγράμματα;

2 απαντήσεις

Ναι, θα ήθελα να μπορώ να ορίσω τις διαστάσεις των διαγραμμάτων στα αρχεία pdf που εξάγω.

Θα ήταν χρήσιμη η δημιουργία διαδραστικών διαγραμμάτων.

Έχετε να κάνετε κάποιο περαιτέρω σχόλιο για τη συγκεκριμένη διαδικασία;

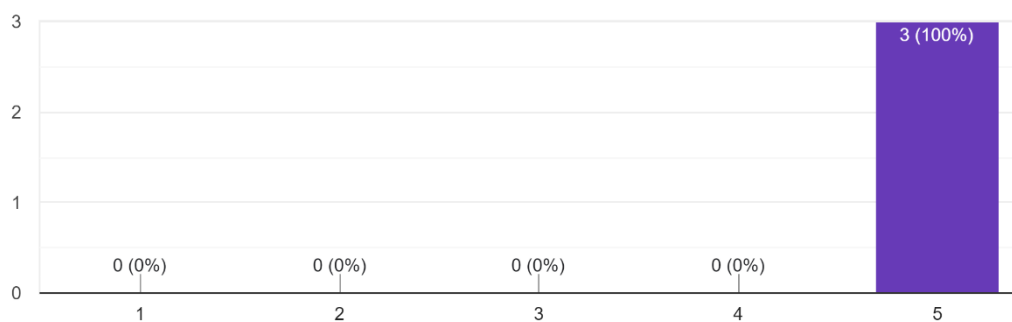
2 απαντήσεις

Phenodata CSV file: [more instructions](#)

Όχι

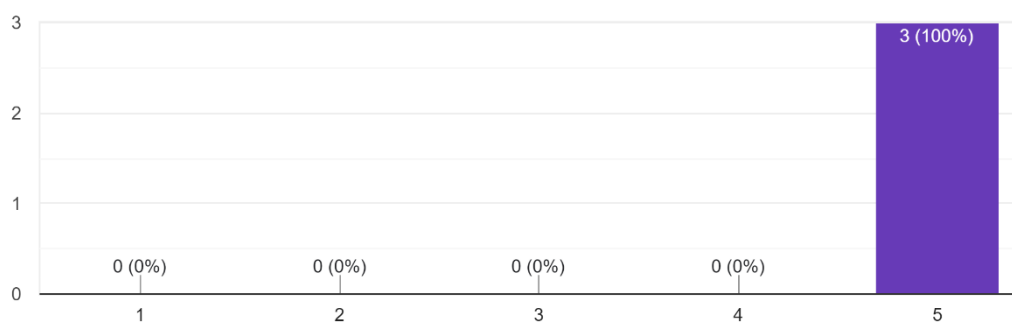
Θα χαρακτηρίζατε εύκολη τη διαδικασία της ανάλυσης εμπλουτισμού γονιδιακών συνόλων
; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



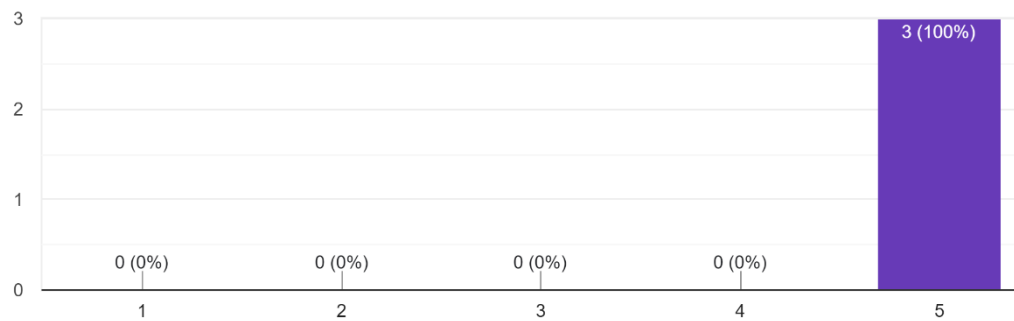
Πως θα αξιολογούσατε τη χρονική διάρκεια της διαδικασίας της ανάλυσης
εμπλουτισμού γονιδιακών συνόλων; Βαθμολογείστε με άριστα το 5:

3 απαντήσεις



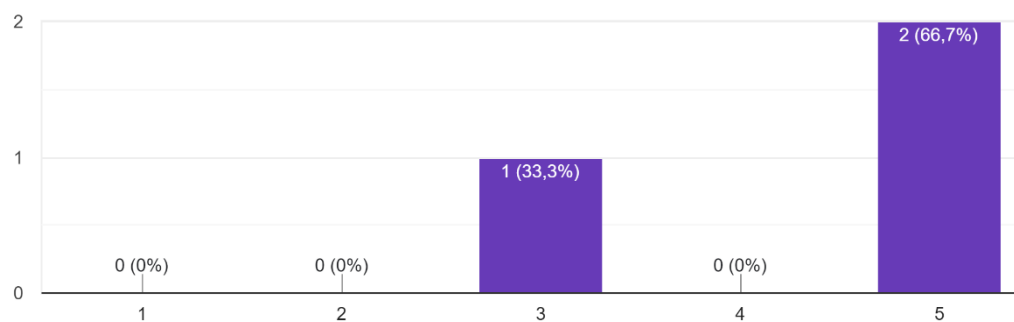
Πως θα αξιολογούσατε τα στατιστικά αποτελέσματα της ανάλυσης εμπλουτισμού γονιδιακών συνόλων; Βαθμολογείτε με άριστα το 5:

3 απαντήσεις



Πως θα αξιολογούσατε το περιεχόμενο των διαγραμμάτων της ανάλυσης αυτής; Βαθμολογείτε με άριστα το 5:

3 απαντήσεις



Θα θέλατε να παρέχονται επιπλέον επιλογές ή φίλτρα για τα διαγράμματα;

1 απάντηση

Ναι, θα ήθελα να μπορώ να ορίσω τις διαστάσεις των διαγραμμάτων στα αρχεία pdf που εξάγω.

Πως θα αξιολογούσατε γενικότερα την εφαρμογή;

3 απαντήσεις

ΑΞΙΟΛΟΓΗ

Πολύ χρήσιμη και εύχρηστη εφαρμογή, ιδιαίτερα για άτομα χωρίς γνώσεις βιοπληροφορικής. 10/10!

Πολύ χρήσιμη εφαρμογή, χωρίς ιδιαίτερες απαιτήσεις σε υλικολογισμικό. Σχετικά γρήγορη με χρήσιμα διαγράμματα. Χρήζει βελτιώσεων σε λεπτομέρειες, ωστόσο είναι πλήρως λειτουργική.

Βάσει των παρατηρήσεων που συγκεντρώθηκαν η εφαρμογή τροποποιήθηκε ανάλογα σε διάφορες λειτουργίες, ώστε να καταλήξουμε στο επιθυμητό αποτέλεσμα.

2.4. Οδηγίες χρήσης της εφαρμογής

Για την δυνατότητα χρήσης της εφαρμογής από χρήστες που δεν έχουν το ανάλογο υπόβαθρο σε τέτοιου είδους αναλύσεις, δημιουργήθηκαν οι παρακάτω οδηγίες προς διευκόλυνσή τους.

Φιλτράρισμα

1. Φορτώστε τα ακατέργαστα δεδομένα σας, τα οποία θα πρέπει να είναι αποσυμπιεσμένα και σε μορφή “.fastq”. (Στο πλαίσιο “Upload files”)
2. Επιλέξτε τα αρχεία που θέλετε να επεξεργαστείτε ανά ζεύγη(forward-reverse files). (Στο πλαίσιο “Trimming Process”)
3. Τα φιλτραρισμένα αρχεία αποθηκεύονται αυτόματα στον κατάλογο εργασίας, με τη χαρακτηριστική κατάληξη:”_trimmed.fastq”
4. Αφού ολοκληρώσετε την διαδικασία του φιλτραρίσματος, μπορείτε να διαγράψετε τα αρχεία που είχατε προηγουμένως μεταφορτώσει. (Διαγράφονται όλα τα αρχεία μέσα στον υποκατάλογο).

Στοιχίση

1. Για αυτό το βήμα, δεν χρειάζεται να μεταφορτώσετε κανένα αρχείο, όλα τα φιλτραρισμένα αρχεία από τον κατάλογο εργασίας σας θα χαρτογραφηθούν. Εάν έχετε φιλτραρισμένα αρχεία τα οποία δεν έχουν επεξεργαστεί μέσω της εφαρμογής, τοποθετήστε τα στον κατάλογο εργασίας.
2. Βεβαιωθείτε ότι το αρχείο του γονιδιώματος αναφοράς είναι σε μορφή “.fasta/.fa/fna” και βρίσκεται μέσα στον κατάλογο εργασίας.
3. Κατασκευάστε το ευρετήριο γονιδιώματος, αν αναλύετε για πρώτη φορά αυτόν τον οργανισμό με την εφαρμογή. Σε επόμενες αναλύσεις του ίδιου οργανισμού δεν χρειάζεται να κατασκευάσετε πάλι ευρετήριο. (Στο πλαίσιο “Build Index”)
4. Μπορείτε να χρησιμοποιήσετε τις προεπιλεγμένες ρυθμίσεις για τη στοίχιση ή να τις προσαρμόσετε. (Στο πλαίσιο “Alignment”)
5. Οι ευθυγραμμισμένες αναγνώσεις αποθηκεύονται αυτόματα στον κατάλογο εργασίας, με τη χαρακτηριστική κατάληξη: “.subread.SAM” ή “.BAM”

Ποσοτικοποίηση

1. Για αυτό το βήμα, δεν χρειάζεται να μεταφορτώσετε κανένα αρχείο, όλες οι στοιχισμένες αναγνώσεις σε αρχεία μορφής “.SAM” ή “.BAM” εντός του καταλόγου εργασίας θα καταμετρηθούν. Εάν έχετε στοιχισμένα αρχεία που δεν έχουν επεξεργαστεί από την εφαρμογή, τοποθετήστε τα στον κατάλογο εργασίας.
2. Βεβαιωθείτε ότι έχετε αποθηκεύσει το αρχείο του γονιδιώματος αναφοράς μέσα στον κατάλογο εργασίας και ότι το αρχείο έχει τη μορφή “.gtf”.
3. Μπορείτε να χρησιμοποιήσετε τις προεπιλεγμένες ρυθμίσεις για την ποσοτικοποίηση ή να τις προσαρμόσετε. (Στο πλαίσιο “Quantification Process”)
4. Μπορείτε να αποθηκεύσετε τα αποτελέσματα σε ένα αρχείο με μορφή “.csv”.

Ανάλυση διαφορικής έκφρασης

5. Φορτώστε το αρχείο με μορφή “.csv” που περιέχει τις μετρήσεις. (Στο πλαίσιο “Upload the counts file”)
6. Ανεβάστε το αρχείο σε μορφή “.csv” που περιέχει τις πληροφορίες για κάθε δείγμα (phenodata) και βεβαιωθείτε ότι τα δείγματα έχουν ακριβώς το ίδιο όνομα και την ίδια σειρά όπως στο αρχείο των μετρήσεων! (Στο πλαίσιο “Upload the phenodata file”)
7. Μόλις ανεβάσετε το αρχείο “phenodata” θα μπορείτε να δείτε όλα τα ονόματα των στηλών στο πλαίσιο της φόρμουλας σχεδιασμού. Επιλέξτε τις στήλες που χρειάζεστε για την ανάλυσή σας. Μπορείτε να σχεδιάσετε μία φόρμουλα που να περιέχει από

έναν έως τρεις παράγοντες(οι οποίοι να αποτελούνται από διαφορετικά στοιχεία). (Στο πλαίσιο “DESeq2 Object Settings”)

8. Μπορείτε να καθορίσετε τη σύγκριση που σας ενδιαφέρει και τα επίπεδα που θα συγκρίνετε. Το επίπεδο που δίνεται τελευταίο είναι το επίπεδο βάσης για τη σύγκριση. Πρώτα πρέπει να επιλέξετε τη στήλη αντίθεσης και στη συνέχεια να πατήσετε το κουμπί «Update levels by your contrast choice» (Ενημέρωση επιπέδων με βάση την επιλογή σας για την αντίθεση). Στη συνέχεια μπορείτε να επιλέξετε τα επίπεδα της αντίθεσης. (Στο πλαίσιο “DESeq2 Object Settings”)
9. Μπορείτε να χρησιμοποιήσετε τις προεπιλεγμένες ρυθμίσεις για την τιμή “p-adjust” και την τιμή “log2FoldChange” ή να τις προσαρμόσετε. (Στο πλαίσιο “DESeq2 Object Settings”)
10. Μπορείτε να αποθηκεύσετε τα αποτελέσματα σε ένα αρχείο με μορφή “.csv”.

Διαγράμματα διαφορικής έκφρασης

Μπορείτε να επιλέξετε χρώματα στην οπτικοποίηση των διαγραμμάτων “Volcano”, “MA plot” και “Boxplot” βάσει του προσωπικού σας γούστου.

“Volcano plot”: Οπτικοποίηση για τα διαφορικά εκφραζόμενα γονίδια. Μπορείτε να αποθηκεύσετε το διάγραμμα σε μορφή “.pdf/.png”

“MA plot”: Οπτικοποίηση για τα διαφορικά εκφραζόμενα γονίδια. Μπορείτε να αποθηκεύσετε το διάγραμμα σε μορφή “.pdf/.png”

“PCA plot”: Οπτικοποίηση για κανονικοποιημένα counts γονιδίων με τη συνάρτηση vst . Μπορείτε να αποθηκεύσετε το διάγραμμα σε μορφή “.pdf/.png”

“Heatmap”: Οπτικοποίηση για τα 30 σημαντικότερα μεταβλητά γονίδια. Μπορείτε να επιλέξετε τον αριθμό των κατηγοριών που θα οπτικοποιηθούν. Πρώτα επιλέξτε τις κατηγορίες που επιθυμείτε και στη συνέχεια τον αριθμό. Μπορείτε να αποθηκεύσετε το διάγραμμα σε μορφή “.pdf/.png”

“Boxplot for gene expression”: Οπτικοποίηση για τα διαφορικά εκφραζόμενα γονίδια. Επιπλέον μπορείτε να επιλέξετε τα γονίδια που θέλετε να οπτικοποιηθούν καθώς και δύο κατηγορίες σύγκρισης. Μπορείτε να αποθηκεύσετε το διάγραμμα σε μορφή “.pdf/.png”

“Boxplot for counts”: Οπτικοποίηση για τις κατανομές καταμέτρησης (counts) μεταξύ διαφορετικών δειγμάτων. Μπορείτε να επιλέξετε χρωματική παλέτα και να αποθηκεύσετε το διάγραμμα σε μορφή “.png”

Ανάλυση εμπλουτισμού γονιδιακών συνόλων

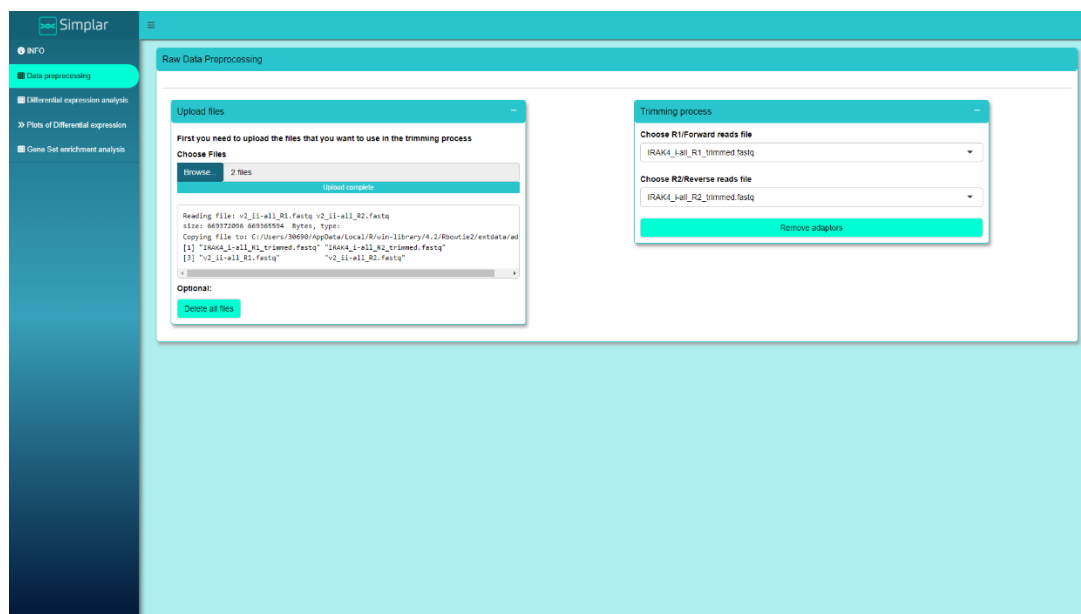
1. Φορτώστε το αρχείο με μορφή “.csv” με τα αποτελέσματα της ανάλυσης διαφορικής έκφρασης. Εάν ανεβάσετε ένα αρχείο που δεν έχει επεξεργαστεί μέσω της εφαρμογής, βεβαιωθείτε ότι η στήλη που περιέχει τα αναγνωριστικά γονιδίων έχει το όνομα "gene_name" και η στήλη που περιέχει τα αποτελέσματα της λογαριθμικής μεταβολής “log2 Fold Change” έχει το όνομα "log2FoldChange". (Στο πλαίσιο “Upload data”)
2. Εάν υπάρχει ένα μοτίβο μπροστά από τα “gene ids” στη στήλη “gene_name” , εκτός από το μοτίβο LOC που αφαιρείται αυτόματα, θα πρέπει να το αφαιρέσετε, γράφοντας το μοτίβο στο κατάλληλο πλαίσιο ("Give the pattern") και στη συνέχεια τσεκάροντας το πλαίσιο "Remove pattern".
3. Πληκτρολογήστε τον οργανισμό της επιλογής σας στο πλαίσιο "Select organism (latin name):". Στη συνέχεια, πατήστε το κουμπί αναζήτησης. (Στο πλαίσιο “Annotation Parameters”).
4. Αντιγράψτε και επικολλήστε τον κωδικό του επιθυμητού αρχείου από το AnnotationHub στο πλαίσιο "Select annotation file code:". Στη συνέχεια, πατήστε το κουμπί λήψης. (Στο πλαίσιο “Annotation Parameters”)
5. Τα ονόματα των οργανισμών στο πλαίσιο “ Select organism (short name):” έχουν μία συγκεκριμένη μορφή, δηλαδή προκύπτουν από τον συνδυασμό του αρχικού της πρώτης λέξης της λατινικής τους ονομασίας με ολόκληρη τη δεύτερη λέξη (π.χ. homo sapiens = hsapiens, mus musculus = mmusculus, vitis vinifera = vvinifera). Ο χρήστης μπορεί να επιλέξει τον επιθυμητό οργανισμό από τη λίστα με τα ονόματα που του παρέχονται. (Στο πλαίσιο “Gene Set Enrichment Object Parameters”)
6. Μπορείτε να χρησιμοποιήσετε τις προεπιλεγμένες ρυθμίσεις για την ανάλυση εμπλουτισμού ή να τις προσαρμόσετε. (Στο πλαίσιο “Gene Set Enrichment Object Parameters”)
7. Μπορείτε να αποθηκεύσετε τα στατιστικά αποτελέσματα σε μορφή “.csv”

Διαγράμματα εμπλουτισμού γονιδιακών συνόλων

“Dotplot”: Μπορείτε να επιλέξετε τον αριθμό των κατηγοριών που θα οπτικοποιηθούν και να το αποθηκεύσετε σε μορφή “.pdf/.png”

“Category Netplot”: Μπορείτε να επιλέξετε τον αριθμό των κατηγοριών που θα οπτικοποιηθούν και να το αποθηκεύσετε σε μορφή “.pdf/.png”

“GSEA plot”: Μπορείτε να επιλέξετε τον αριθμό των κατηγοριών που θα οπτικοποιηθούν και να το αποθηκεύσετε σε μορφή “.pdf/.png”



Εικόνα 5 : Το γραφικό περιβάλλον της καρτέλας “Data Preprocessing”, στην οποία πραγματοποιείται η φόρτωση των ακατέργαστων αρχείων στην εφαρμογή καθώς και η διαδικασία του φιλτραρίσματος.

The screenshot displays the Simlpar web interface for Differential Expression Analysis. The interface is organized into a sidebar and a main content area. The sidebar on the left contains navigation links: 'INFO', 'Data preprocessing', 'Differential expression analysis' (highlighted), 'Plots of Differential expression', and 'Gene Set enrichment analysis'. The main content area is divided into three primary sections: 'Mapping process', 'Quantification process', and 'Differential Expression Analysis'.

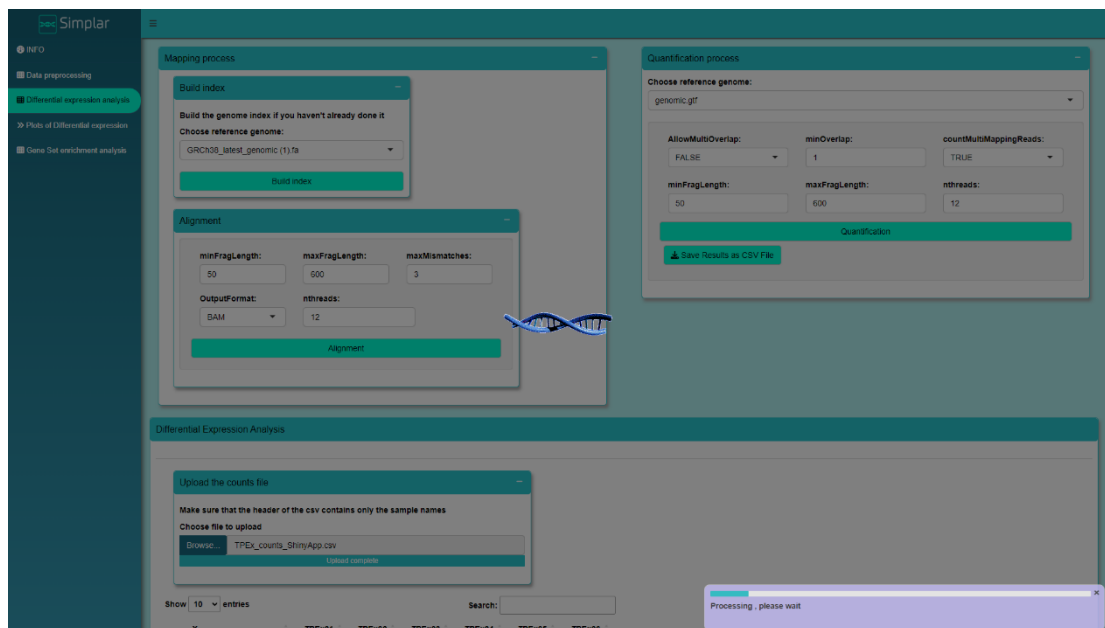
Mapping process: This section contains two sub-panels. The 'Build index' panel prompts the user to 'Build the genome index if you haven't already done it' and provides a dropdown for 'Choose reference genome:' (set to 'GRCh38_latest_genomic (1).fa') and a 'Build index' button. The 'Alignment' panel includes input fields for 'minFragLength:' (50), 'maxFragLength:' (600), 'maxMismatch:' (3), 'OutputFormat:' (BAM), and 'nthreads:' (12), with an 'Alignment' button.

Quantification process: This panel allows the user to 'Choose reference genome:' (set to 'genomic.gtf'). It includes input fields for 'AllowMultiOverlap:' (FALSE), 'minOverlap:' (1), 'countMultiMappingReads:' (TRUE), 'minFragLength:' (50), 'maxFragLength:' (600), and 'nthreads:' (12). A 'Quantification' button and a 'Save Results as CSV File' button are also present.

Differential Expression Analysis: This section contains two sub-panels. The 'Upload the counts file' panel instructs the user to 'Make sure that the header of the csv contains only the sample names' and provides a 'Choose file to upload' dropdown (set to 'No file selected') and a 'Browse...' button. The 'Upload the phenodata file' panel instructs the user to 'Make sure that the file contains the same samples as the counts file and that the names match' and provides a 'Choose file to upload' dropdown (set to 'phenodata1.csv') and an 'Upload complete' button.

DESeq2 Object Settings: This panel allows the user to 'Design your own formula depending on your analysis needs (differential expression with multiple or single factors)'. It includes a 'Design Formula:' input field (set to 'condition + variety'), a 'Choose the contrast column:' dropdown (set to 'condition'), and 'Update Levels by your contrast choice' button. It also includes 'Choose Level 1:' (set to 'control') and 'Choose Level 2:' (set to 'drought') dropdowns. At the bottom, there are input fields for 'P-Adjust Cutoff:' (0.05) and 'log2FoldChange Cutoff:' (1), a 'Get Results' button, and a 'Save Results as CSV File' button.

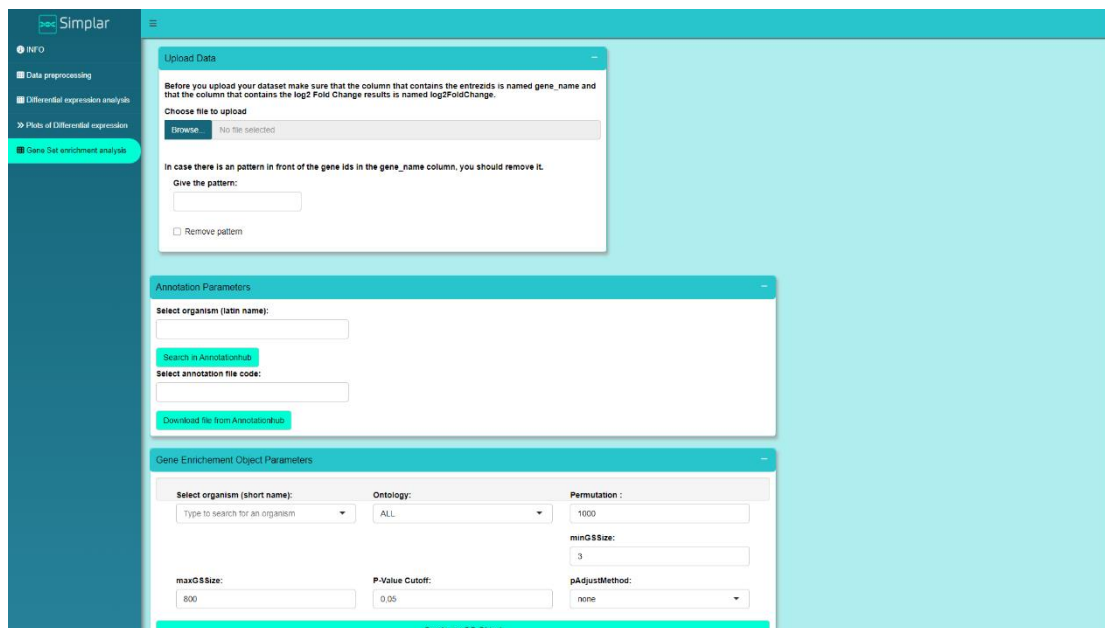
Εικόνα 6 : Το γραφικό περιβάλλον της καρτέλας “Differential Expression analysis”, στην οποία πραγματοποιείται η διαδικασία της στοίχισης, της ποσοτικοποίησης, και η ανάλυση διαφορικής έκφρασης.



Εικόνα 7: Το γραφικό περιβάλλον της καρτέλας “Differential Expression analysis”, κατά τη διάρκεια εκτέλεσης μιας διεργασίας. Παρατηρούμε ότι εμφανίζεται το “loading gif” και μία ειδοποίηση εκτέλεσης διεργασίας καθώς και ότι το γραφικό περιβάλλον γίνεται πιο σκούρο.



Εικόνα 8 : Παρουσιάζεται το γραφικό περιβάλλον της καρτέλας “Plots of Differential expression”, στην οποία εμφανίζονται τα διαγράμματα της συγκεκριμένης διαδικασίας.



Εικόνα 9 : Το γραφικό περιβάλλον της καρτέλας “Gene Set Enrichment analysis”, στην οποία πραγματοποιείται η διαδικασία της ανάλυσης εμπλουτισμού γονιδιακών συνόλων και παράγονται τα στατιστικά αποτελέσματα και τα αντίστοιχα διαγράμματα. Αρχεία VCF

3.2. Αρχεία VCF

Από την διαδικασία της στοίχισης των αλληλουχιών, πέραν των αρχείων “.BAM” και “.SAM”, προκύπτουν και τα αρχεία “.vcf”. Το “Variant Call Format” (VCF) καθορίζει τη μορφή ενός αρχείου κειμένου που χρησιμοποιείται στη βιοπληροφορική για την αποθήκευση δεδομένων γενετικών παραλλαγών. Χρησιμοποιείται συνήθως στη γονιδιωματική έρευνα και είναι ιδιαίτερα χρήσιμο για την αναπαράσταση και την ανταλλαγή πληροφοριών σχετικά με γενετικές παραλλαγές, όπως πολυμορφισμοί ενός νουκλεοτιδίου (SNPs), παρεμβολές, διαγραφές και άλλες δομικές παραλλαγές. Τα αρχεία VCF είναι αρχεία απλού κειμένου με συγκεκριμένη μορφή που περιλαμβάνει ένα τμήμα κεφαλίδας και γραμμές δεδομένων. Ακολουθούν τα κύρια συστατικά στοιχεία ενός αρχείου VCF:

Κεφαλίδα: Το τμήμα κεφαλίδας παρέχει πληροφορίες σχετικά με το αρχείο, συμπεριλαμβανομένης της πηγής των δεδομένων, πληροφορίες για το γονιδίωμα αναφοράς και διάφορα μεταδεδομένα. Περιέχει γραμμές που αρχίζουν με το σύμβολο “##”, οι οποίες ορίζουν πεδία μεταδεδομένων και τις τιμές τους. Η επικεφαλίδα περιλαμβάνει επίσης τα ονόματα των στηλών και τις περιγραφές τους.

Σειρές δεδομένων: Οι γραμμές δεδομένων αντιπροσωπεύουν μεμονωμένες γενετικές παραλλαγές που παρατηρούνται σε ένα δείγμα. Κάθε γραμμή αντιστοιχεί σε μια γονιδιωματική θέση και περιέχει πληροφορίες σχετικά με τη μεταβολή στη συγκεκριμένη θέση.

Στήλες δείγματος: Οι στήλες περιέχουν πληροφορίες γονότυπου για κάθε δείγμα, που συνήθως αντιπροσωπεύονται από κωδικούς γονότυπου ή αριθμούς αλληλομόρφων.

Τα αρχεία “VCF” μπορούν να περιέχουν πολλαπλά δείγματα, όπου κάθε δείγμα έχει το δικό του σύνολο πληροφοριών γονότυπου. Αυτό επιτρέπει στους ερευνητές να συγκρίνουν και να αναλύουν γενετικές παραλλαγές μεταξύ ατόμων ή πληθυσμών. Τα αρχεία “VCF” χρησιμοποιούνται ευρέως στη γενετική έρευνα για εργασίες όπως η κλήση παραλλαγών, η γενετική πληθυσμών και οι μελέτες συσχέτισης. Παρέχουν μια ευέλικτη και τυποποιημένη μορφή για την αποθήκευση και την ανταλλαγή δεδομένων γενετικών παραλλαγών, επιτρέποντας στους ερευνητές να αναλύουν και να ερμηνεύουν αποτελεσματικότερα τις γονιδιωματικές πληροφορίες.

3.1. Στατιστικά αποτελέσματα

Στην ανάλυση διαφορικής έκφρασης με το “DESeq2”, τα “baseMean”, “log2FoldChange”, “lfcSE”, “stat”, “pvalue” και “adj” είναι βασικά στατιστικά αποτελέσματα που παρέχουν πληροφορίες σχετικά με τη διαφορική γονιδιακή έκφραση. Ακολουθεί επεξήγηση για καθένα από αυτά τα αποτελέσματα:

Το “baseMean” αντιπροσωπεύει το μέσο επίπεδο γονιδιακής έκφρασης σε όλα τα δείγματα της ανάλυσης. Παρέχει μια εκτίμηση του συνολικού επιπέδου έκφρασης του γονιδίου και υπολογίζεται με βάση τις κανονικοποιημένες τιμές έκφρασης, συνήθως χρησιμοποιώντας τον γεωμετρικό μέσο όρο ή τη διάμεσο.

Το “log2FoldChange” αντιπροσωπεύει τη μετασχηματισμένη κατά τον δεύτερο λογάριθμο ($\log_2$) αναδίπλωση της γονιδιακής έκφρασης μεταξύ δύο συνθηκών. Υποδεικνύει το μέγεθος και την κατεύθυνση της αλλαγής της γονιδιακής έκφρασης. Μια θετική τιμή “log2FoldChange” υποδηλώνει ανοδική ρύθμιση ή υπερέκφραση (υψηλότερη έκφραση στην πρώτη συνθήκη), ενώ μια αρνητική τιμή υποδηλώνει καθοδική ρύθμιση ή υποέκφραση (χαμηλότερη έκφραση στην πρώτη συνθήκη).

Το “lfcSE” είναι το τυπικό σφάλμα που σχετίζεται με την εκτίμηση της τιμής “log2FoldChange”. Αντιπροσωπεύει την αβεβαιότητα ή τη μεταβλητότητα της τιμής “log2FoldChange”. Μία χαμηλή τιμή “lfcSE” υποδηλώνει μια πιο ακριβή και αξιόπιστη εκτίμηση της τιμής “log2FoldChange”.

Το “stat”, επίσης γνωστό ως στατιστικό Wald, υπολογίζεται με βάση το “log2FoldChange” και το τυπικό σφάλμα της (lfcSE). Το στατιστικό Wald ακολουθεί κανονική κατανομή και χρησιμοποιείται για τον υπολογισμό των τιμών “p-values” για τη διαφορική έκφραση.

Η τιμή “p-value” αντιπροσωπεύει τη στατιστική σημαντικότητα της παρατηρούμενης “log2FoldChange”. Μετρά την πιθανότητα να παρατηρηθεί μια “log2FoldChange” εξίσου ακραία ή πιο ακραία από την υπολογιζόμενη, υποθέτοντας ότι δεν υπάρχει διαφορική έκφραση. Μια χαμηλή τιμή “p-value” υποδηλώνει μεγαλύτερη εμπιστοσύνη στην παρουσία διαφορικής έκφρασης. Οι τιμές που μπορεί να πάρει είναι από 0 (απολύτως βέβαιο) έως 1 (αβέβαιο).

Το “padj”, είναι μια διόρθωση για τον έλεγχο πολλαπλών υποθέσεων. Προσαρμόζει τις τιμές “p-values” για τον έλεγχο του αναμενόμενου ποσοστού ψευδώς θετικών αποτελεσμάτων μεταξύ των σημαντικών αποτελεσμάτων. Οι τιμές “padj” βοηθούν στον προσδιορισμό της σημαντικότητας της διαφορικής έκφρασης μετά τον συνυπολογισμό των πολλαπλών δοκιμών. Ένα χαμηλό “padj” υποδηλώνει υψηλότερο επίπεδο εμπιστοσύνης στη διαφορική έκφραση. Οι τιμές που μπορεί να πάρει είναι από 0 (απολύτως βέβαιο) έως 1 (αβέβαιο).

Αυτά τα στατιστικά αποτελέσματα παρέχουν συλλογικά πληροφορίες σχετικά με την κατάσταση διαφορικής έκφρασης των γονιδίων. Το “log2FoldChange” και το σχετικό τυπικό σφάλμα (lfcSE) παρέχουν εκτιμήσεις του μεγέθους και της αβεβαιότητας των αλλαγών έκφρασης. Η τιμή “p-value” και η προσαρμοσμένη τιμή “p-value” (padj) υποδεικνύουν τη στατιστική σημαντικότητα της διαφορικής έκφρασης. Οι ερευνητές μπορούν να χρησιμοποιήσουν αυτά τα αποτελέσματα για τον εντοπισμό και την ιεράρχηση γονιδίων που εκφράζονται σημαντικά διαφορετικά μεταξύ των συνθηκών, βοηθώντας στην ερμηνεία και την κατανόηση των υποκείμενων βιολογικών διεργασιών.

Στην ανάλυση εμπλουτισμού γονιδιακών συνόλων με τη βοήθεια της συνάρτησης “gseGO” του πακέτου “clusterProfiler” βασικά στατιστικά αποτελέσματα που προκύπτουν είναι:

Το “ONTOLOGY” υποδεικνύει τον τύπο οντολογίας που σχετίζεται με το σύνολο γονιδίων. Καθορίζει τη βάση δεδομένων οντολογίας που χρησιμοποιήθηκε, όπως η γονιδιακή οντολογία (GO) (που εμφανίζεται στα αποτελέσματα) και αποτελεί μοναδικό αναγνωριστικό για κάθε σύνολο γονιδίων ή λειτουργική κατηγορία. Οι όροι “GO” ταξινομούνται σε τρία υποομάδες: τις Βιολογικές διεργασίες (BP), τις μοριακές λειτουργίες (MF) και τα κυτταρικά συστατικά (CC).

Η στήλη “Description” παρέχει μια σύντομη περιγραφή ή όνομα για το σύνολο γονιδίων ή τη λειτουργική κατηγορία. Περιγράφει τη βιολογική διαδικασία, τη μοριακή λειτουργία, το κυτταρικό συστατικό ή το μονοπάτι που σχετίζεται με το σύνολο γονιδίων.

Το “setSize” αντιπροσωπεύει τον αριθμό των γονιδίων από τον κατάλογο γονιδίων εισόδου που έχουν σχολιαστεί στο αντίστοιχο σύνολο γονιδίων ή στη λειτουργική κατηγορία. Δείχνει το μέγεθος ή την κάλυψη του συνόλου γονιδίων στα δεδομένα εισόδου.

Το “enrichmentScore” μετρά τον βαθμό εμπλουτισμού του συνόλου γονιδίων ή της λειτουργικής κατηγορίας στον κατάλογο γονιδίων εισόδου. Βασίζεται στην κατάταξη ή την ταξινόμηση των γονιδίων στον κατάλογο εισόδου και αντανακλά τη θέση των γονιδίων στο σύνολο γονιδίων εντός του καταταγμένου καταλόγου.

Το “NES” είναι μια κανονικοποιημένη έκδοση του σκορ εμπλουτισμού. Προσαρμόζει τις διαφορές στο μέγεθος και τη διακύμανση του συνόλου γονιδίων και παρέχει ένα πιο αξιόπιστο μέτρο εμπλουτισμού. Το “NES” επιτρέπει την άμεση σύγκριση των βαθμολογιών εμπλουτισμού μεταξύ διαφορετικών συνόλων γονιδίων.

Το “pvalue” αντιπροσωπεύει τη στατιστική σημαντικότητα του εμπλουτισμού του γονιδιακού συνόλου ή της λειτουργικής κατηγορίας.

Η στήλη “padjust” παρέχει προσαρμοσμένες τιμές “p-values”, οι οποίες διορθώνονται για τον έλεγχο πολλαπλών υποθέσεων. Οι συνήθεις μέθοδοι διόρθωσης περιλαμβάνουν τη διόρθωση Benjamini-Hochberg ή Bonferroni. Οι προσαρμοσμένες τιμές “p-value” βοηθούν στον έλεγχο του ποσοστού ψευδούς ανακάλυψης (FDR) ή του ποσοστού σφάλματος (FWER) κατά την αξιολόγηση της σημαντικότητας του εμπλουτισμού.

Το “qvalue» είναι ένα άλλο μέτρο στατιστικής σημαντικότητας που είναι παρόμοιο με την προσαρμοσμένη τιμή “p-value” και αντιπροσωπεύει το ελάχιστο “FDR” στο οποίο μια δεδομένη δοκιμή θεωρείται σημαντική.

Το “rank” υποδεικνύει την κατάταξη ή τη σειρά του συνόλου γονιδίων εντός του καταλόγου γονιδίων εισόδου με βάση τη βαθμολογία εμπλουτισμού. Η κατάταξη παρέχει πληροφορίες σχετικά με τη θέση του συνόλου γονιδίων σε σχέση με άλλα σύνολα γονιδίων όσον αφορά τον εμπλουτισμό τους.

Η στήλη “leading edge” παρέχει πληροφορίες σχετικά με τα γονίδια που συμβάλλουν πιο σημαντικά στον εμπλουτισμό του συνόλου γονιδίων.

Η στήλη “core enrichment” αντιπροσωπεύει ένα υποσύνολο γονιδίων εντός του συνόλου γονιδίων που σχετίζονται πιο σημαντικά με τον εμπλουτισμένο φαινότυπο ή κατάσταση. Αυτά τα γονίδια του πυρήνα θεωρούνται οι βασικοί συντελεστές του σήματος εμπλουτισμού και μπορούν να παράσχουν πληροφορίες για διάφορες βιολογικές διεργασίες.

Αυτά τα αποτελέσματα παρέχουν συλλογικά πληροφορίες σχετικά με την κατάσταση εμπλουτισμού, τη στατιστική σημαντικότητα και τη βιολογική συνάφεια γονιδιακών συνόλων ή λειτουργικών κατηγοριών. Βοηθούν τους ερευνητές να εντοπίσουν και να ερμηνεύσουν τα εμπλουτισμένα σύνολα γονιδίων.

3.2. Οπτικοποίηση αποτελεσμάτων

3.2.1. Διαγράμματα της ανάλυσης διαφορικής έκφρασης

Σε όλα τα διαγράμματα της παρούσας εφαρμογής, εκτός από το διάγραμμα ανάλυσης κυρίων συνιστωσών (Principal Component Analysis, PCA) και το διάγραμμα θερμικού χάρτη (Heatmap), δίνεται η δυνατότητα στον χρήστη να επιλέξει τα χρώματα των διαγραμμάτων βάσει των εκάστοτε αναγκών και προτιμήσεων του. Ακολουθούν αναλυτικά οι τύποι των διαγραμμάτων που προκύπτουν με το πέρας των στατιστικών αναλύσεων.

3.2.1.1. Ηφαιστειόγραμμα (Volcano plot)

Στο συγκεκριμένο διάγραμμα παρουσιάζεται το σύνολο των εκφραζόμενων γονιδίων τα οποία αναπαρίστανται ως κουκίδες. Οι τιμές των επιπέδων διαφορικής έκφρασης ($\log_2$ fold change) απεικονίζονται στον άξονα x και οι αρνητικοί λογάριθμοι των τιμών p-value ($-\log_{10}$

(p-value)) απεικονίζονται στον άξονα γ'γ. Η οριζόντια γραμμή αντιπροσωπεύει το όριο σημαντικότητας που καθορίζεται στην ανάλυση ($p\text{ value} = 0.05$). Τα γονίδια των οποίων η έκφραση είναι μειωμένη σε σχέση με την ομάδα σύγκρισης βρίσκονται αριστερά του μηδενός στον άξονα x'x, ενώ τα γονίδια των οποίων η έκφραση είναι αυξημένη απεικονίζονται δεξιά του μηδενός. Τα γονίδια με στατιστικά σημαντική διαφορετική έκφραση βρίσκονται πάνω από την οριζόντια γραμμή. Η προσέγγιση προς το μηδέν υποδηλώνει μικρότερη μεταβολή, ενώ η απομάκρυνση από το μηδέν προς οποιαδήποτε κατεύθυνση υποδηλώνει μεγαλύτερη μεταβολή. Τα ηφαιστειογράμματα παρέχουν ένα αποτελεσματικό μέσο για την απεικόνιση της κατεύθυνσης, του μεγέθους και της σημασίας των αλλαγών στη γονιδιακή έκφραση.



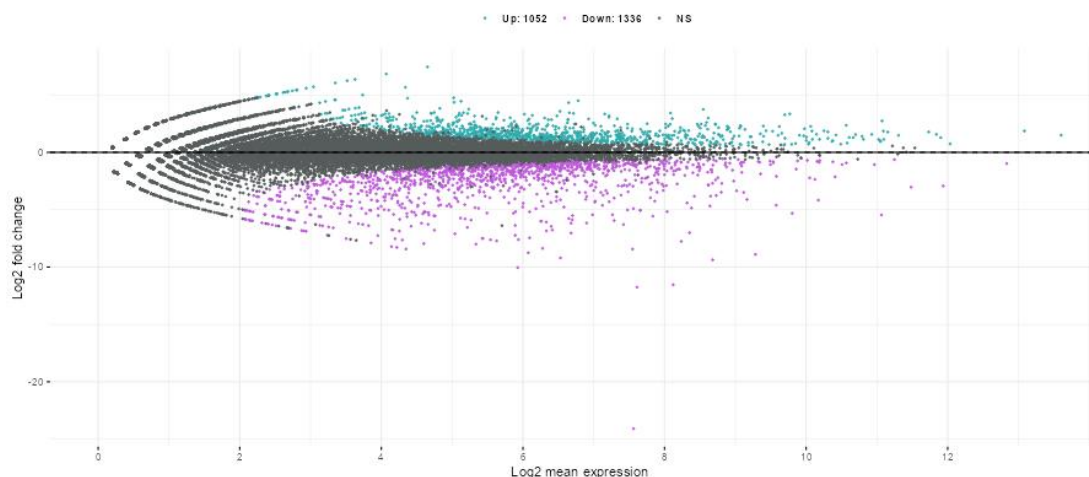
Εικόνα 10: “Volcano plot” στο οποίο παρουσιάζονται διαφορικά εκφραζόμενα γονίδια από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι) σε συνθήκες ελέγχου και ξηρασίας. Τα γονίδια που είναι ρυθμισμένα ανοδικά είναι όσα έχουν τιμή “log2Foldchange” > 1 και τιμή “pvalue” $< 0,05$. Τα γονίδια που είναι ρυθμισμένα καθοδικά είναι όσα έχουν τιμή “log2Foldchange” < -1 και τιμή “pvalue” $< 0,05$. Τα υπέρ εκφραζόμενα γονίδια παρουσιάζονται με μωβ χρώμα και τα υπό εκφραζόμενα με μπλε χρώμα, ενώ αυτά που δεν είναι στατιστικά σημαντικά παρουσιάζονται με γκρι χρώμα.

Ηφαιστειογράμματα ηφαιστείου όπως το παραπάνω (Εικόνα 4) είναι χρήσιμα όταν υπάρχουν πολλές παρατηρήσεις με μεγάλο εύρος διαφορών, τόσο θετικών όσο και αρνητικών. Σε τέτοιες περιπτώσεις το διάγραμμα παρουσιάζει ένα πυκνό, συμμετρικό σχήμα “V”. Όταν ο αριθμός των παρατηρήσεων μειώνεται, η διακύμανση της απόκρισης δεν είναι τόσο ομοιόμορφα κατανομημένη.

3.2.1.2. Διάγραμμα διασποράς (MA plot)

Το διάγραμμα MA είναι ένα ευρέως χρησιμοποιούμενο εργαλείο οπτικοποίησης στην ανάλυση διαφορικής έκφρασης, το οποίο αποσκοπεί στον εντοπισμό γονιδίων που εκφράζονται διαφορετικά μεταξύ ποικίλων συνθηκών. Το διάγραμμα MA βοηθά στην αξιολόγηση του μεγέθους και της σημασίας των αλλαγών γονιδιακής έκφρασης και στον εντοπισμό των γονιδίων ενδιαφέροντος. Σε ένα διάγραμμα MA, στον άξονα x'x παρουσιάζεται το μέσο επίπεδο έκφρασης (M) ($\log_2$ mean expression) του κάθε γονιδίου, ενώ στον άξονα γ'γ παρουσιάζεται η λογαριθμισμένη μεταβολή (A) ($\log_2$ fold change) μεταξύ των δύο συγκρινόμενων συνθηκών.

Το διάγραμμα MA επιτρέπει την οπτική αναπαράσταση της κατάστασης διαφορικής έκφρασης των γονιδίων. Κάθε γονίδιο απεικονίζεται ως ένα σημείο και τα γονίδια με παρόμοια επίπεδα έκφρασης μεταξύ των συνθηκών συγκεντρώνονται γύρω από τον άξονα x'x ($M = 0$). Τα διαφορετικά εκφραζόμενα γονίδια απεικονίζονται ως σημεία που αποκλίνουν από τον άξονα x'x, υποδεικνύοντας ανοδική ή καθοδική ρύθμιση (upregulation - downregulation). Το διάγραμμα MA μπορεί να βοηθήσει στον εντοπισμό γονιδίων που παρουσιάζουν σημαντική διαφορική έκφραση. Τα σημεία που βρίσκονται μακριά από τον άξονα x'x ($M = 0$) και τοποθετούνται προς το άνω ή το κάτω άκρο του άξονα γ'γ (υποδεικνύοντας υψηλή μεταβολή) είναι πιθανότερο να αντιπροσωπεύουν γονίδια των οποίων η έκφραση παρουσιάζει στατιστικώς σημαντικές διαφορές. Επιπλέον, με την εξέταση της κατανομής και της διασποράς των σημείων στο διάγραμμα MA, οι ερευνητές μπορούν να αξιολογήσουν τη συνολική ποιότητα των δεδομένων. Μια στενή συστάδα σημείων γύρω από τον άξονα x'x υποδηλώνει υψηλή λογαριθμισμένη μεταβολή ($\log$ fold change), ενώ μια ευρύτερη διασπορά σημείων μπορεί να υποδηλώνει μεγαλύτερη μεταβλητότητα ή χαμηλότερη ποιότητα δεδομένων.¹⁴



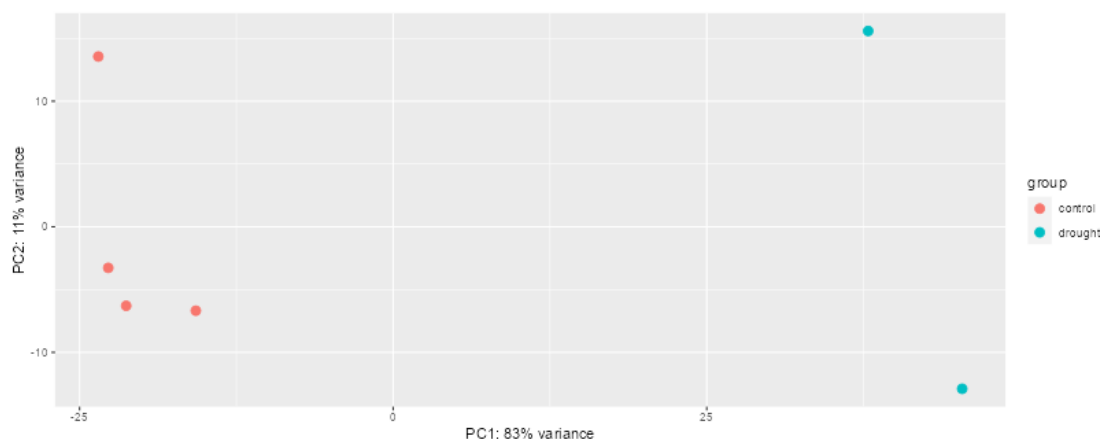
Εικόνα 11: “MA plot” στο οποίο παρουσιάζονται διαφορετικά εκφραζόμενα γονίδια από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι) σε συνθήκες ελέγχου και ξηρασίας. Μόνο τα γονίδια με τιμή “fold change” ≥ 1 and τιμή “*padj*” $\leq 0,05$ θεωρούνται σημαντικά διαφορετικά εκφραζόμενα. Τα υπέρ εκφραζόμενα γονίδια παρουσιάζονται με μπλε χρώμα και τα υπό εκφραζόμενα με κόκκινο χρώμα, ενώ αυτά που δεν είναι στατιστικά σημαντικά παρουσιάζονται με γκρι χρώμα.

3.1.2.3. Ανάλυση κύριων συνιστωσών (PCA plot)

Η ανάλυση κύριων συνιστωσών (PCA) είναι μια κλασική προσέγγιση μείωσης των διαστάσεων. Τα δεδομένα γονιδιακής έκφρασης αποτελούνται συνήθως από μεγάλο αριθμό μεταβλητών (γονίδια) που μετρήθηκαν σε πολλαπλά δείγματα ή συνθήκες. Η PCA βοηθά στη μείωση της διαστατικότητας των δεδομένων προβάλλοντάς τα σε έναν χώρο λιγότερων διαστάσεων, διατηρώντας παράλληλα όσο το δυνατόν περισσότερες πληροφορίες. Αυτό επιτρέπει την οπτικοποίηση των δεδομένων σε δύο διαστάσεις, οι οποίες είναι πιο εύκολα ερμηνεύσιμες. Σε ένα διάγραμμα PCA, κάθε δείγμα αναπαρίσταται ως ένα σημείο στον χώρο μειωμένων διαστάσεων. Δείγματα με παρόμοια έκφραση θα συγκεντρωθούν μεταξύ τους, ενώ εκείνα με διαφορετικά πρότυπα έκφρασης θα απέχουν περισσότερο. Έτσι οι ερευνητές μπορούν να αποκτήσουν πληροφορίες για τη συνολική δομή των δεδομένων, εντοπίζοντας πιθανές ομαδοποιήσεις ή τάσεις. Επιπλέον οι ακραίες τιμές, που μπορούν να διαστρεβλώσουν την ερμηνεία της ανάλυσης διαφορικής έκφρασης, τοποθετούνται μακριά από την κύρια ομάδα δειγμάτων. Με την εξέταση του διαγράμματος PCA, οι ερευνητές μπορούν να εντοπίσουν τέτοια δείγματα και να αξιολογήσουν το αντίκτυπό τους στην ανάλυση. Εκτός αυτού, η κατανομή και η διασπορά των δειγμάτων στο διάγραμμα PCA μπορεί να παρέχει πληροφορίες σχετικά με την ποιότητα των δεδομένων γονιδιακής έκφρασης. Καλά διαχωρισμένα και σφιχτά ομαδοποιημένα δείγματα υποδεικνύουν καλή

ποιότητα δεδομένων με χαμηλό θόρυβο και αξιόπιστα μοτίβα. Αντίθετα, διάσπαρτα ή επικαλυπτόμενα δείγματα μπορεί να υποδηλώνουν υψηλότερη μεταβλητότητα.

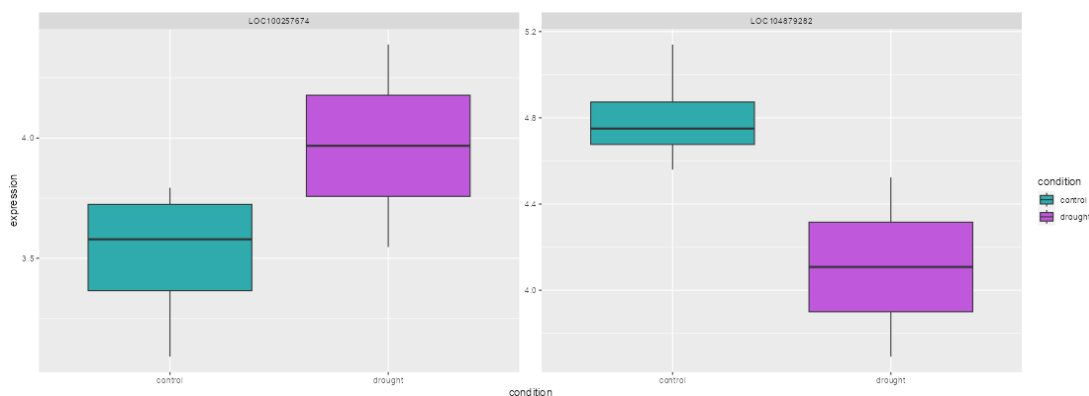
Στην εφαρμογή δίνεται η δυνατότητα στον χρήστη να επιλέξει την ομάδα (από τις συνθήκες στο `rphenodata`) βάση της οποίας θέλει να γίνει η οπτικοποίηση.



Εικόνα 12: “PCA plot” στο οποίο παρουσιάζονται ποσοτικοποιημένα γονίδια (*counts*) από δείγματα του οργανισμού “*vitis vinifera*” (αμπέλι) σε συνθήκες ελέγχου (με κόκκινο χρώμα) και ξηρασίας (με μπλε χρώμα) μετά από κανονικοποίηση “*vst*” (*varianceStabilizingTransformation*).

3.1.2.4. Θηκόγραμμα (Boxplot) για τη γονιδιακή έκφραση

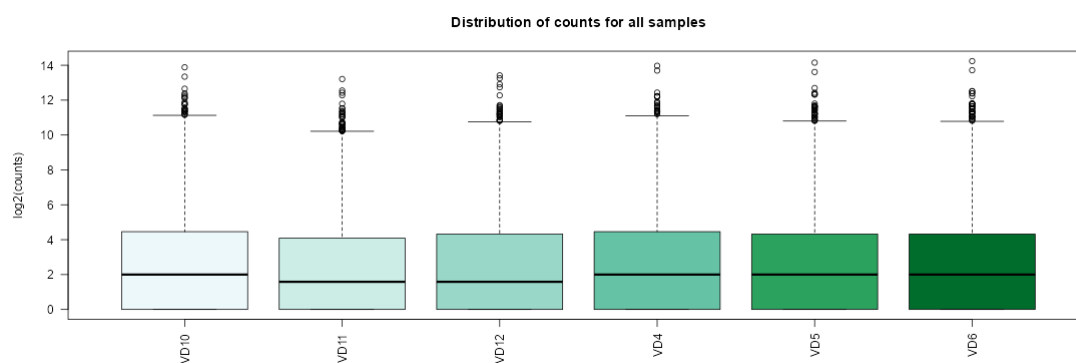
Στο θηκόγραμμα παρουσιάζεται η κατανομή των επιπέδων έκφρασης ενός γονιδίου μεταξύ δύο συνθηκών. Στον άξονα x’x αναπαρίστανται κάθε συνθήκη ή ομάδα και στον άξονα y’y τα επίπεδα έκφρασης του εκάστοτε γονιδίου. Παρέχει πληροφορίες σχετικά με την κατανομή και τη διακύμανση των επιπέδων έκφρασης μεταξύ των συνθηκών και επιτρέπει τον εντοπισμό των διαφορών στα επίπεδα έκφρασης, καθώς και των διαφορών στη διασπορά ή τη μεταβλητότητα των δεδομένων. Συγκρίνοντας τα boxplots διαφορετικών γονιδίων (επιλέγοντας ποια θα οπτικοποιηθούν κάθε φορά), οι χρήστες μπορούν να εντοπίσουν εκείνα που εκφράζονται σημαντικά διαφορετικά μεταξύ των συνθηκών (Εικόνα 7).



Εικόνα 13: “Boxplot” στο οποίο παρουσιάζονται διαφορετικά εκφραζόμενα σε συνθήκες ελέγχου και ξηρασίας από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι). Στον άξονα x’x αναπαρίσταται κάθε συνθήκη (με μωβ η κατάσταση ξηρασίας και με μπλε η κατάσταση ελέγχου), στον άξονα y’y τα επίπεδα έκφρασης του εκάστοτε γονιδίου.

3.1.2.5. Θηκόγραμμα (Boxplot) για την κατανομή καταμέτρησης επιπέδων έκφρασης μεταξύ διαφορετικών δειγμάτων

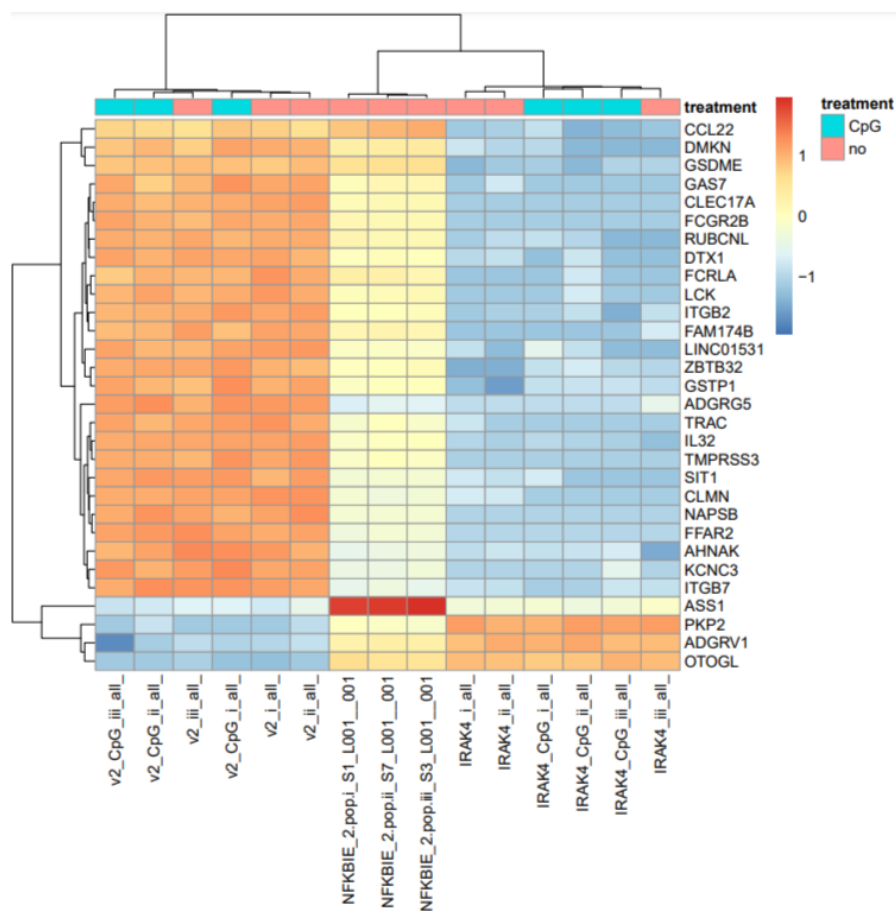
Με τη χρήση πολλαπλών boxplots μαζί, καθίσταται δυνατή η σύγκριση των κατανομών της γονιδιακής έκφρασης μεταξύ διαφορετικών δειγμάτων (Εικόνα 8). Κάθε «κουτί»/«θήκη» (box) αντιπροσωπεύει ένα διαφορετικό δείγμα, επιτρέποντας στους ερευνητές να αξιολογήσουν τις διαφοροποιήσεις στις κατανομές μεταξύ τους. Το ύψος των θηκών μπορεί να παρέχει μια εκτίμηση των συνολικών επιπέδων γονιδιακής έκφρασης ή και της κεντρικής τάσης σε κάθε δείγμα. Οι ακραίες τιμές, οι οποίες είναι τιμές μέτρησης που αποκλίνουν σημαντικά από την πλειονότητα των παρατηρήσεων, μπορούν εύκολα να εντοπιστούν με τη χρήση ενός boxplot. Η παρουσία τους μπορεί να υποδηλώνει σφάλματα κατά την απόκτηση δεδομένων, τη μέτρηση ή την επιμόλυνση του δείγματος και θα πρέπει να εξετάζονται προσεκτικά για να διαπιστωθεί αν πρόκειται για πραγματικές βιολογικές ή τεχνικές ανωμαλίες.



Εικόνα 14: “Boxplot” στο οποίο παρουσιάζεται η κατανομή των ποσοτικοποιημένων γονιδίων από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι).

3.1.2.6. Διάγραμμα θερμικού χάρτη (heatmap)

Μια διαδεδομένη ακόμα μέθοδος οπτικοποίησης δεδομένων γονιδιακής έκφρασης, είναι η απεικόνισή τους ως θερμικός χάρτης (heatmap) (Εικόνα 7). Στους θερμικούς χάρτες τα δεδομένα εμφανίζονται σε ένα πλέγμα όπου κάθε γραμμή αντιπροσωπεύει ένα γονίδιο και κάθε στήλη αντιπροσωπεύει ένα δείγμα. Η ένταση του χρώματος στον χάρτη θερμότητας αντιπροσωπεύει το επίπεδο έκφρασης κάθε γονιδίου ή χαρακτηριστικού, με τα υψηλότερα επίπεδα έκφρασης να εμφανίζονται με πιο έντονα χρώματα. Οι Heatmaps επιτρέπουν τον εντοπισμό μοτίβων ή συστάδων γονιδίων με παρόμοια πρότυπα έκφρασης σε διάφορα δείγματα ή συνθήκες. Παρέχουν μια σφαιρική εικόνα των μεταβολών της γονιδιακής έκφρασης, επιτρέποντας στους ερευνητές να παρατηρούν τις συνολικές τάσεις και να εντοπίζουν διαφορετικά εκφραζόμενα γονίδια που παρουσιάζουν διακριτά πρότυπα έκφρασης μεταξύ συνθηκών. Στο heatmap της εφαρμογής (Εικόνα 8), εμφανίζονται τα 30 σημαντικότερα διαφορικά εκφραζόμενα γονίδια. Επίσης ο χρήστης μπορεί να επιλέξει τη συνθήκη ή τις συνθήκες βάσει των οποίων θα γίνει η οπτικοποίηση.



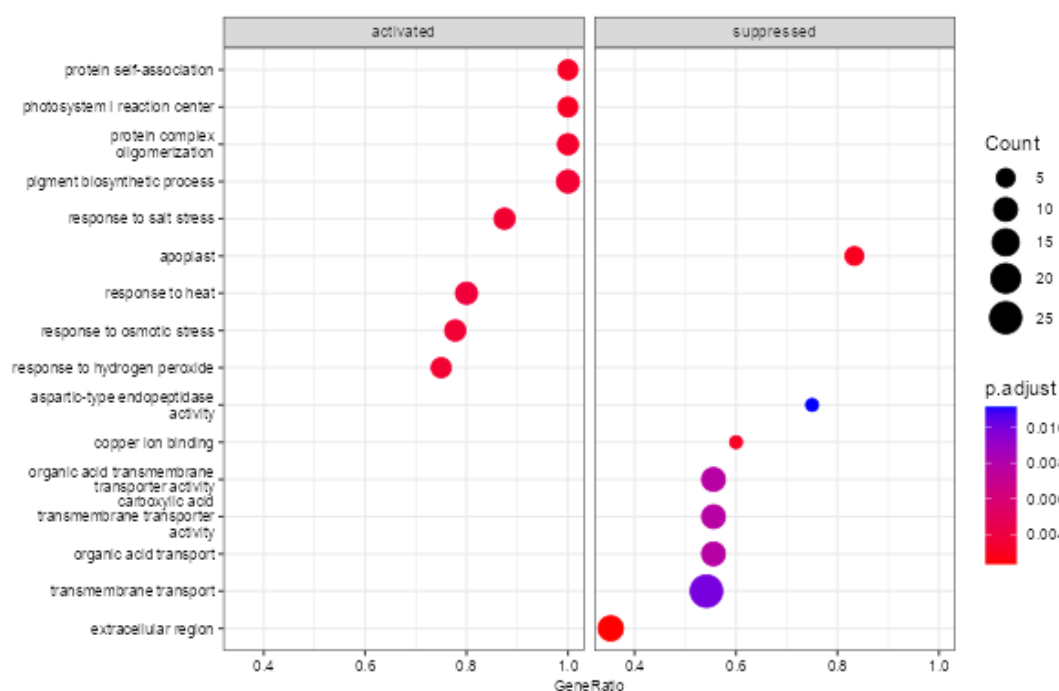
Εικόνα 15: “Heatmap” στο οποίο παρουσιάζονται τα επίπεδα έκφρασης 30 γονιδίων από δείγματα από τον οργανισμό “*Homo sapiens*”. Τα ποσοτικοποιημένα γονίδια (counts) κανονικοποιούνται με τη βοήθεια του μετασχηματισμού “vst”. Η ομαδοποίηση στα αριστερά γίνεται βάσει των κοντινότερων επιπέδων έκφρασης. Η ομαδοποίηση στο πάνω μέρος του διαγράμματος γίνεται βάσει των συνθηκών που επιλέγονται κάθε φορά για οπτικοποίηση.

3.2.2. Διαγράμματα της ανάλυσης εμπλουτισμού γονιδιακών συνόλων

3.2.2.1. Σημειόγραμμα (Dotplot)

Το σημειόγραμμα (dotplot) είναι μια γραφική αναπαράσταση της ανάλυσης γονιδιακού εμπλουτισμού, η οποία χρησιμοποιείται για τον εντοπισμό βιολογικών εννοιών που υπέρ εκφράζονται ή υπό εκφράζονται, όπως οι όροι της γονιδιακής οντολογίας (GO). Αναπαριστά διαφορετικά σύνολα γονιδίων ως κουκκίδες κατά μήκος του άξονα γ’γ και εμφανίζει τις

βαθμολογίες εμπλουτισμού τους ή τη στατιστική τους σημαντικότητα στον άξονα x'x. Τα σημειογράμματα επιτρέπουν τη γρήγορη σύγκριση πολλαπλών συνόλων γονιδίων, εντοπίζοντας τα πιο σημαντικά εμπλουτισμένα με βάση τη θέση τους κατά μήκος του άξονα x'x. Επιπλέον, γίνεται χρωματική κωδικοποίηση των κουκκίδων ανάλογα με την τιμή “p.adjust” και το μέγεθος των κουκκίδων αντιστοιχεί στο επίπεδο έκφρασης των γονιδίων. Το μεγαλύτερο μέγεθος υποδηλώνει υψηλότερο επίπεδο έκφρασης. Στην παρούσα εφαρμογή, ο χρήστης μπορεί να ορίσει τον αριθμό των κατηγοριών που θα οπτικοποιηθούν.

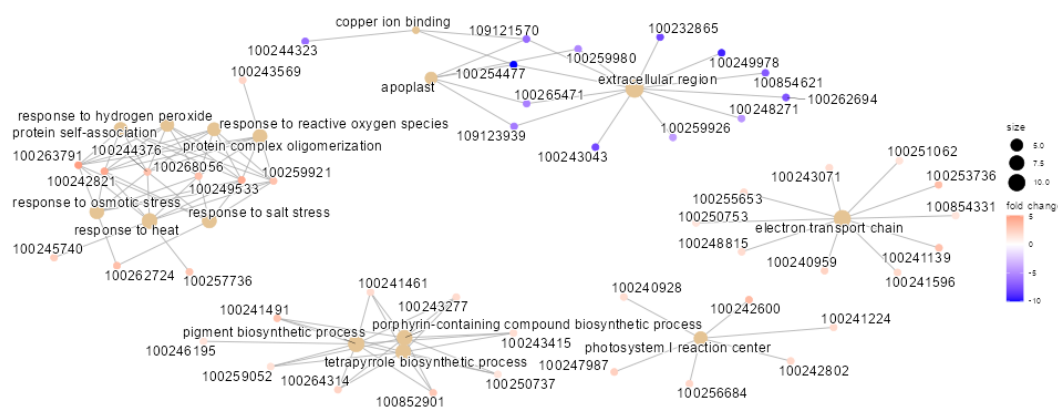


Εικόνα 15: “Dotplot” στο οποίο παρουσιάζονται 16 οντολογίες για εμπλουτισμένα σύνολα γονιδίων από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι), που υπέρ εκφράζονται ή υπό εκφράζονται αντίστοιχα. Το μέγεθος των κουκκίδων αντιστοιχεί στο επίπεδο έκφρασης των γονιδίων. Το μεγαλύτερο μέγεθος υποδηλώνει υψηλότερο επίπεδο έκφρασης και παράλληλα γίνεται χρωματική κωδικοποίηση των κουκκίδων ανάλογα με την τιμή “p.adjust”.

3.2.2.2. Κατηγορικό δικτυόγραμμα (Category Netplot)

Το κατηγορικό δικτυόγραμμα (category netplot) παρέχει μια γραφική αναπαράσταση των σχέσεων μεταξύ διαφορετικών λειτουργικών κατηγοριών ή συνόλων γονιδίων. Απεικονίζει τον τρόπο με τον οποίο οι κατηγορίες συνδέονται με βάση τα κοινά γονίδια ή τα κοινά

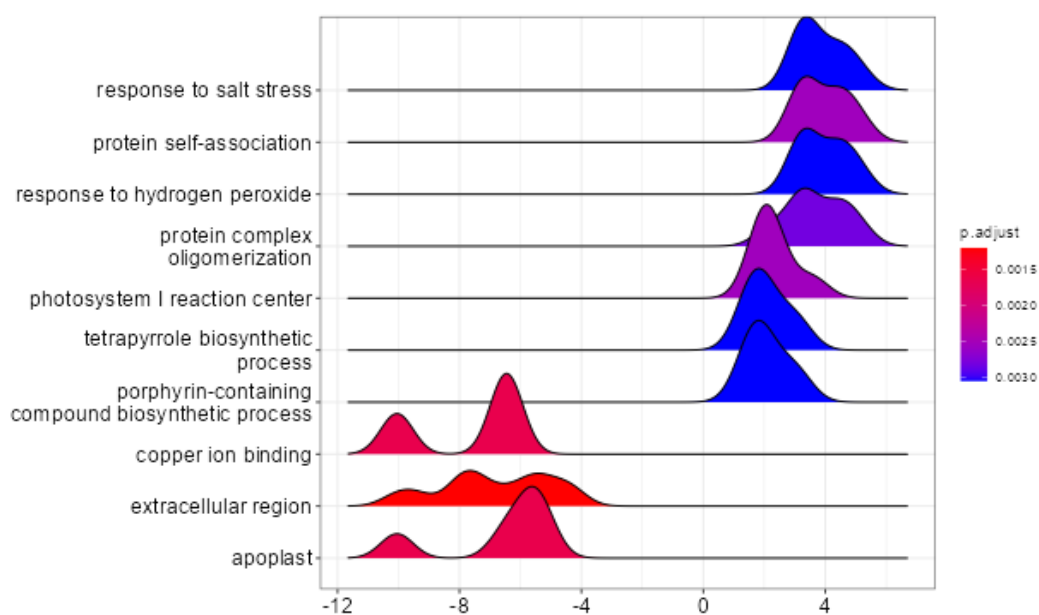
βιολογικά χαρακτηριστικά. Οι κόμβοι αντιπροσωπεύουν λειτουργικές κατηγορίες και οι ακμές τις σχέσεις μεταξύ τους. Αυτού το είδους διαγράμματα, βοηθούν στον εντοπισμό συστάδων ή ενοτήτων συναφών κατηγοριών, αναδεικνύοντας πιθανές βιολογικές αλληλεπιδράσεις ή μονοπάτια (Εικόνα 11). Στην παρούσα εφαρμογή, ο χρήστης μπορεί να ορίσει τον αριθμό των κατηγοριών που θα οπτικοποιηθούν.



Εικόνα 17: “Category Netplot” στο οποίο παρουσιάζονται 15 κόμβοι που αντιπροσωπεύουν λειτουργικές κατηγορίες για εμπλουτισμένα σύνολα γονιδίων από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι). Επίσης υπάρχει χρωματική κωδικοποίηση βάσει της λογαριθμισμένης μεταβολής ($\log$ fold change) και το μέγεθος των κουκίδων αντιστοιχεί στο επίπεδο έκφρασης των γονιδίων.

3.2.2.3. Διάγραμμα κορυφογραμμών (Ridgeplot)

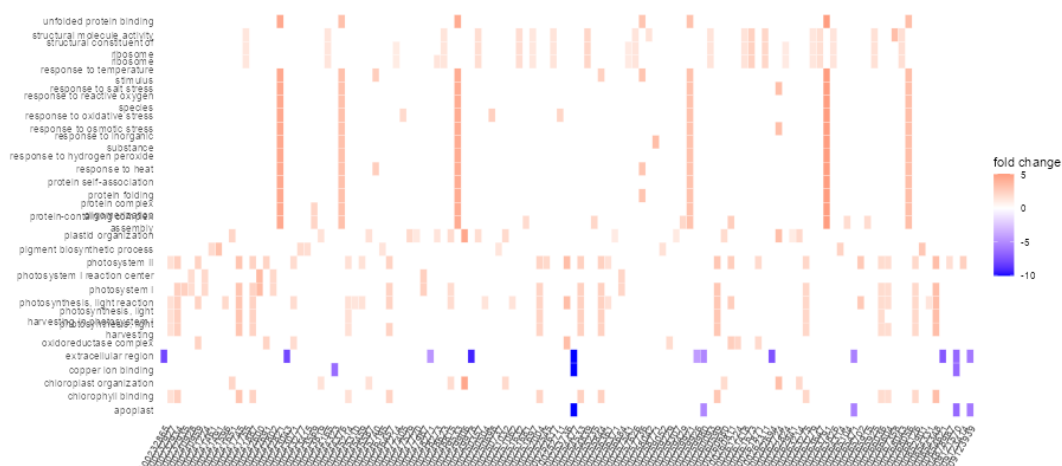
Τα διαγράμματα κορυφογραμμών (ridgeplots) είναι χρήσιμα για την οπτικοποίηση της κατανομής των βαθμολογιών εμπλουτισμού σε πολλαπλά σύνολα γονιδίων ή λειτουργικές κατηγορίες. Σε ένα τέτοιο διάγραμμα, κάθε σύνολο γονιδίων αναπαρίσταται ως διάγραμμα πυκνότητας, τα οποία στοιβάζονται κάθετα. Το πλάτος κάθε διαγράμματος πυκνότητας αντιστοιχεί στον αριθμό των γονιδίων στο σύνολο, ενώ το ύψος υποδεικνύει το σκορ εμπλουτισμού. Τα διαγράμματα κορυφογραμμών επιτρέπουν τον εντοπισμό συνόλων γονιδίων με παρόμοια προφίλ εμπλουτισμού και παρέχουν πληροφορίες για τη συνολική κατανομή των βαθμολογιών εμπλουτισμού. Σε αυτή την εφαρμογή, ο χρήστης έχει τη δυνατότητα να ορίσει τον αριθμό των κατηγοριών που θα οπτικοποιηθούν.



Εικόνα 18: “Ridgeplot” στο οποίο παρουσιάζονται 10 οντολογίες για εμπλουτισμένα σύνολα γονιδίων από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι), που υπέρ εκφράζονται ή υπό εκφράζονται αντίστοιχα. Επίσης υπάρχει χρωματική κωδικοποίηση βάσει της μεταβολής της τιμής του “*p.adjust*”.

3.2.2.4. Διάγραμμα θερμικού χάρτη εμπλουτισμού γονιδίων (Heatplot)

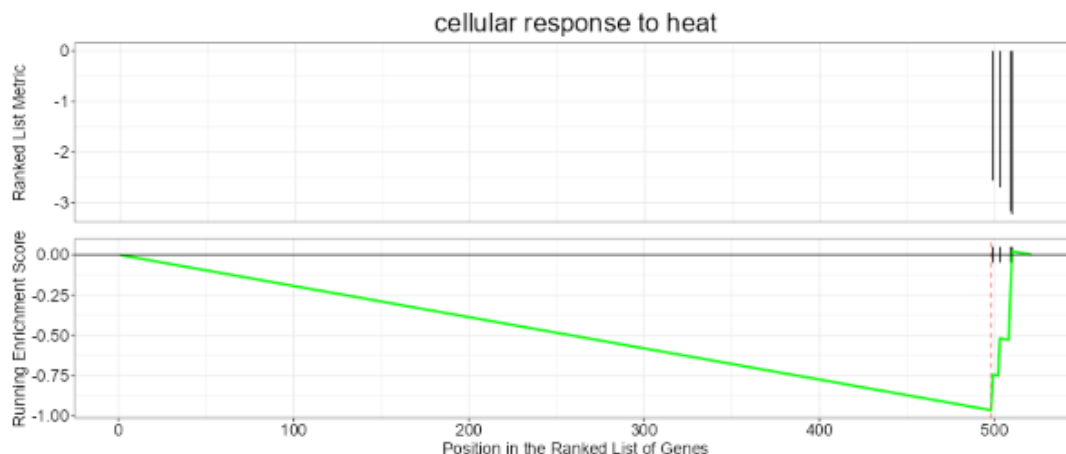
Το διάγραμμα θερμικού χάρτη (heatplot) χρησιμοποιεί έναν πίνακα με χρωματική κωδικοποίηση για την αναπαράσταση των βαθμολογιών εμπλουτισμού των γονιδιακών συνόλων σε διάφορες λειτουργικές κατηγορίες. Οι γραμμές αντιπροσωπεύουν γονιδιακά σύνολα και οι στήλες αντιπροσωπεύουν κατηγορίες. Η ένταση ή η διαβάθμιση του χρώματος αντικατοπτρίζει το μέγεθος του εμπλουτισμού, οι υψηλότερες βαθμολογίες υποδεικνύονται από πιο έντονα χρώματα. Με αυτόν τον τρόπο επιτρέπεται ο εντοπισμός εμπλουτισμένων συνόλων γονιδίων και λειτουργικών κατηγοριών, αναδεικνύοντας μοτίβα και σχέσεις εντός των δεδομένων.



Εικόνα 19: “Heatplot” στο οποίο παρουσιάζονται εμπλουτισμένα σύνολα γονιδίων από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι). Η ένταση ή η διαβάθμιση του χρώματος αντικατοπτρίζει το μέγεθος του εμπλουτισμού βάσει της λογαριθμικής μεταβολής. Οι υψηλότερες βαθμολογίες υποδεικνύονται από πιο έντονα χρώματα. Στον άξονα x’x παρατηρούμε τα ονόματα των γονιδίων και στον άξονα γ’γ παρατηρούμε τους όρους των οντολογιών.

3.2.2.5. Διάγραμμα εμπλουτισμού γονιδιακών συνόλων (GSEA)

Το διάγραμμα εμπλουτισμού γονιδιακών συνόλων (GSEA plot) παρέχει μια γραφική αναπαράσταση των αποτελεσμάτων εμπλουτισμού και αποκαλύπτει τα μοτίβα εμπλουτισμού γονιδιακών συνόλων κατά μήκος ενός καταταγμένου καταλόγου γονιδίων. Αρχικά δημιουργείται ένας ταξινομημένος κατάλογος γονιδίων με βάση μια στατιστική μετρική, όπως η διαφορική έκφραση. Τα γονίδια ταξινομούνται σε αυξανόμενη ή μειούμενη σειρά βάση της διαφορικής τους έκφρασης. Για κάθε σύνολο γονιδίων ή λειτουργική κατηγορία, υπολογίζεται ένα σκορ εμπλουτισμού με τη χρήση στατιστικών αλγορίθμων ή μεθόδων. Ο άξονας x’x αναπαριστά τη θέση στον καταταγμένο κατάλογο γονιδίων, ενώ ο άξονας γ’γ αναπαριστά την τρέχουσα βαθμολογία εμπλουτισμού. Η γραμμή ξεκινά από μηδενική βαθμολογία εμπλουτισμού και κινείται προς τα πάνω ή προς τα κάτω, ανάλογα με την κατάσταση εμπλουτισμού των συνόλων γονιδίων. Τα σημαντικά εμπλουτισμένα σύνολα γονιδίων παρουσιάζονται ως μια υψηλή κορυφή ή έχουν ανοδική τάση στο τρέχον σκορ εμπλουτισμού. Τα σύνολα γονιδίων που δεν είναι σημαντικά εμπλουτισμένα αντιπροσωπεύονται από επίπεδες περιοχές ή ελάχιστες διακυμάνσεις στην τρέχουσα βαθμολογία εμπλουτισμού.¹⁵



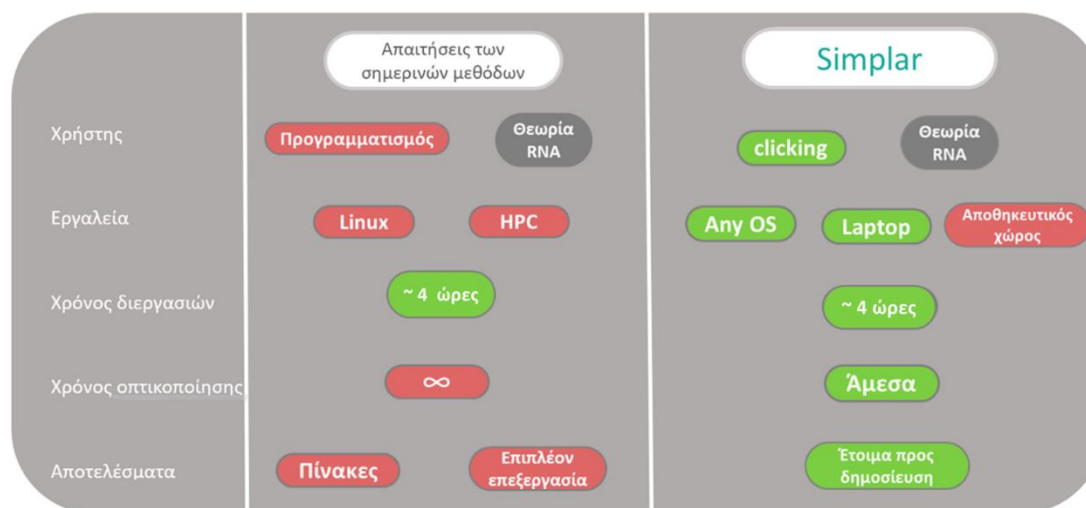
Εικόνα 20: “GSEA plot” στο οποίο παρουσιάζεται η στατιστική σημαντικότητα της οντολογίας “Cellular response to heat” στο σύνολο των εμπλουτισμένων γονιδίων από δείγματα του οργανισμού “*Vitis vinifera*” (αμπέλι).

4. Συγκριτική Αξιολόγηση

Η εφαρμογή έχει φτιαχτεί ώστε να εκτελείται στον προσωπικό υπολογιστή του χρήστη με αποτέλεσμα η απόδοσή της να εξαρτάται από την υπολογιστική δύναμη που διαθέτει κάθε χρήστης (δηλαδή το μέγεθος της CPU και της RAM). Σε περίπτωση που τα αρχεία είναι αρκετά μεγάλα και η υπολογιστική δύναμη μικρή, οι διεργασίες ενδέχεται να επηρεαστούν και να απαιτήσουν περισσότερο χρόνο, ειδικότερα αυτές του build index και του alignment που είναι έτσι και αλλιώς χρονοβόρες. Επιπλέον παρατηρήθηκαν διαφορές στη χρονική διάρκεια της επεξεργασίας των δεδομένων, ανάλογα και με τον οργανισμό της ανάλυσης. Δηλαδή οργανισμοί όπως ο άνθρωπος και κάποια μοντέλα οργανισμοί απαιτούν περισσότερο χρόνο καθώς διαθέτουν πολύ μεγάλα δεδομένα.

Για την συγκριτική αξιολόγηση της εφαρμογής (benchmarking), χρησιμοποιήθηκε, υπολογιστής με 6 υπολογιστικούς πυρήνες (cpu) και 16 GB μνήμης τυχαίας προσπέλασης (RAM).

Επίσης, για την σύγκριση του σε σχέση με τις αναλύσεις που λαμβάνουν χώρα σε κατάλληλα διαμορφωμένους εξυπηρετητές (servers), αξιολογήθηκαν οι χρόνοι που απαιτούνται για μια ολοκληρωμένη ανάλυση στις υποδομές του Ινστιτούτο Εφαρμοσμένων Βιοεπιστημών, το οποίο στο σύνολο διαθέτει 268 υπολογιστικούς πυρήνες (cpu) και 2,6 TB μνήμης τυχαίας προσπέλασης (RAM).



Εικόνα 21: Σύγκριση των πλεονεκτημάτων και των μειονεκτημάτων της εφαρμογής Simplar σε σχέση με τις υπόλοιπες μεθόδους που χρησιμοποιούνται για τις αναλύσεις RNA-seq.

4.1. Χρονικές διαφορές στην ανάλυση βάσει οργανισμού και μεγέθους αρχείων

4.1.1. Δεδομένα από τον οργανισμό *Vitis vinifera* (αμπέλι)

Το σύνολο των δεδομένων περιλάμβανε 12 ακατέργαστα αρχεία, εμπρόσθιων και ανάστροφων αναγνώσεων σε ζεύγη (R1/R2). Το κάθε αρχείο είχε μέγεθος από 640 MB μέχρι 1,3 GB. Για κάθε μία από τις διαδικασίες της στοίχισης και της ποσοτικοποίησης χρησιμοποιήθηκαν 12 νήματα ("threads").

Φόρτωση αρχείων: 45 λεπτά συνολικά

Διαδικασία φιλτραρίσματος: 60 λεπτά συνολικά

Κατασκευή ευρετηρίου: 10 λεπτά

Διαδικασία στοίχισης: 90 λεπτά συνολικά

Διαδικασία ποσοτικοποίησης: 3 λεπτά

Διαδικασία διαφορικής έκφρασης: 3 λεπτά

Διαδικασία ανάλυσης γονιδιακού εμπλουτισμού: 3 λεπτά

Σύνολο: 3 ώρες και 35 λεπτά

4.1.2. Δεδομένα από τον μοντέλο οργανισμό *Homo sapiens* (άνθρωπος)

Το σύνολο των δεδομένων περιλάμβανε 42 ακατέργαστα αρχεία(R1/R2). Το μέγεθος του κάθε αρχείου ήταν μεταξύ 220 MB με 5,8 GB. Για κάθε μία από τις διαδικασίες της στοίχισης και της ποσοτικοποίησης χρησιμοποιήθηκαν 12 “threads”.

Φόρτωση αρχείων: 25 λεπτά συνολικά

Διαδικασία φιλτραρίσματος: 17 λεπτά συνολικά

Κατασκευή ευρετηρίου: 45 λεπτά

Διαδικασία στοίχισης: 100 λεπτά συνολικά

Διαδικασία ποσοτικοποίησης: 3 λεπτά συνολικά

Διαδικασία διαφορικής έκφρασης: 3 λεπτά

Διαδικασία ανάλυσης γονιδιακού εμπλουτισμού: 3 λεπτά

Σύνολο: 3 ώρες και 16 λεπτά

Ακόμη έγιναν αναλύσεις δεδομένων σε οργανισμούς όπως το μοντέλο οργανισμός “*Mus musculus*” (ποντίκι) και ο οργανισμός “*Ovis aries*” (πρόβατο) .

5. Συμπεράσματα- Συζήτηση

Κύριος στόχος της εφαρμογής Simplar είναι η διευκόλυνση βιολόγων χωρίς προγραμματιστικό υπόβαθρο, σε αναλύσεις δεδομένων από πειράματα RNA-seq. Οι αναλύσεις των RNA-seq είναι ευρέως χρησιμοποιούμενες στην ερευνητική κοινότητα καθώς επιτρέπει στους ερευνητές να μετρούν ταυτόχρονα τα επίπεδα έκφρασης χιλιάδων γονιδίων σε ένα μόνο πείραμα. Η εφαρμογή Simplar παρέχει άμεσα και γρήγορα μία ολοκληρωμένη άποψη για την γονιδιακή έκφραση δίνοντας πολύτιμες πληροφορίες στους ερευνητές, εντοπίζοντας γονίδια που εκφράζονται διαφορετικά σε διαφορετικές συνθήκες ή ιστούς, βοηθώντας στην αποκάλυψη ρυθμιστικών δικτύων, μοτίβων συν-έκφρασης γονιδίων και βιολογικών διεργασιών που διέπουν τη φυσιολογική ανάπτυξη ή καταστάσεις ασθενειών. Επίσης τα αποτελέσματα από RNA-seq αναλύσεις μπορούν να συνδυαστούν με άλλα “omics” δεδομένα, όπως η γονιδιωματική, η επιγονιδιωματική και η πρωτεομική, ώστε να παρέχεται μια ολοκληρωμένη κατανόηση των βιολογικών διεργασιών. Η ενσωμάτωση των δεδομένων “multi-omics” επιτρέπει στους ερευνητές να διερευνήσουν τις αλληλεπιδράσεις γονιδίων-γονιδίων, να εντοπίσουν βασικούς ρυθμιστικούς παράγοντες και να αποκτήσουν γνώσεις σχετικά με πολύπλοκα μοριακά δίκτυα. Επιπλέον τα δεδομένα RNA-seq αναλύσεων έχουν καταστήσει δυνατή τη μεταγραφωμική ενός κυττάρου (“single-cell transcriptomics”)¹⁶, στην οποία τα προφίλ γονιδιακής έκφρασης μπορούν να ληφθούν με την ανάλυση μεμονωμένων κυττάρων. Αυτή η τεχνολογία παρέχει πληροφορίες σχετικά με την κυτταρική ετερογένεια, τις σχέσεις κυτταρικής σειράς και τον εντοπισμό σπάνιων κυτταρικών τύπων. Η RNA-seq ανάλυση για ένα κύτταρο έχει βοηθήσει στην κατανόηση της ανάπτυξης, της κυτταρικής διαφοροποίησης και των μηχανισμών ασθενειών.

Εκτός αυτού, η εφαρμογή θα μπορούσε χρησιμοποιηθεί και στην κλινική μελέτη, καθώς οι μεθοδολογίες RNA-seq έχουν αρχίσει να εφαρμόζονται σταδιακά στην κλινική έρευνα και διάγνωση.¹⁷

Τα επεξεργασμένα δεδομένα των RNA-seq πειραμάτων επιτρέπουν τη μέτρηση των επιπέδων γονιδιακής έκφρασης σε ολόκληρο το μεταγράψωμα. Συγκρίνοντας τα πρότυπα γονιδιακής έκφρασης μεταξύ διαφορετικών συνθηκών και καταστάσεων, οι ερευνητές μπορούν να εντοπίσουν γονίδια που εκφράζονται διαφορετικά σε συγκεκριμένες ασθένειες ή καταστάσεις. Έτσι παρέχονται πληροφορίες για τους μηχανισμούς των εκάστοτε οργανισμών και θα μπορούσαν να βοηθήσουν στην επίλυση βιολογικών προβλημάτων

καθώς και στην ανάπτυξη χρήσιμων βιοδεικτών. Επί παραδείγματι, θα μπορούσαν να εντοπιστούν πιθανοί βιοδείκτες για τη διάγνωση ή την παρακολούθηση ασθενειών. Συγκρίνοντας τα προφίλ γονιδιακής έκφρασης μεταξύ ασθενών και υγιών, οι ερευνητές μπορούν να εντοπίσουν γονίδια που σχετίζονται ειδικά με την ασθένεια που τους ενδιαφέρει. Αυτοί οι βιοδείκτες μπορούν να χρησιμεύσουν ως διαγνωστικά εργαλεία ή δείκτες εξέλιξης της νόσου, βοηθώντας στην έγκαιρη ανίχνευση και παρακολούθηση της ανταπόκρισης στη θεραπεία.

Μέσω της ανάλυσης εμπλουτισμού γονιδιακών συνόλων μπορεί να γίνει εξαγωγή συμπερασμάτων για βιολογικά μονοπάτια και δίκτυα που απορυθμίζονται σε συγκεκριμένες καταστάσεις. Αναλύοντας δεδομένα γονιδιακής έκφρασης στο πλαίσιο γνωστών βιολογικών δικτύων και γονιδιακών συνόλων, οι ερευνητές μπορούν να αποκτήσουν μια ολοκληρωμένη κατανόηση των μοριακών μηχανισμών που διέπουν τις ασθένειες. Αυτό μπορεί να βοηθήσει στον εντοπισμό των βασικών γονιδιακών συνόλων που επηρεάζονται από την κατάσταση αυτή, και έτσι να γίνουν στοχευμένες παρεμβάσεις.

Επιπλέον τα δεδομένα RNA-seq αναλύσεων μπορούν να χρησιμοποιηθούν για την ταξινόμηση ασθενειών σε διακριτούς υποτύπους με βάση τα προφίλ γονιδιακής έκφρασης. Αυτό μπορεί να βοηθήσει στον εντοπισμό μοριακά διακριτών υποτύπων ασθενειών, οι οποίες μπορεί να έχουν διαφορετικά κλινικά χαρακτηριστικά, προβλέψεις ή θεραπευτικές αποκρίσεις. Αυτή η ομαδοποίηση μπορεί να επιτρέψει προσεγγίσεις εξατομικευμένης ιατρικής, επιτρέποντας στοχευμένες θεραπείες και βελτιωμένα αποτελέσματα για τους ασθενείς. Εκτός αυτού τα δεδομένα RNA-seq αναλύσεων θα μπορούσαν να βοηθήσουν στην πρόβλεψη της ανταπόκρισης σε φάρμακα και στην ανάλυση αποτελεσμάτων θεραπειών. Συγκρίνοντας τα προφίλ γονιδιακής έκφρασης δειγμάτων ασθενών πριν και μετά τη θεραπεία, οι ερευνητές μπορούν να εντοπίσουν γονίδια που σχετίζονται με την ανταπόκριση ή την αντίσταση στη θεραπεία. Αυτές οι πληροφορίες μπορούν να βοηθήσουν στην καθοδήγηση εξατομικευμένων θεραπευτικών αποφάσεων και στον εντοπισμό πιθανών θεραπευτικών συνδυασμών.¹⁸

Αυτά είναι μερικά μόνο παραδείγματα του τρόπου με τον οποίο οι μέθοδοι RNA-seq μπορούν να συμβάλουν στη διάγνωση της κλινικής έρευνας. Η ευελιξία της μεθοδολογίας RNA-seq δίνει τη δυνατότητα στους ερευνητές να διερευνούν και να αναλύουν διάφορες πτυχές της γονιδιακής έκφρασης, επιτρέποντας τη βαθύτερη κατανόηση των ασθενειών και ανοίγοντας το δρόμο για βελτιωμένη διάγνωση και εξατομικευμένη ιατρική.

Συνοπτικά, η ανάλυση RNA-seq έχει μεταμορφώσει την έρευνα γονιδιωματικής παρέχοντας μια ολοκληρωμένη και λεπτομερή εικόνα του μεταγραφώματος και έχει ευρείες εφαρμογές σε διάφορους τομείς, από τη βασική βιολογία έως την κλινική ιατρική. Έχοντας συμβάλει σημαντικά στην κατανόηση της γονιδιακής ρύθμισης, των μηχανισμών των ασθενειών και της ανάπτυξης νέων θεραπευτικών ουσιών.

Για τη βελτιστοποίηση του χρόνου των διεργασιών και των λειτουργιών της εφαρμογής, θα μπορούσε μελλοντικά να εγκατασταθεί σε έναν εξυπηρετητή (server) και όχι να λειτουργεί τοπικά στον προσωπικό υπολογιστή του χρήστη. Έτσι δε θα είναι αναγκαίο να διατίθενται υψηλής απόδοσης επεξεργαστές και μεγάλης χωρητικότητας RAM από τον χρήστη.

6. Παράρτημα Κώδικα

6.1. Κώδικας από το “script” με όνομα 'global2v.r'

Περιέχει όλα τα απαραίτητα πακέτα και τις βιβλιοθήκες για την εφαρμογή.

```
if (!require("tidyverse"))install.packages("tidyverse")
if (!require("shiny"))install.packages("shiny")
if (!require("shinydashboard"))install.packages("shinydashboard")
if (!require("shinycssloaders"))install.packages("shinycssloaders")
if (!require("shinyjs"))install.packages("shinyjs")
if (!require("shinyBS"))install.packages("shinyBS")
if (!require("DT"))install.packages('DT')
if (!require("pheatmap"))install.packages('pheatmap')
if (!require("RColorBrewer"))install.packages('RColorBrewer')
if (!require("ggpubr"))install.packages('ggpubr')
if (!require("stringr"))install.packages('stringr')
if (!require("rmarkdown"))install.packages("rmarkdown")
if (!require("colourpicker"))install.packages("colourpicker")
if (!require("ggfortify"))install.packages("ggfortify")
if (!require("dendsort"))install.packages("dendsort")
if (!require("shinybusy"))install.packages("shinybusy")
if (!require("dashboardthemes"))install.packages("dashboardthemes")
if (!require("BiocManager"))install.packages('BiocManager')

if (!require("Rbowtie2"))BiocManager::install("Rbowtie2")
if (!require("Rsubread"))BiocManager::install("Rsubread")
if (!require("Rsamtools"))BiocManager::install("Rsamtools")
if (!require("clusterProfiler"))BiocManager::install("clusterProfiler")
if (!require("AnnotationHub"))BiocManager::install("AnnotationHub")
if (!require("DOSE"))BiocManager::install("DOSE")
if (!require("enrichplot"))BiocManager::install("enrichplot")
if (!require("ggribes"))install.packages('ggribes')
if (!require("ggnewscale"))install.packages('ggnewscale')
if (!require("genefilter"))BiocManager::install("genefilter")
if (!require("DESeq2"))BiocManager::install("DESeq2")
if (!require("markdown"))install.packages("markdown")
if (!require("genekitr"))install.packages("genekitr")

suppressMessages(library(shinybusy))
suppressMessages(library(genekitr))
suppressMessages(library(dashboardthemes))
suppressMessages(library(markdown))
suppressMessages(library(dendsort))
suppressMessages(library(colourpicker))
suppressMessages(library(stringr))
suppressMessages(library(genefilter))
suppressMessages(library(ggpubr))
suppressMessages(library(pheatmap))
suppressMessages(library(clusterProfiler))
suppressMessages(library(DOSE))
suppressMessages(library(AnnotationHub))
suppressMessages(library(enrichplot))
suppressMessages(library(shiny))
suppressMessages(library(DT))
suppressMessages(library(shinyjs))
suppressMessages(library(shinyBS))
suppressMessages(library(shinycssloaders))
suppressMessages(library(shinydashboard))
suppressMessages(library(ggplot2))
suppressMessages(library(Rsubread))
suppressMessages(library(Rbowtie2))
suppressMessages(library(DESeq2))
```

```
suppressMessages(library(tidyverse))
suppressMessages(library(ggthemes))
suppressMessages(library(ggnewscale))
suppressMessages(library(RColorBrewer))
suppressMessages(library(rmarkdown))
```

6.2. Κώδικας από το “script” με όνομα 'shiny_universal.r'

Περιέχει τον κώδικα του “UI” και του “server”.

```
source(paste(getwd(), 'global2v.r', sep="/"))

sidebar <- dashboardSidebar(
  sidebarMenu(
    menuItem("INFO", icon = icon("dashboard"), tabName = "dashboard"),
    menuItem("Data preprocessing", icon = icon("th"), tabName = "widgets1"),
    menuItem("Differential expression analysis", icon = icon("th"), tabName =
"charts", startExpanded = TRUE),
    menuSubItem('Plots of Differential expression ', tabName='tab2'),
    menuItem("Gene Set enrichment analysis ", icon = icon("th"), tabName =
"widgets2")
  )
)

body <- dashboardBody(
  tabItems(

    tabItem(tabName = "dashboard",
      fluidRow(

        box(title = "User Guide", width = 10, solidHeader = T, status =
"info",
          column(12,
            includeMarkdown("Shiny_App_Manual.RMD")
          )
        )
      )
    ),
    tabItem(tabName = "widgets1",
      ##### Trimming Box #####
      box(title = "Raw Data Preprocessing", solidHeader = T, status =
"primary", width = 12,
        hr(),
        fluidRow(
          add_busy_gif(
            src="https://i.gifer.com/4vsm.gif",
            timeout = 100,
            position = c("full-page"),
            margins = c(10, 10),
            overlay_color = "rgba(0, 0, 0, 0.5)",
            overlay_css = NULL,
            height = "50px",
            width = "50px"
          ),

          tags$head(
            tags$style(
              HTML(".shiny-notification {
```

```

        background-color:#bfb7ee;
    }"
    )
    )
    ),

    column(6,

        box(title = "Upload files ", solidHeader = T, status
= "primary", width = 10, collapsible = T,id = "inputbox",
        h5(strong("First you need to upload the files
that you want to use in the trimming process")),
        fileInput('file1', 'Choose Files',multiple =
TRUE,

            accept=c('.fq','.fa',
                '.fastq' , '.fasta')) ,

            withSpinner( verbatimTextOutput('result')),
            h5(strong("Optional:")),
            actionButton("delete", "Delete all files"),
        )
    ),
    column(6,
        box(title = "Trimming process", solidHeader = T, status =
"primary", width = 10, collapsible = T,id = "trimmingbox",

            selectizeInput("pair1", label = "Choose R1/Forward reads
file",choices = NULL ),

            selectizeInput("pair2", label = "Choose R2/Reverse reads
file",choices = NULL),

            actionButton("initRbowtie","Remove adaptors", class =
"btn-info", style = "width: 100%")

        )
    ),
    )
    ),
    ),
    tabItem(tabName = "charts",

        fluidRow(

            add_busy_gif(
                src="https://i.gifer.com/4vsm.gif",
                timeout = 100,
                position = c( "full-page"),
                margins = c(10, 10),
                overlay_color = "rgba(0, 0, 0, 0.5)",
                overlay_css = NULL,
                height = "50px",
                width = "50px"
            ),

            column(6,
                ##### Mapping Box #####

```

```

        box(title = "Mapping process", solidHeader = T,
status = "primary", width = 12, collapsible = T,id = "mappingbox",

        box(title = "Build index", solidHeader = T,
status = "primary", width = 7, collapsible = T,id = "indexbox",
        h5(strong("Build the genome index if you
haven't already done it"))),

        selectizeInput("reference_fasta",
label = "Choose reference genome:",choices = NULL ),

        actionButton("initBuild","Build index", class
= "btn-info", style = "width: 100%")),

        box(title = "Alignment", solidHeader = T, status
= "primary", width = 10, collapsible = T,id = "alignmentbox",
        wellPanel(

            column(4,
                numericInput("minFrag","minFragLength:",
value = 50)
            ),

            column(4,
                numericInput("maxFrag","maxFragLength:",
value = 600)
            ),

            column(4,
                numericInput("misMatch","maxMismatches:",
value = 3)
            ),

            column(4,
                selectInput("format","OutputFormat:",
choices = c("SAM", "BAM"), selected = "BAM")
            ),

            column(5,
                numericInput("nthreads","nthreads:", value =
12, max = 40)
            ),

            #h5(strong("Proceed to alignment")),

            actionButton("initAlign","Alignment", class =
"btn-info", style = "width: 100%")

        ))
    ),

##### Quantification Box #####

        column(6,
            box(title = "Quantification process", solidHeader = T, status
= "primary", width = 12, collapsible = T,id = "quantifybox",

            selectizeInput("reference_gtf", label = "Choose reference
genome:",choices = NULL ),

            wellPanel(
                #column(4,

                #selectInput("isGTFAnnotationFile","isGTFAnnotationFile:", choices = c(TRUE,
FALSE), selected = TRUE)
                # ),

```

```

        column(4,
            selectInput("overlap", "AllowMultiOverlap:", choices
= c(TRUE, FALSE), selected = FALSE)
        ),

        column(4,
            numericInput("minOverlap", "minOverlap:", value = 1)
        ),
        column(4,

selectInput("countMultiMappingReads", "countMultiMappingReads:", choices = c(TRUE,
FALSE), selected = TRUE)
        ),

        column(4,
            numericInput("minFragL", "minFragLength:", value =
50)
        ),

        column(4,
            numericInput("maxFragL", "maxFragLength:", value =
600)
        ),
        column(4,
            numericInput("nthreads", "nthreads:", value = 12, max
= 40)
        ),

        hr(),

        actionButton("initQuantify", "Quantification", class =
"btn-info", style = "width: 100%"),

        downloadButton('downloadcountCSV', 'Save Results as CSV
File', class = "btn btn-info", style="margin: 7px;")

    ))

),

column(10,
    tags$div(class = "BoxArea2",
        withSpinner(dataTableOutput('Table'))
    ),
    tags$div(class = "clearBoth")
),
tags$div(class = "clearBoth"),

##### Differential Expression Analysis Box #####
box(title = "Differential Expression Analysis", solidHeader = T,
status = "success", width = 12,
    hr(),
    column(6,
        box(title = "Upload the counts file", solidHeader = T, status
= "success", width = 10, collapsible = T, id = "uploadcsvbox",
            h5(strong("Make sure that the header of the csv contains
only the sample names"))),

        fileInput('target_upload_csv', 'Choose file to upload',
            accept = c(
                'text/csv',
                'text/comma-separated-values',
                '.csv'
            ))),

```

```

    ),
    DT::dataTableOutput("count_table")),

    column(10,
      box(title = "Upload the phenodata file", solidHeader = T,
        status = "success", width = 8, collapsible = T, id = "uploadcsvbox2",
        h5(strong("Make sure that the file contains the same
        samples as the counts file and that the names match")),
        fileInput('metadataFile', '',
          accept=c('text/csv',
            'text/comma-separated-
values,text/plain',
            '.csv'),multiple = FALSE
        ),
      ),
    ),

    column(10,
      box(title = "DESeq2 Object Settings ", solidHeader
= T, status = "success", width = 10, collapsible = T, id = "DDSBox",

      h5(strong("Design your own formula depending on
your analysis needs (differential expression with multiple or single factors)")),
      textInput("designFormula", "Design Formula:",
placeholder = "~ Conditions"),

      h5(strong(" You can specify the comparison of
interest, and the levels to compare.
The level given last is the base level
for the comparison.")),
      selectInput("factorNameInput", label ="Choose
the contrast column:", choices = NULL),
      actionBar("choice", "Update Levels by
your contrast choice"),
      selectInput("condition1", label ="Choose Level
1:", choices = NULL),
      selectInput("condition2", label ="Choose Level
2:", choices = NULL),
      column(4,
        numericInput("padj", "P-Adjust Cutoff:",
value = 0.05)
      ),
      column(4,
        numericInput("log2F", "log2FoldChange
Cutoff:", value = 1)
      ),
      actionBar("initResults", "Get Results",
class = "btn-info", style = "width: 100%"),
      downloadButton('downloadDDSCSV2', 'Save
Results as CSV File', class = "btn btn-info", style="margin: 7px;"),

    ),

    DT::dataTableOutput("Table2"),

  )
)

```

```

    )),

##### Differential Expression Plots Box #####
    tabItem(tabName = "tab2",

      fluidRow(

        add_busy_gif(
          src="https://i.gifer.com/4vsm.gif",
          timeout = 100,
          position = c( "full-page"),
          margins = c(10, 10),
          overlay_color = "rgba(0, 0, 0, 0.5)",
          overlay_css = NULL,
          height = "50px",
          width = "50px"
        ),

        box(title = "Color palette", solidHeader = T, status = "success",
width = 8, collapsible = T,id = "palette",

          column(4,
            colourInput("col1", "Select color 1", "#30ABAD", allowTransparent =
T)),
          column(4,
            colourInput("col2", "Select color 2", "#BF58DB", allowTransparent =
T)),
          column(4,
            colourInput("col3", "Select color 3", "#555C5C", allowTransparent =
T))),

        box(title = "Volcano plot", solidHeader = T, status = "success",
width = 10, collapsible = T,id = "volcanoplot2",

          column(9,
            wellPanel(
              withSpinner(plotOutput(outputId = "volcanoplot2"))
            ),
            downloadLink("downloadPlotvol", "Download Plot")
          ),

        box(title = "Ma plot", solidHeader = T, status = "success", width =
10, collapsible = T,id = "maplot2",

          column(9,
            wellPanel(
              withSpinner(plotOutput(outputId = "maplot2"))
            ),
            downloadLink("downloadPlotma", "Download Plot")
          ),

        box(title = "PCA plot", solidHeader = T, status = "success", width
= 10, collapsible = T,id = "pcaplot",
          fluidRow(
            column(6,
              h4(strong("Plot Settings:")),
              wellPanel(

                fluidRow(
                  column(12,
                    selectInput("group","Group of interest:",
choices = c())

                    ))),

            box(title = "PCA ", solidHeader = T, status = "success", width
= 10, collapsible = T,id = "pcaplot2",
              column(10,

```

```

        wellPanel(
          withSpinner(plotOutput(outputId = "pcaplot"))
        ),

      ),

      downloadLink("downloadPlot7", "Download Plot"))
    )),

    box(title = "Heatmap ", solidHeader = T, status = "success", width
= 10, collapsible = T, id = "heatmap",
      fluidRow(
        column(6,
          h4(strong("Plot Settings:")),
          wellPanel(

            fluidRow(

              column(12,
                selectInput("group1", "Group 1:", choices =
c()

              ),
              column(12,
                selectInput("group2", "Group 2 (Optional):",
choices = c()

              ),
              column(12,
                radioButtons("n", "Number of groups:", c("1",
"2")))

            ),

            div(style = "clear:both;")
          )
        ),

        box(title = "Heatmap", solidHeader = T, status = "success",
width = 11, collapsible = T, id = "heatmap",

          column(10,
            wellPanel(
              withSpinner(plotOutput(outputId = "heatmap"))
            )
          ),
          downloadLink("downloadPlot6", "Download Plot")
        )
      ),

      box(title = "Gene Expression Boxplot", solidHeader = T, status =
"success", width = 12,

        fluidRow(
          column(6,
            wellPanel(
              column(12,
                selectizeInput("sel_gene",
label="Gene Name/Id (Select 1
or more)",

                choices = NULL,
                multiple=TRUE,
                options = list(
                  placeholder =
name/id'

                ) #,

              )
            ),

          )
        )
      )

```

```

        div(style = "clear:both;")
      ),
      column(6,
        h4(strong("Plot Settings:")),
        wellPanel(
          fluidRow(
            column(12,
              selectInput("boxplotX", "X-axis", choices =
c()),
            ),
            column(12,
              selectInput("boxplotFill", "Grouped by",
choices = c())
            )
          ),
          div(style = "clear:both;")
        )
      ),
      box(title = "Boxplot", solidHeader = T, status = "success",
width = 11, collapsible = T, id = "boxPlot",
        column(10,
          wellPanel(
            withSpinner(plotOutput(outputId = "boxPlot"))
          ),
          downloadLink("downloadPlot8", "Download Plot")
        )),
      box(title = "Count Boxplot", solidHeader = T, status =
"success", width = 11, collapsible = T, id = "boxplot2",
        column(6,
          h4(strong("Choose color palette:")),
          wellPanel(
            fluidRow(
              column(12,
                selectInput("Palette", "Palette:",
c("BuGn", "Blues", "OrRd", "Spectral", "Purples", "YlGnBu", "YlOrBr"), selected =
"BuGn")
              )
            )
          ),
          column(10,
            wellPanel(
              withSpinner(plotOutput(outputId = "boxplot2"))
            ),
          ),
        ),
      ),
      tabItem(tabName = "widgets2",
        fluidRow(
          add_busy_gif(

```

```

        src="https://i.gifer.com/4vsm.gif",
        timeout = 100,
        position = c( "full-page"),
        margins = c(10, 10),
        overlay_color = "rgba(0, 0, 0, 0.5)",
        overlay_css = NULL,
        height = "50px",
        width = "50px"
    ),

    #box(title = "Gene Enrichment Analysis", solidHeader = T, status =
"success", width = 12,
    # hr(),
    column(6,
        box(title = "Upload Data", solidHeader = T, status =
"success", width = 12, collapsible = T,id = "uploadbox",

            h5(strong("Before you upload your dataset make
sure that the column that contains the entrezids
is named gene_name and that the column that contains the log2
Fold Change results is named log2FoldChange.")),

            fileInput('target_upload3', 'Choose file to
upload',

                accept = c(
                    'text/csv',
                    'text/comma-separated-values',
                    '.csv'
                )),

            h5(strong("In case there is an pattern in front
of the gene ids in the gene_name column, you should remove it.")),

            column(4,
                textInput("pattern",label="Give the
pattern: ")),

            column(10,
                checkboxInput("removePattern","Remove pattern",
value = F))
        ),

        DT::dataTableOutput("sample_table")),

##### Annotation Box #####

        box(title = "Annotation Parameters", solidHeader = T, status =
"primary", width = 8, collapsible = T,id = "Annotationhub",

            fluidRow(
                column(4,
                    textInput("organism",label="Select organism (latin
name):")),

                column(10,
                    actionButton("search","Search in Annotationhub", class = "btn-
info")),

                column(10,
                    withSpinner(verbatimTextOutput('text'))),
                column(4,
                    textInput("annot",label="Select annotation file code:")),
                column(10,
                    actionButton("annotation","Download file from Annotationhub",
class = "btn-info")),
            ),

```

```
##### Enrichment Box #####
box(title = "Gene Enrichment Object Parameters", solidHeader = T,
status = "primary", width = 8, collapsible = T, id = "createGoBox",

  wellPanel(
    # column(4,
      # selectizeInput("keytypes", "Keytypes:", choices =
c("MF", "BP", "CC", "ALL"), selected = "ALL")
      # ),
      column(4,

        selectizeInput("org",
          label="Select organism (short name):",
          choices = NULL,

          options = list(
            placeholder =
              'Type to search for an organism
ex:human'
          ) ) ,

        column(4,
          selectInput("ontology", "Ontology:", choices = c("MF",
"BP", "CC", "ALL"), selected = "ALL")
        ),
        column(4,
          numericInput("nPerm", "Permutation :", value = 1000,
min = 1, max = 100000)
        ),
        column(4,
          numericInput("minGSSize", "minGSSize:", value = 3)
        ),
        column(4,
          numericInput("maxGSSize", "maxGSSize:", value = 800)
        ),
        column(4,
          numericInput("pvalCutoff", "P-Value Cutoff:", value =
0.05)
        ),
        column(4,
          selectInput("pAdjustMethod", "pAdjustMethod:", choices
= c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"),
selected = "none")
        ),

        tags$div(class = "clearBoth")
      ),
      actionButton("initGo", "Create gseGO Object", class = "btn-
info", style = "width: 100%"),

      downloadButton('downloadgseGoCSV', 'Save Results as CSV File',
class = "btn btn-info", style="margin: 7px;"),

      fluidRow(

        column(10,
          tags$div(class = "BoxArea2",
            withSpinner(dataTableOutput('gseGoTable'))
          ),
          tags$div(class = "clearBoth")
        ),
        tags$div(class = "clearBoth")
      ),
    ),
  ),
),
```

```
##### Enrichment Plots Box #####

box(title = "Dot Plot", solidHeader = T, status = "success",
width = 10, collapsible = T, id = "dotPlot",
  fluidRow(
    column(3,
      wellPanel(
        numericInput("showCategory_dot", "number of
categories to show", value = 10)
      ),
    column(6,
      wellPanel(
        withSpinner(plotOutput(outputId = "dotPlot"), type
= 8)
      ),
    ),
    downloadLink("downloadPlot", "Download Plot")
  ),
  box(title = "Category Netplot", solidHeader = T, status =
"success", width = 10, collapsible = T, id = "cnetplot",
  fluidRow(
    column(9,
      wellPanel(
        numericInput("showCategory_cne", "number of
categories to show", value = 5)
      )
    ,
    column(9,
      wellPanel(
        plotOutput(outputId = "cnetplot")
      )
    ),
    downloadLink("downloadPlot4", "Download Plot")
  ),
  box(title = "Heat Plot", solidHeader = T, status = "success",
width = 10, collapsible = T, id = "heatplot",
  column(9,
    wellPanel(
      withSpinner(plotOutput(outputId = "heatplot"))
    )
  ),
  downloadLink("downloadPlot5", "Download Plot")
),
  box(title = "Ridge Plot", solidHeader = T, status = "success",
width = 10, collapsible = T, id = "ridgeplot",
  fluidRow(
    column(3,
      wellPanel(
        numericInput("showCategory_ridge", "number of
categories to show", value = 10)
      )
    ),
    column(6,
      wellPanel(
        withSpinner(plotOutput(outputId =
"ridgePlot"), type = 8)
      )
    )
  )
)
```

```

    ),
    downloadLink("downloadPlot2", "Download Plot")
  ),
  box(title = "GSEA Plot", solidHeader = T, status = "success",
width = 10, collapsible = T, id = "gseaplot",
  fluidRow(
    column(9,
      wellPanel(
        numericInput("geneSetId_gsea", "Gene Set ID",
value = 1)
      ),
    column(9,
      wellPanel(
        plotOutput(outputId = "gseaplot", width = "100%",
height = "400px")
      ),
    ),
    downloadLink("downloadPlot3", "Download Plot")
  ),
),
)
)

```

```
),
```

```
customTheme <- shinyDashboardThemeDIY(
```

```

appFontFamily = "Arial"
,appFontColor = "rgb(0,0,0)"
,primaryFontColor = "rgb(0,0,0)"
,infoFontColor = "rgb(0,0,0)"
,successFontColor = "rgb(0,0,0)"
,warningFontColor = "rgb(0,0,0)"
,dangerFontColor = "rgb(0,0,0)"
,bodyBackColor = "rgb(174 ,238, 238)"

# header
,logoBackColor = "rgb(20,97,117)"

,headerButtonBackColor = "rgb(40, 197 ,205)"
,headerButtonIconColor = "rgb(75,75,75)"
,headerButtonBackColorHover = "rgb(210,210,210)"
,headerButtonIconColorHover = "rgb(40, 197 ,205)"

,headerBackColor = "rgb(40, 197 ,205)"
,headerBoxShadowColor = "#515A5A"
,headerBoxShadowSize = "2px 2px 2px"

# sidebar
,sidebarBackColor = cssGradientThreeColors(
  direction = "down"
  ,colorStart = "rgb(20,97,117)"
  ,colorMiddle = "rgb(56,161,187)"
  ,colorEnd = "rgb(3,22,56)"
  ,colorStartPos = 0
  ,colorMiddlePos = 50
  ,colorEndPos = 100
)
,sidebarPadding = 0

,sidebarMenuBackColor = "transparent"
,sidebarMenuPadding = 0
,sidebarMenuBorderRadius = 0

,sidebarShadowRadius = "3px 8px 9px"

```

```

,sidebarShadowColor = "#515A5A"

,sidebarUserTextColor = "rgb(255,255,255)"
,sidebarSearchBackColor = "rgb(23,103,124)"
,sidebarSearchIconColor = "rgb(255,255,255)"
,sidebarSearchBorderColor = "rgb(23,103,124)"

,sidebarTabTextColor = "rgb(255,255,255)"
,sidebarTabTextSize = 13
,sidebarTabBorderStyle = "none none solid none"
,sidebarTabBorderColor = "rgb(35,106,135)"
,sidebarTabBorderWidth = 1

,sidebarTabBackColorSelected = cssGradientThreeColors(
  direction = "right"
  ,colorStart = "rgba(0,255,213,1)"
  ,colorMiddle = "rgba(0,255,213,1)"
  ,colorEnd = "rgba(0,255,213,1)"
  ,colorStartPos = 0
  ,colorMiddlePos = 30
  ,colorEndPos = 100
)
,sidebarTabTextColorSelected = "rgb(0,0,0)"
,sidebarTabRadiusSelected = "0px 20px 20px 0px"

,sidebarTabBackColorHover = cssGradientThreeColors(
  direction = "right"
  ,colorStart = "rgba(0,255,213,1)"
  ,colorMiddle = "rgba(0,255,213,1)"
  ,colorEnd = "rgba(0,255,213,1)"
  ,colorStartPos = 0
  ,colorMiddlePos = 30
  ,colorEndPos = 100
)
,sidebarTabTextColorHover = "rgb(50,50,50)"
,sidebarTabBorderStyleHover = "none none solid none"
,sidebarTabBorderColorHover = "rgb(75,126,151)"
,sidebarTabBorderWidthHover = 1
,sidebarTabRadiusHover = "0px 20px 20px 0px"

# boxes
,boxBackColor = "rgb(255,255,255)"
,boxBorderRadius = 5
,boxShadowSize = "3px 5px 5px"
,boxShadowColor = "#aaaaaa"
,boxTitleSize = 16
,boxDefaultColor = "rgb(210,214,220)"
,boxPrimaryColor = "rgba(40, 197 ,205)"
,boxInfoColor = "rgb(40, 197 ,205)"
,boxSuccessColor = "rgba(40, 197 ,205)"
,boxWarningColor = "rgb(244,156,104)"
,boxDangerColor = "rgb(255,88,55)"

,tabBoxTabColor = "rgb(255,255,255)"
,tabBoxTabTextSize = 14
,tabBoxTabTextColor = "rgb(0,0,0)"
,tabBoxTabTextColorSelected = "rgb(0,0,0)"
,tabBoxBackColor = "rgb(255,255,255)"
,tabBoxHighlightColor = "rgba(0,255,213,1)"
,tabBoxBorderRadius = 5

# inputs
,buttonBackColor = "rgb(0,255,213,1)"
,buttonTextColor = "rgb(0,0,0)"
,buttonBorderColor = "rgb(200,200,200)"
,buttonBorderRadius = 5

,buttonBackColorHover = "rgb(56,161,187)"
,buttonTextColorHover = "rgb(0,255,213,1)"
,buttonBorderColorHover = "rgb(56,161,187)"

,textboxBackColor = "rgb(255,255,255)"

```

```

,textboxBorderColor = "rgb(200,200,200)"
,textboxBorderRadius = 5
,textboxBackColorSelect = "rgb(245,245,245)"
,textboxBorderColorSelect = "rgb(200,200,200)"

# tables
,tableBackColor = "rgb(255,255,255)"
,tableBorderColor = "rgb(240,240,240)"
,tableBorderTopSize = 1
,tableBorderRowSize = 1

)

)

ui <- dashboardPage(
  dashboardHeader(title
=tags$a(tags$img(src='logo5.png',height='50px',width='200px')),
  sidebar,
  body
)

#max file size is 12 GB (for the uploaded files)
options(shiny.maxRequestSize = 12000*1024^2)

server <- function(input, output, session) {

  #path to Rbowtie2/extdata/adrm
  destDir <- system.file("extdata", "adrm",package = "Rbowtie2")

  #copy user's raw .fastq files inside the subfolder adrm
  output$result <- renderPrint({

    inFile <- input$file1
    if (is.null(inFile)) {
      cat("NOT FILE\n")
      #Update the files inside adrm and render the names in the textoutput
      result <- list.files(system.file("extdata", "adrm", package="Rbowtie2"),
                           pattern="*.fastq$")
      updateForm()
      return(result)
    }
    cat("Reading file:", inFile$name, "\n")
    cat("size:", inFile$size, " Bytes, type:", inFile$type, "\n")
    if (dir.exists(destDir)){
      cat("Copying file to:", destDir,"\n")
      result <- file.copy( inFile$datapath,
                           file.path(destDir, inFile$name) )
      result<- list.files(system.file("extdata", "adrm", package="Rbowtie2"),
                           pattern="*.fastq$")

    } else {
      result <- FALSE
    }

    updateForm()
    result
  })

  observe({

    delete_files()

```

```

    })

    #delete all the files from extdata\adrm after you are done with the trimming
    process
    delete_files <- eventReactive(input$delete,{

    files_to_delete <- dir(path=destDir ,pattern=".fastq")
    file.remove(file.path(destDir, files_to_delete))
    showNotification("Files are deleted!", duration = 5,
                    type = ( "message"))
    })

    #update the file lists (forward and reverse) in the trimming box in ui
    updateForm = function()
    {
        isolate({

            groupv = list.files(system.file(package =
            "Rbowtie2", "extdata", "adrm"), pattern = "R1.*.fastq$" )
            groupv2 = list.files(system.file(package =
            "Rbowtie2", "extdata", "adrm"), pattern = "R2.*.fastq$" )

            updateSelectizeInput(session, "pair1", choices = groupv, selected =
            groupv[1], server=TRUE)
            updateSelectizeInput(session, "pair2", choices = groupv2, selected =
            groupv2[1], server=TRUE)

        })
    }

    ##### Trimming #####
    observe({

        Rbowtie2Reactive()

    })

    td<-getwd()

    Rbowtie2Reactive <- eventReactive(input$initRbowtie,{
        withProgress(message = "Trimming , please wait",{

            pair1<-input$pair1
            pair2<-input$pair2
            reads_1 <- system.file(package = "Rbowtie2", "extdata", "adrm" , pair1)
            reads_2 <-system.file(package = "Rbowtie2", "extdata", "adrm" , pair2)

            (cmdout<-remove_adapters(file1=reads_1,
                                    file2=reads_2,
                                    adapter1 = "TACACTCTTTCCCTACACGACGCTCTTCCGATCT",
                                    adapter2 = "GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT" ,
                                    basename=file.path(td, "trimmed"),

                                    overwrite=TRUE, "--threads 3"))

            #path to "trimmed.pair1.truncated" file
            file1<-file.path(td, "trimmed.pair1.truncated")
            #rename pair1 from "trimmed.pair1.truncated" to
            "original_file_name_trimmed.fastq"
            file.rename(file1, (sub("*.fastq$", "_trimmed.fastq", pair1)))
            #path to "trimmed.pair2.truncated" file
            file2<-file.path(td, "trimmed.pair2.truncated")

```

```

        #rename pair2 from "trimmed.pair2.truncated" to
        "original_file_name_trimmed.fastq"
        file.rename(file2, (sub("*.fastq$", "_trimmed.fastq", pair2)))
        #path to "trimmed.settings" file
        file3<-file.path(td, "trimmed.settings")
        file.rename(file3, (sub("*.fastq$", "_trimmed.settings", pair1)))

    })
    showNotification("Trimming completed!", duration = NULL,
                    type = ( "message"))
})

##### Build index #####
observe({

    BuildindexReactive()

})
#list reference genome file (.fastq/.fa/.fna) file from the working directory
REF_GENOME <- list.files(td, pattern = "*.fasta$|*.fa$|*.fna$" )
updateSelectizeInput(session, "reference_fasta", choices = REF_GENOME, selected
= REF_GENOME[1],server=TRUE)
BuildindexReactive <- eventReactive(input$initBuild,{
    withProgress(message = "Building index , please wait",{

        #define the basename for the index
        RSUBREAD_INDEX_BASE <- "genome"

        setProgress(value = 0.3, detail = "This might take a while, please wait
        ...")
        #build gapped and splitted index to save storage
        buildindex(basename=file.path(td,RSUBREAD_INDEX_BASE),
                    reference=input$reference_fasta,
                    gappedIndex=TRUE,
                    indexSplit=TRUE,
                    memory=4000)

    })
    showNotification("Completed!", duration = NULL,
                    type = ( "message"))
})

##### Mapping #####
observe({

    MappingReactive()

})

MappingReactive <- eventReactive(input$initAlign,{

    withProgress(message = "Mapping , please wait",{

```

```

# define the basename for the index
RSUBREAD_INDEX_BASE <- "genome"

#list trimmed forward reads
inputfilefwd <- list.files(td, pattern = "_R1_*.fastq$")
#list trimmed reverse reads
inputfilervs <- list.files(td, pattern = "_R2_*.fastq$")

setProgress(value = 0.3, detail = "This might take a while, please wait
...")

align(index=file.path(td,RSUBREAD_INDEX_BASE),
      minFragLength = input$minFrag,
      maxFragLength = input$maxFrag,
      maxMismatches=input$misMatch,
      nthreads = input$nthreads,
      readfile1=inputfilefwd,
      readfile2=inputfilervs,
      output_format=input$format)

})

showNotification("Mapping completed!", duration = NULL,
                 type = ( "message"))

})

##### Quantification #####

#list reference genome files (.gtf) from working directory
gtf <- list.files(td, pattern = "*.gtf$")

updateSelectizeInput(session, "reference_gtf", choices = gtf, selected =
gtf[1],server=TRUE)

QuantifyReactive <- eventReactive(input$initQuantify,{

  withProgress(message = "Counting , please wait",{

    #list the .SAM or .BAM files from the working directory
    inputfile <- list.files(td, pattern = "*.SAM$|*.BAM$")

    mycounts<-featureCounts(inputfile,
                           annot.ext= input$reference_gtf,
                           isGTFAnnotationFile=TRUE,
                           isPairedEnd=TRUE,
                           allowMultiOverlap =input$overlap,
                           minOverlap = input$minOverlap,
                           countMultiMappingReads =
input$countMultiMappingReads,
                           minFragLength = input$minFragL,
                           maxFragLength = input$maxFragL,
                           nthreads=input$nthreads)

    mycounts$counts
    return(list('counts'=mycounts$counts))

  })

})

```

```

#read the counts file
df_products <- reactive({
  inFile4 <- input$target_upload_csv
  if (is.null(inFile4))
    return(NULL)
  dataframe <- read.csv(inFile4$datapath, header = TRUE, sep = ',')

  return(dataframe)
})

output$count_table<- DT::renderDataTable({
  d <- df_products()
  DT::datatable(d)
})

#read the phenodata file
observe({
  metadataFileReactive()
})
metadataFileReactive <- reactive({

  #check if a file is selected, if not then ask to upload a file
  shiny:: validate(
    need( (!is.null(input$metadatafile)),
      message = "Please select a file")
  )

  inFile <- input$metadatafile
  if (is.null(inFile))
    return(NULL)

  inFile = inFile$datapath

  sep = '\t'
  if(length(inFile) > 0 ){
    testSep = read.csv(inFile[1], header = TRUE, sep = '\t')
    if(ncol(testSep) < 2)
      sep = ','
  }
  else
    return(NULL)

  fileContent = read.csv(inFile[1], header = TRUE, sep = sep)

  sampleN = colnames(fileContent)[-1]
  metaData <- fileContent[,sampleN]
  metaData <- data.frame(sapply( metaData, as.factor ))

  row.names(metaData) <- fileContent[,1]
  if(length(colnames) == 1)
    colnames(metaData)[1] <- sampleN
  myValues$DF = metaData

  updateDesignFormula()
  updateFormula()

  return(metaData)
})

#Update the design formula in DESeq2 depending on the phenodata file
updateDesignFormula = function()
{
  isolate({
    groupvars = colnames(myValues$DF)
    #for 1 condition
    if(length(groupvars) == 1)
    {

```

```

    designFormula = paste("~ ",groupvars)

  }
  #for more conditions
  else
    designFormula = paste(" ~ ",paste(groupvars, collapse=" + "))

  updateTextInput(session,"designFormula", value = designFormula)

})

}

#Update contrast choice
updateFormula = function()
{
  isolate({

    groupvars = colnames(myValues$DF)

    updateSelectizeInput(session, "factorNameInput", choices = groupvars,
selected = groupvars[1])

  })
}

observe({

  Select_name()

})

#Update levels depending on the contrast choice
Select_name <- eventReactive(input$choice,{

  isolate({

    groupvars2 = myValues$DF[, (input$factorNameInput)]

    updateSelectizeInput(session, "condition1", choices = groupvars2, selected
= groupvars2[1])
    updateSelectizeInput(session, "condition2", choices = groupvars2, selected
= groupvars2[2])

  })
})

##### Differential expression #####
observe({

  ResultsReactive()

})

ResultsReactive <- eventReactive(input$initResults,{
  withProgress(message = "Processing , please wait",{
    df2<-df_products()
    #remove the NA values from the counts dataframe
    df2<-na.omit(df2)

    #convert the counts dataframe to a data matrix
    countDataMatrix <- as.matrix(df2[ , -1])
    rownames(countDataMatrix) <- df2[ , 1]

    samples <- myValues$DF

```

```

dds1 <- DESeqDataSetFromMatrix(countData=countDataMatrix,
                               colData=samples,
                               design=as.formula(input$designFormula))

dds <- DESeq(dds1)
myValues$dds<-dds

contrast1<-input$condition1
contrast2<-input$condition2

myValues$countDataMatrix <- countDataMatrix
myValues$sample_names<- colnames(countDataMatrix)

#contrast column
col<-input$factorNameInput
#contrast column and levels
contr <- c(col, contrast1,contrast2)

res1<- results(dds, contrast=contr, alpha= 0.05)
myValues$res1<-res1

#padj input
padj.cutoff <- input$padj
#lfc input
lfc.cutoff <- input$log2F

#convert the dds results to a dataframe and add the header name "gene_name"
#to the gene ids column
results <- res1 %>%
  data.frame() %>%
  rownames_to_column(var="gene_name") %>%
  as_tibble()

myValues$names<-rownames(res1)

differ_expr <- results %>%
  filter(padj < padj.cutoff & abs(log2FoldChange) > lfc.cutoff)

vsd <- varianceStabilizingTransformation(dds)
myValues$svstMat <- assay(vsd)
myValues$vsd <- vsd
myValues$vsdColNames <- colnames(vsd)

return(list('differ_expr'=differ_expr))
})
})

#Read the DEGs file
df_products_upload3 <- reactive({
  inFile3 <- input$target_upload3

  if (is.null(inFile3))
    return(NULL)
  dataframe <- read.csv(inFile3$datapath, header = TRUE, sep = ',')
  #remove LOC pattern if you locate it
  if(sum(str_detect(dataframe$gene_name, 'LOC')) > 0){
    dataframe$gene_name<-sub("LOC","",as.character(dataframe$gene_name))
  }

```

```

#Extra pattern removal (OPTIONAL)
if(input$removePattern)
{
  string<-input$pattern

  if(sum(str_detect(dataframe$gene_name, string)) > 0){
    dataframe$gene_name<-sub(string,"",as.character(dataframe$gene_name))}

  return(dataframe)
})

output$sample_table<- DT::renderDataTable({
  df <- df_products_upload3()
  DT::datatable(df)
})

output$text <- renderPrint({

  if (!is.null(AnnotationReactive())) {
    return(myValues$annotation)
  }
})

##### Annotation #####
observe({

  AnnotationReactive()

})

#search for the organism input in AnnotationHub data base
AnnotationReactive <- eventReactive(input$search,{
  withProgress(message = "Searching , please wait",{
    organism <- input$organism

    hub <- AnnotationHub()
    annotation<-query(hub,c("OrgDb", organism))
    myValues$annotation<-annotation
    annotation

  })
})

#download the file that the user chose
observe({
  keytypesReactive()
})
keytypesReactive <- eventReactive(input$annotation,{
  withProgress(message = "Downloading , please wait",{
    # download the chosen file
    hub <- AnnotationHub()

```

```

    annot <- hub[[input$annot]]
    #update organism names for the transformation function
    names<- genekitr::ensOrg_name
    latin_short_name<-names$latin_short_name
    updateSelectizeInput(session, "org", choices = latin_short_name, selected =
latin_short_name[1],server=TRUE)

    myValues$annot<-annot
    annot

  })

  showNotification("Downloaded!", duration = 10,
                    type = ( "message"))
  })

##### Gene Set enrichment #####
myValues = reactiveValues()

observe({

  gseGoReactive()

})

gseGoReactive <- eventReactive(input$initGo,{

  withProgress(message = "Processing , please wait",{

    isolate({

      validate(need(tryCatch({

        annotation <- myValues$annot
        df1<- df_products_upload3()

        #transform gene_names to ENTREZIDs
        f<-transId(df1$gene_name,org=input$org, transTo = "entrez")
        gene_name<-f$entrezid

        original_gene_list <- df1$log2FoldChange

        # name the vector
        names(original_gene_list) <- gene_name

        # omit any NA values
        gene_list<-na.omit(original_gene_list)

        # sort the list in decreasing order (required for clusterProfiler)
        gene_list = sort(gene_list, decreasing = TRUE)

        myValues$gene_list<-gene_list

        setProgress(value = 0.3, detail = "Performing GSE analysis, please wait
...")

        go_gse <- gseGO(geneList=gene_list,
                        ont = input$ontology,

```

```

        keyType = "ENTREZID",
        nPerm = input$nPerm,
        minGSSize = input$minGSSize,
        maxGSSize = input$maxGSSize,
        pvalueCutoff = input$pvalCutoff,
        verbose = T,
        OrgDb = annotation,
        pAdjustMethod = input$pAdjustMethod)

    if(nrow(go_gse) < 1)
    {
        showNotification(id="warnNotify", "No gene can be mapped ...", type =
"warning", duration = NULL)
        showNotification(id="warnNotify2", "Tune the parameters and try
again.", type = "warning", duration = NULL)
        return(NULL)
    }

    updateNumericInput(session, "showCategory_dot", max =
nrow(go_gse@result) , min = 0, value = ifelse(nrow(go_gse@result) > 0, 5,0))
    updateNumericInput(session, "showCategory_ridgeplot", max =
nrow(go_gse@result) , min = 0, value = ifelse(nrow(go_gse@result) > 0, 5,0))
    updateNumericInput(session, "showCategory_cnet", max =
nrow(go_gse@result) , min = 0, value = ifelse(nrow(go_gse@result) > 0, 5,0))
    updateNumericInput(session, "showCategory_gsea", max =
nrow(go_gse@result) , min = 0, value = ifelse(nrow(go_gse@result) > 0, 5,0))
    updateNumericInput(session, "showCategory_heatplot", max =
nrow(go_gse@result) , min = 0, value = ifelse(nrow(go_gse@result) > 0, 5,0))

    }, error = function(e) {
        myValues$status = paste("Error: ",e$message)

        showNotification(id="errorNotify", myValues$status, type = "error",
duration = NULL)
        showNotification(id="errorNotify1", "Make sure that the gene_name
column has the right format!", type = "error", duration = NULL)
        showNotification(id="errorNotify2", "Make sure that the needed columns
have the appropriate names! ", type = "error", duration = NULL)
        return(NULL)
    }

    ),
    "Error merging files"))

})

return(list('go_gse'=go_gse))

})
})

#Render the counts from the quantification process, as a data table
output$Table <- renderDataTable({
    featureCounts <- QuantifyReactive()

    if(!is.null(featureCounts)){
        df<-featureCounts$counts
        for ( col in 1:ncol(df)){
            colnames(df)[col] <- sub("_trimmed", "", colnames(df)[col])
        }

        for ( col in 1:ncol(df)){
            colnames(df)[col] <- sub("*.fastq.subread.SAM$", "", colnames(df)[col])
        }

        for ( col in 1:ncol(df)){
            colnames(df)[col] <- sub("*.fastq.subread.BAM$", "", colnames(df)[col])
        }
    }

```

```

    for ( col in 1:ncol(df)){
      colnames(df)[col] <- sub("-", "_", colnames(df)[col])
    }

    for ( col in 1:ncol(df)){
      colnames(df)[col] <- sub("R1", "", colnames(df)[col])
    }

    resultDF = df
  }

})

#Download the counts data table in .csv format
output$downloadcountCSV <- downloadHandler(
  filename = function() {paste0("counts",".csv")},
  content = function(file) {
    #Remove _trimmed string if you find it
    df=QuantifyReactive()$counts
    for ( col in 1:ncol(df)){
      colnames(df)[col] <- sub("_trimmed", "", colnames(df)[col])
    }
    #Remove .fastq.subread.SAM if you find it
    for ( col in 1:ncol(df)){
      colnames(df)[col] <- sub("*.fastq.subread.SAM$", "", colnames(df)[col])
    }
    #Remove .fastq.subread.BAM string if you find it
    for ( col in 1:ncol(df)){
      colnames(df)[col] <- sub("*.fastq.subread.BAM$", "", colnames(df)[col])
    }
    #Change - string to _
    for ( col in 1:ncol(df)){
      colnames(df)[col] <- sub("-", "_", colnames(df)[col])
    }
    #Remove R1 string if you find it
    for ( col in 1:ncol(df)){
      colnames(df)[col] <- sub("R1", "", colnames(df)[col])
    }

    write.csv(df, file, row.names=TRUE)
  }
)

#Render the differentially expressed genes as a data table
output$Table2 <- renderDataTable({
  differ_expr<- ResultsReactive()$differ_expr

  if(!is.null(differ_expr)){
    resultDF =differ_expr
  }
})

#Download the differentially expressed genes table in .csv format
output$downloadDDSCSV2 <- downloadHandler(
  filename = function() {paste0("dds",".csv")},
  content = function(file) {
    differ_expr<- ResultsReactive()$differ_expr

    write.csv( differ_expr, file, row.names=TRUE)
  }
)

```

```

#Render the enriched genes as a data table
output$gseGoTable <- renderDataTable({
  gseGo <- gseGoReactive()

  if(!is.null(gseGo)){
    resultDF = gseGo$go_gse@result

    DT::datatable(resultDF, options = list(scrollX = TRUE, columnDefs =
list(list(visible=input$showAllColumns, targets= 10:12 )) ))
  }

})

#Download the enriched genes in .csv format
output$downloadgseGoCSV <- downloadHandler(
  filename = function() {paste0("gsego",".csv")},
  content = function(file) {

    write.csv(gseGoReactive()$go_gse@result, file, row.names=TRUE)}
)

#Render Dotplot
output$dotPlot = renderPlot({
  go_gse = gseGoReactive()$go_gse
  plot<-dotplot(go_gse,
    #number of enriched terms to display
    showCategory = input$showCategory_dot,
    font.size = 8,
    #separate result by 'category' variable
    split=".sign") + facet_grid(.~.sign)

  myValues$plot<-plot
  print(plot)
})

#Download Dotplot
output$downloadPlot <- downloadHandler(
  filename = function(){paste("dotplot", '.pdf', sep = '')},

  content = function(file){
    pdf(file, width = 8, height = 8)
    print(myValues$plot)
    dev.off()
  })

#Render ridgePlot
output$ridgePlot = renderPlot({
  go_gse = gseGoReactive()$go_gse
  ridge<- ridgeplot(go_gse,
    #number of enriched terms to display
    showCategory = input$showCategory_ridge
  )

  myValues$ridge<-ridge
  print(ridge)
})

#Download ridgePlot
output$downloadPlot2 <- downloadHandler(
  filename = function(){paste("ridge", '.pdf', sep = '')},

  content = function(file){
    pdf(file, width = 8, height = 8)
    print(myValues$ridge)
    dev.off()
  })

```

```

#Render gseaplot
output$gseaplot = renderPlot({
  go_gse = gseGoReactive()$go_gse
  gsea<- gseaplot(go_gse,
                  #"runningScore" and "position"
                  by = "all",
                  title = go_gse$Description[input$geneSetId_gsea],
                  geneSetID = input$geneSetId_gsea)

  myValues$gsea<-gsea
  print(gsea)

})

#Download gseaplot
output$downloadPlot3 <- downloadHandler(
  filename = function(){paste("gsea", '.pdf', sep = '')},

  content = function(file){
    pdf(file, width = 8, height = 8)
    print(myValues$gsea)
    dev.off()
  })

#Render cnetplot
output$cnetplot = renderPlot({
  go_gse = gseGoReactive()$go_gse
  cnet<-cnetplot(go_gse,
                 categorySize="pvalue",
                 foldChange= myValues$gene_list,
                 #number of enriched terms to display
                 showCategory = input$showCategory_cne)

  myValues$cnet<-cnet
  print(cnet)

})

#Download cnetplot
output$downloadPlot4 <- downloadHandler(
  filename = function(){paste("cnet", '.pdf', sep = '')},

  content = function(file){
    pdf(file, width = 8, height = 8)
    print(myValues$cnet)
    dev.off()
  })

#Render heatplot
output$heatplot = renderPlot({
  go_gse = gseGoReactive()$go_gse
  heat<-heatplot(go_gse, foldChange= myValues$gene_list)

  myValues$heat<-heat
  print(heat)

})

#Download heatplot
output$downloadPlot5 <- downloadHandler(
  filename = function(){paste("heatplot", '.pdf', sep = '')},

  content = function(file){
    pdf(file, width = 8, height = 8)
    print(myValues$heat)
  })

```

```

    dev.off()
  })

#Render maplot
output$maplot2 = renderPlot({
  if(!is.null(ResultsReactive()))
  {
    de<- myValues$res1

    ma<-ggmaplot(de,
      #Accepted false discovery rate for considering genes as
      differentially expressed.
      fdr = 0.05,
      #the fold change threshold.
      #Only genes with a fold change >= fc and padj <= fdr are considered
      as significantly differentially expressed.
      fc = 1,
      #points size
      size = 0.6,
      palette = c(input$col1,input$col2,input$col3),
      genenames = as.vector(de$name),
      legend = "top", top = 0,
      #select.top.method = c( "fc"),
      font.label = c("bold", 11),
      font.legend = "bold",
      font.main = "bold",
      #ggplot2 theme name
      ggtheme = ggplot2::theme_minimal())

    myValues$ma<-ma
    print(ma)
  }
})

#Download maplot
output$downloadPlotma <- downloadHandler(
  filename = function() {paste("ma_plot", ".pdf", sep = "")},

  content = function(file) {
    pdf(file, width = 8, height = 8)
    print(myValues$ma)
    dev.off()
  })

#Render volcanoplot
output$volcanoplot2 = renderPlot({
  if(!is.null(ResultsReactive()))
  {

    res1<- myValues$res1
    res2<- as.data.frame(res1)

    # add a column of NAs
    res2$diffexpressed <- "NO"
    # if log2Foldchange > 0.6 and pvalue < 0.05, set as "UP"
    res2$diffexpressed[res2$log2FoldChange > 1 & res2$pvalue < 0.05] <- "UP"
    # if log2Foldchange < -0.6 and pvalue < 0.05, set as "DOWN"
    res2$diffexpressed[res2$log2FoldChange < -1 & res2$pvalue < 0.05] <- "DOWN"

    vol2<- ggplot(data=res2, aes(x=log2FoldChange, y=-log10(pvalue),
    col=diffexpressed)) +
      geom_point() +
      theme_minimal() +
      #color choice and add lines
      scale_color_manual(values=c(input$col1,input$col3,input$col2)) +
      geom_vline(xintercept=c(-1, 1), col="black") +
      geom_hline(yintercept=-log10(0.05), col="black")
  }
})

```

```

myValues$vol2<-vol2
print(vol2)
}
})

#Download volcanoplot
output$downloadPlotvol <- downloadHandler(
  filename = function() {paste("volcano_plot", '.pdf', sep = '')},

  content = function(file) {
    pdf(file, width = 8, height = 8)
    print(myValues$vol2)
    dev.off()
  })

#Update condition choices for pcaplot
observe({

  updateSelectInput(session, 'group',
    choices= colnames(myValues$DF),
    selected = colnames(myValues$DF)[1])

})

#Render pcaplot
output$pcaplot = renderPlot({
  if(!is.null(ResultsReactive()))
  {
    intgroups=input$group
    #pca function from DESeq2
    pca<- plotPCA(myValues$vsd, intgroup=intgroups)

    myValues$pca<-pca
    print(pca)
  }
})

#Download pcaplot
output$downloadPlot7 <- downloadHandler(
  filename = function() {paste("pca", '.pdf', sep = '')},

  content = function(file) {
    pdf(file, width = 8, height = 8)
    print(myValues$pca)
    dev.off()
  })

#Update gene choices for boxplot
observe({

  updateSelectizeInput(session, 'sel_gene',
    choices= myValues$names,
    server=TRUE)

})

#Update condition choices for boxplot
observe({

  updateSelectInput(session, 'boxplotX',
    choices= colnames(myValues$DF),
    selected = colnames(myValues$DF)[1])

  updateSelectInput(session, 'boxplotFill',

```

```

        choices= colnames(myValues$DF),
        selected = colnames(myValues$DF)[1])

    })

    observe({
      geneExprReactive()
    })

    #
    geneExprReactive <- reactive({

      validate(need(length(input$sel_gene)>0,"Please select a gene. "))
      #filter genes by log2
      filtered <- t(log2((counts(myValues$dds[input$sel_gene, ], normalized=TRUE,
replaced=FALSE)+.5)))) %>%
        merge(colData(myValues$dds), ., by="row.names") %>%
        gather(gene, expression, (ncol(.)-length(input$sel_gene)+1):ncol(.))

    })
    #Render boxPlot for gene expression
    output$boxPlot <- renderPlot({
      if(!is.null(geneExprReactive()))
      {
        filtered = geneExprReactive()

        validate(need(length(input$boxplotX)>0,"Please select a group. "))
        validate(need(length(input$boxplotFill)>0,"Please select a fill by
group. "))

        p <- ggplot(filtered, aes_string(input$boxplotX, "expression",
fill=input$boxplotFill)) +
          geom_boxplot() +
          scale_fill_manual(values=c(input$col1,input$col2))+facet_wrap(~gene,
scales="free_y")

        myValues$p<-p
        print(p)

      }

    })

    #Download boxPlot for gene expression
    output$downloadPlot8 <- downloadHandler(
      filename = function(){paste("boxplot", '.pdf', sep = ' ')},

      content = function(file){
        pdf(file, width = 8, height = 8)
        print(myValues$p)
        dev.off()
      })

    #Update condition choices for heatmap
    observe({

      updateSelectInput(session,'group1',
        choices= colnames(myValues$DF),
        selected = colnames(myValues$DF)[1])

      updateSelectInput(session,'group2',
        choices= colnames(myValues$DF),
        selected = colnames(myValues$DF)[1])
    })

```

```

))

observe({
  heatmapReactive()
})

heatmapReactive <- reactive({
  if(!is.null(ResultsReactive()))

  #for 1 condition
  if (input$n == 1)
  {
    isolate({

      df <- as.data.frame(colData(myValues$dds)[, (input$group1)])

      row.names(df) <- colnames(myValues$vsd)
      colnames(df)[1] <- input$group1
      #select the top 30 differentially expressed genes
      topVarGenes <- head(order(-rowVars(assay(myValues$vsd))), 30)
      myValues$df <- df
      mat <- assay(myValues$vsd)[ topVarGenes, ]
      mat <- mat - rowMeans(mat)

      return(mat)
    })
  }

  #for 2 conditions
  if (input$n == 2) {
    isolate({

      df <-
as.data.frame(colData(myValues$dds)[, c(input$group1, input$group2)])
      row.names(df) <- colnames(myValues$vsd)
      #select the top 30 differentially expressed genes
      topVarGenes <- head(order(-rowVars(assay(myValues$vsd))), 30)
      myValues$df <- df

      mat <- assay(myValues$vsd)[ topVarGenes, ]
      mat <- mat - rowMeans(mat)

      return(mat)
    })
  }

})

#Render heatmap
output$heatmap <- renderPlot({
  if(!is.null(heatmapReactive()))
  {
    mat=heatmapReactive()

    heatmap<- pheatmap(mat,
      annotation_col=myValues$df,
      scale = "row",
      cluster_rows = T,
      angle_col = 90,

    )

    myValues$heatmap<-heatmap
    print(heatmap)
  }
})

```

```

#Download heatmap
output$downloadPlot6 <- downloadHandler(
  filename = function() {paste("heatmap", '.pdf', sep = '')},

  content = function(file){
    pdf(file, width = 8, height = 8)
    print(myValues$heatmap)
    dev.off()
  })

#Render boxplot for counts
output$boxplot2 <- renderPlot({
  if(!is.null(myValues$dds))
  {
    min_nonzero=1
    #filtered counts by log2
    b<-boxplot(log2(myValues$countDataMatrix+min_nonzero),
               col= brewer.pal(n = ncol(myValues$countDataMatrix),
                                name = input$Palette),
               names=myValues$sample_names,
               las=2,
               ylab="log2(counts)",
               main="Distribution of counts for all samples")
    myValues$b<-b
    print(b)
  }
})

}
shinyApp(ui=ui, server=server)

```

7. Βιβλιογραφία

1. Giani, A. M., Gallo, G. R., Gianfranceschi, L. & Formenti, G. Long walk to genomics: History and current approaches to genome sequencing and assembly. *Computational and Structural Biotechnology Journal* vol. 18 9–19 Preprint at <https://doi.org/10.1016/j.csbj.2019.11.002> (2020).
2. Pease, J. & Sooknanan, R. A rapid, directional RNA-seq library preparation workflow for Illumina® sequencing. *Nat Methods* **9**, i–ii (2012).
3. Kumar Mondal, R. & Dey, A. *BIOINFORMATICS: PAST, PRESENT AND FUTURE. International Research Journal of Modernization in Engineering Technology and Science* www.irjmets.com @International Research Journal of Modernization in Engineering vol. 442 www.irjmets.com.
4. <https://cran.r-project.org/bin/windows/base/>.
5. <https://posit.co/download/rstudio-desktop/>.
6. *Package ‘dashboardthemes’*. (2022).
7. *Type Package Title An R Wrapper for Bowtie2 and AdapterRemoval*. <https://git.bioconductor.org/packages/Rbowtie2> (2023).
8. Wei Shi, A., Liao, Y., Smyth with contributions from Jenny Dai Maintainer Wei Shi, G. K. & Smyth, G. K. *Title Mapping, quantification and variant analysis of sequencing data*. (2023).
9. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* **15**, (2014).
10. https://hbctraining.github.io/DGE_workshop/lessons/05_DGE_DESeq2_analysis2.html.
11. *Type Package Title Client to access AnnotationHub resources*. (2023).
12. Wu, T. et al. clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. *Innovation* **2**, (2021).
13. *Package ‘genekitr’ Type Package Title Gene Analysis Toolkit*. (2023).
14. McDermaid, A., Monier, B., Zhao, J., Liu, B. & Ma, Q. Interpretation of differential gene expression results of RNA-seq data: Review and integration. *Briefings in Bioinformatics* vol. 20 2044–2054 Preprint at <https://doi.org/10.1093/bib/bby067> (2019).
15. Subramanian, A. et al. *Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles*. www.pnas.org/cgi/doi/10.1073/pnas.0506580102 (2005).

16. Haque, A., Engel, J., Teichmann, S. A. & Lönnberg, T. A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications. *Genome Medicine* vol. 9 Preprint at <https://doi.org/10.1186/s13073-017-0467-4> (2017).
17. Peymani, F., Farzeen, A. & Prokisch, H. RNA sequencing role and application in clinical diagnostic. *Pediatric Investigation* vol. 6 29–35 Preprint at <https://doi.org/10.1002/ped4.12314> (2022).
18. Marco-Puche, G., Lois, S., Benítez, J. & Trivino, J. C. RNA-Seq Perspectives to Improve Clinical Diagnosis. *Frontiers in Genetics* vol. 10 Preprint at <https://doi.org/10.3389/fgene.2019.01152> (2019).

