ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ

Ταυτοποίηση βιοδεικτών χρησιμοποιώντας αναπαράσταση με k-mers και ταξινόμηση πολλαπλών ετικετών

Μάκρα Χρυσούλα

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Υπεύθυνος
Άρτεμις Χατζηγεωργίου
Καθηγήτρια

Λαμία, 2022-2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ

Ταυτοποίηση βιοδεικτών χρησιμοποιώντας αναπαράσταση με k-mers και ταξινόμηση πολλαπλών ετικετών

Μάκρα Χρυσούλα

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων/σα
Άρτεμις Χατζηγεωργίου
Καθηγήτρια

Λαμία, 2022-2023

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 επεξεργάσθηκαν, 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.

Ταυτοποίηση βιοδεικτών χρησιμοποιώντας αναπαράσταση με k-mers και ταξινόμηση πολλαπλών ετικετών

Μάκρα Χρυσούλα

Τριμελής Επιτροπή:

Ονοματεπώνυμο, Βαθμίδα …..(επιβλέπων/σα)

Ονοματεπώνυμο, Βαθμίδα……

Ονοματεπώνυμο, Βαθμίδα…….

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μεθοδολογίας με την χρήση τεχνικών μηχανικής μάθησης και δεδομένων DNA ακολουθιών συγκεκριμένου μήκους (k-mers) για τον εντοπισμό πιθανών γενετικών μοριακών δεικτών που σχετίζονται με αλληλεπιδράσεις ξενιστή – παθογόνου μικροοργανισμού. Η ανάλυση αποσκοπεί στην αναγνώριση πιθανών k-mers που να έχουν στατιστικά σημαντική συσχέτιση με τον ξενιστή από τον οποίο απομονώνεται το εκάστοτε δείγμα DNA του μικροοργανισμού. Οι k-mers αυτοί ενδέχεται να σχετίζονται με τη λειτουργική προσαρμογή του μικροοργανισμού στον εκάστοτε ξενιστή και πιθανώς μπορούν να χρησιμοποιηθούν για την εκπαίδευση μοντέλου πρόβλεψης της προέλευσης παθογόνων μικροβίων, γεγονός που θα είχε ιδιαίτερη αξία στο πεδίο των ζωονόσων. Ως πιλοτικό σενάριο, θα γίνει εφαρμογή δεδομένων που είναι διαθέσιμα για δυνητικά και μη παθογόνους μικροοργανισμούς, όπως το Gram αρνητικό βακτήριο Klebsiella pneumoniae.

Η επιλογή του συγκεκριμένου θέματος έγινε με σκοπό την αξιοποίηση της πτυχιακής εργασίας ως ευκαιρία να γνωρίσω για πρώτη φορά τον τομέα της μηχανικής μάθησης. Η πρακτική εφαρμογή της εργασίας βοήθησε πολύ σε αυτό το σκοπό και η εργασία η ίδια είναι το αποτέλεσμα του χρόνου που αφιέρωσα για τη διεκπεραίωση της. Η επιλογή θέματος επί του οποίου υπάρχει έλλειψη θεωρητικών γνώσεων μπορεί να στάθηκε τροχοπέδη στη πορεία της ανάλυσης αλλά δεν έμεινε εμπόδιο στην ολοκλήρωσή της. Κλείνοντας να προσθέσω πως η εργασία έχει υλοποιηθεί με τη μορφή κώδικα σε γλώσσα R και τα αποτελέσματά εμφανίζονται στη πορεία της ανάλυσης

Λέξεις – κλειδιά: μηχανική μάθηση, k-mers, ταξινόμηση πολλαπλών ετικετών, επιλογή χαρακτηριστικών, σημασία χαρακτηριστικών

Abstract

The aim of this thesis is the development of a methodology, using machine learning techniques and specific length DNA sequences (k-mers) to identify possible genetic molecular markers related to host-pathogen interactions. The analysis aims to identify potential k-mers that have a statistically significant correlation with the host from which the respective DNA sample of the microorganism is isolated. These k-mers may be related to the functional adaptation of the microorganism to the respective host and possibly can be used to train a model to predict the origin of pathogenic microbes, which would be of particular value in the field of zoonosis. As a pilot project, data available on potential and non-pathogenic microorganisms, such as the Gram-negative bacterium Klebsiella pneumoniae, will be applied.

The choice of the specific topic was made in order to use the thesis as an opportunity to familiarize myself with the field of machine learning for the first time. The practical application of the project helped a great deal in this end and the work itself is the result of the time I have devoted to complete it. Picking a topic on which there is a lack of theoretical knowledge may have hindered the course of the analysis but did not stand in the way of its completion. Finally, I would like to add that the work has been implemented in the form of code in R language and the results are displayed during the analysis

Keywords: machine learning, k-mers, multi-class classification, feature selection, feature/ variable importance

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ-ΣΧΗΜΑΤΩΝ

1. Εισαγωγή

Η μηχανική μάθηση έχει χρησιμοποιηθεί για τη λύση προβλημάτων σε ένα μεγάλο εύρος τομέων, όπως στη Βιομηχανία, στη Ρομποτική, στη Βιολογία, στην Ιατρική και στις επιστήμες Υγείας, στα Οικονομικά κ.α. Στη παρούσα εργασία θα δούμε την εφαρμογή στον τομέα της Βιοπληροφορικής και της Υγείας, με μία εφαρμογή στο κομμάτι των ζωονόσων. Αυτό γιατί το θέμα είναι η αξιοποίηση τεχνικών μηχανικής μάθησης για την ανάλυση δεδομένων αλληλουχιών DNA, μέσω αναπαράστασής τους ως k-mers με σκοπό την αναζήτηση πιθανών γενετικών μοριακών δεικτών που σχετίζονται με αλληλεπιδράσεις ξενιστή – παθογόνου μικροοργανισμού. Η υπόθεση που μελετάται είναι η περίπτωση να υπάρχουν k-mers που να έχουν στατιστικά σημαντική συσχέτιση με τον ξενιστή από τον οποίο απομονώνεται το εκάστοτε δείγμα DNA του μικροοργανισμού. Αυτό γιατί ενδέχεται οι k-mers αυτοί να σχετίζονται με τη λειτουργική προσαρμογή του μικροοργανισμού στον εκάστοτε ξενιστή, μπορεί να βρίσκονται σε γονίδια ή να είναι κομμάτι μεταλλάξεων μεταξύ των διαφόρων ξενιστών. Έτσι ενδέχεται να βρεθούν γενετικοί μοριακοί βιοδείκτες που υποδεικνύουν την συσχέτιση των μικροοργανισμών με τους αντίστοιχους ξενιστές.

Στα επόμενα κεφάλαια βρίσκεται η μεθοδολογία, ξεκινώντας με τη συλλογή των δεδομένων. Ως πιλοτικό σενάριο, γίνεται εφαρμογή δεδομένων που είναι διαθέσιμα για δυνητικά και μη παθογόνους μικροοργανισμούς, όπως το Gram αρνητικό βακτήριο Klebsiella Pneumoniae. Συλλέγονται δηλαδή πληροφορίες για δείγματα βακτηρίου από διάφορους ξενιστές μαζί με το γονιδίωμά τους. Με αυτή τη πληροφορία δημιουργούνται τα k-mers, συγκεκριμένα αλληλουχίες 9 βάσεων και η συχνότητα εμφάνισης τους στον κάθε οργανισμό σε κάθε δείγμα. Στη συνέχεια, αυτά τα δεδομένα επεξεργάζονται μερικώς ώστε να έρθουν σε μια μορφή κατάλληλη για την ανάλυση. Σημαντικό βήμα της διαδικασίας διότι όσο καλύτερα δεδομένα δίνονται στο μοντέλο, τόσο καλύτερα αποτελέσματα παράγει, ή αλλιώς, «garbage in, garbage out". Τα δεδομένα αυτά δίνονται στο μοντέλο, το οποίο θα χρειαστεί να ταξινομήσει δείγματα σε ξενιστές προέλευσης μεταξύ πολλών επιλογών. Η τεχνική που εφαρμόζεται σε αυτή τη περίπτωση ονομάζεται εποπτευόμενη μάθηση (supervised learning) για πολλαπλή κατηγοριοποίηση (multi- label classification). Εκτελώντας το μοντέλο πρόβλεψης/ ταξινόμησης δειγμάτων σε οργανισμούς προέλευσης αναζητούνται τα k-mers που βοήθησαν περισσότερο, υποθέτοντας πως αυτά ίσως

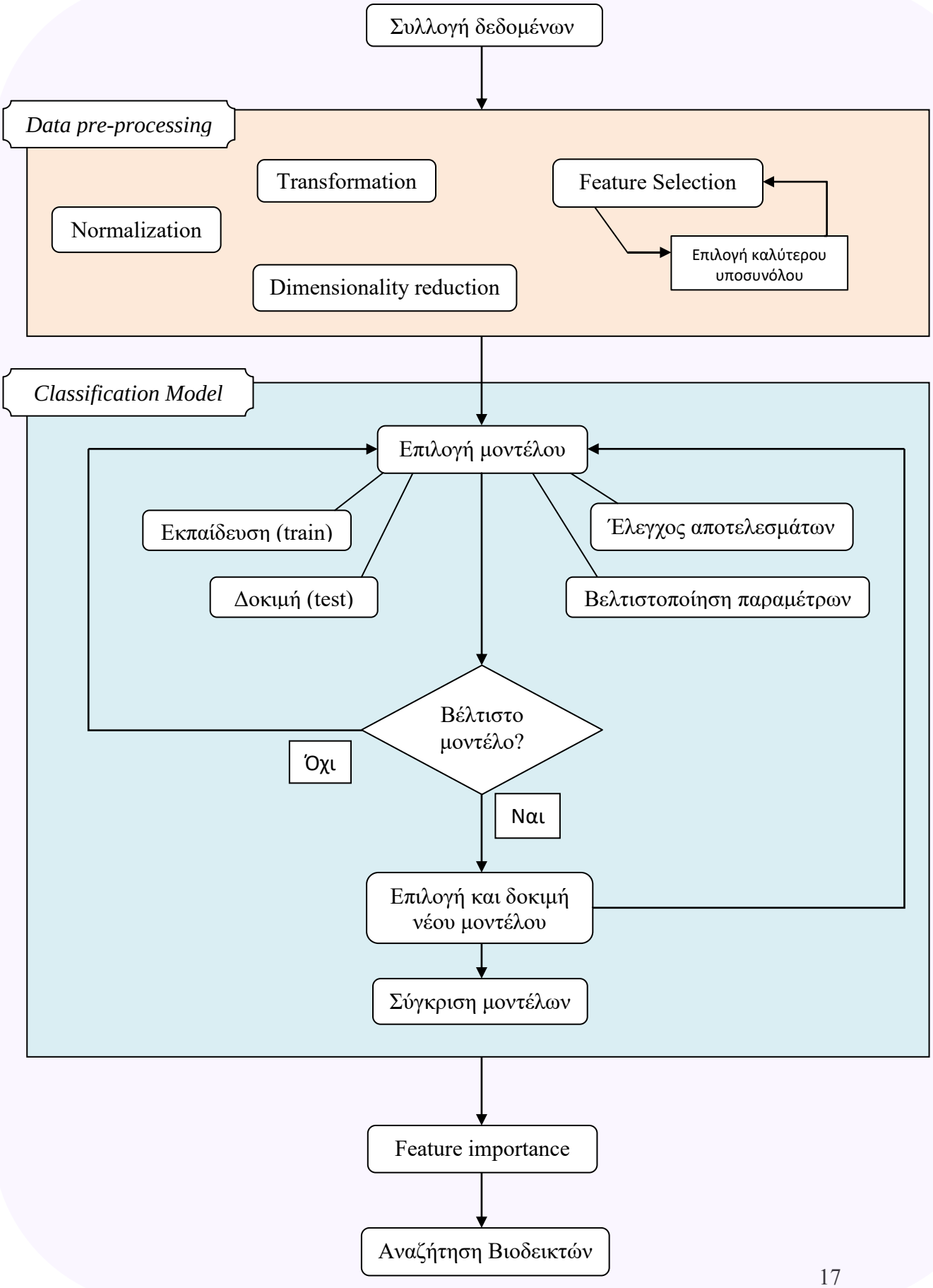
μπορούν να οδηγήσουν σε μοριακούς βιοδείκτες. Θα χρειαστεί να μελετηθεί η τοποθεσία των κμερς στο γονιδίωμα, σε ποιον ξενιστή συναντώνται σε σύγκριση με τους άλλους, αν είναι μοναδικοί ή αν αυτό που διαφέρει είναι η συχνότητα εμφάνισης και αν μπορεί να χαρακτηρίσει τελικά κάποιο μοριακό/ γενετικό βιοδείκτη. Μια αναπαράσταση της διαδικασίας φαίνεται παρακάτω στο διάγραμμα ροής και η ανάλυση των βημάτων στο κεφάλαιο Μεθοδολογία. Στα επόμενα κεφάλαια είναι τα αποτελέσματα που βγήκαν σε κάθε βήμα της μεθοδολογίας αναλυτικά και η συζήτηση όπου περιλαμβάνει σχόλια και παρατηρήσεις από διάφορα μέρη της ανάλυσης. Στο τέλος βρίσκονται τα συμπεράσματα της μελέτης και η βιβλιογραφία.

Σε ορισμένα κεφάλαια της μεθοδολογίας θεωρήθηκε σημαντική η θεωρητική περιγραφή των θεμάτων και βρίσκεται ως αριθμημένο κεφάλαιο (βλ. ΠΕΡΙΕΧΟΜΕΝΑ).

Τέλος, ο κώδικας που χρησιμοποιήθηκε για την εκπόνηση της εργασίας υλοποιήθηκε χρήση γλώσσας R και εφαρμόστηκε στο Rstudio. Ο κώδικας βρίσκεται στον παρακάτω σύνδεσμο του GitHub .

https://github.com/XrysaM/KlebsiellaML/blob/master/code.R

2. Διάγραμμα ροής



3. Μεθοδολογία

3.0 Βήματα

1. **Συλλογή δεδομένων**
    a. Εξαγωγή μέσω ENA Pathogens
    b. Paired end Illumina sequencing
    c. Απόρριψη δειγμάτων, ξενιστών
    d. Εξισορρόπηση δεδομένων
    e. Trim galore
        i. Φιλτράρισμα
        ii. Αποκοπή adapter sequences
    f. De novo genome assembly
    g. Kleborate
    h. AMR FinderPlus

2. **Προετοιμασία δεδομένων**
    a. Καταμέτρηση k-mer
    b. Δημιουργία πινάκων:
        i. Μετρήσεις k-mer
        ii. Μετά-δεδομένα
    c. Log2(X+1) transformation
    d. z-score normalization
    e. Feature selection
        i. Boruta
        ii. RFE
        iii. Correlation
        iv. Variance
    f. Dimensionality Reduction
        i. PCA
        ii. LDA
        iii. MDS

3. **Μέθοδος Κατηγοριοποίησης δεδομένων**
    a. Multi-class classification
        i. Random Forest
        ii. Binary classification
            ¬ Random Forest One vs All
    b. Σύγκριση μοντέλων
    c. Εξαγωγή σημαντικών k-mers (variable importance)

4. **Αναζήτηση Βιοδεικτών**
    a. BLAST
    b. Αναζήτηση γονιδίων

Εδώ σημαντικό είναι να αναφερθεί η συμβολή του Ινστιτούτου Εφαρμοσμένων Βιοεπιστημών του Εθνικού Κέντρου Έρευνας και Τεχνολογίας που προσέφερε τα πρώτα δεδομένα της εργασίας αυτής και συγκεκριμένα τα βήματα από τη συλλογή των δεδομένων στην αρχή μέχρι τα πρώτα βήματα της επεξεργασίας τους. Συγκεκριμένα, στα Βήματα παραπάνω είναι τα επίπεδα "1. Συλλογή δεδομένων" και στο "2. Προετοιμασία δεδομένων" τα "2.a. Καταμέτρηση k-mer", "2.b. Δημιουργία πινάκων:" , "2.c. Log2(X+1) transformation", "2.d. z-score normalization". Επιγραμματικά η μεθοδολογία των βημάτων αυτών είναι παρακάτω στα κεφάλαια.

3.1 Συλλογή δεδομένων

Τα δεδομένα του βακτηρίου Klebsiella Pneumoniae έχουν συλλεχθεί από τη βάση δεδομένων ENA Pathogens. **Ανακτήθηκαν** 746 δείγματα, τα οποία απομονώθηκαν από ανθρώπους και ξενιστές ζώων από 47 χώρες, κατά τη διάρκεια διαφορετικών χρονικών περιόδων και αλληλουχήθηκαν χρησιμοποιώντας τεχνικές αλληλούχισης paired-end Illumina. **Απορρίφθηκαν** δείγματα που είχαν αλληλουχηθεί αποκλειστικά με τεχνικές PacBio ή Nanopore λόγω μικρού αριθμού δειγμάτων και δείγματα που συλλέχθηκαν από περιβαλλοντικές πηγές ή από λιανικές αγορές. Τα δείγματα K. Pneumoniae που απομονώθηκαν από ανθρώπους ξενιστές αποτελούν την πλειοψηφία των διαθέσιμων δειγμάτων, επομένως επιλέχθηκε ένα τυχαίο υποσύνολο 250 δειγμάτων προκειμένου να αποφευχθεί η ανισορροπία των δεδομένων.

Το **φιλτράρισμα** με βάση την ποιότητα ανάγνωσης και την περικοπή των adapter sequences Illumina πραγματοποιήθηκε χρησιμοποιώντας το Trim galore στις προεπιλεγμένες ρυθμίσεις. Για κάθε απομόνωση, η **de novo συγκρότηση γονιδιώματος** πραγματοποιήθηκε χρησιμοποιώντας SPAdes. Ο **ποιοτικός έλεγχος** στα σύνολα αυτά πραγματοποιήθηκε χρησιμοποιώντας το QUAST. Τα σύνολα σχολιάστηκαν περαιτέρω για είδη, πληροφορίες γενεαλογίας (Sequence Type; ST) και προφίλ αντοχής στα αντιβιοτικά και λοιμογόνου δράσης χρησιμοποιώντας Kleborate. Σχολιάστηκαν επιπλέον για την παρουσία γονιδίων που σχετίζονται με την αντίσταση στα αντιβιοτικά, τη λοιμογόνο δύναμη και το στρες χρησιμοποιώντας το AMRFinderPlus. Οι ξενιστές άγριων ζώων με μικρό αριθμό δειγμάτων **εξαιρέθηκαν** από την ανάλυση ταξινόμησης.

3.2 Προετοιμασία δεδομένων (pre-processing)

Η κανονική μέτρηση k-mer, με k=9 και απόρριψη μοναδικών k-mers (k-mer count=1), πραγματοποιήθηκε με το Jellyfish χρησιμοποιώντας ως είσοδο τα quality-filtered και adapter-trimmed sequencing read files. Οι μετρήσεις K-mer μετατράπηκαν σε μορφή πίνακα μέσω R, με δείγματα (n=680) ως σειρές(samples) και k-mers (n=131076) ως στήλες (features). Για την διαδικασία της μηχανικής μάθησης χρησιμοποιείται μια επιπλέον στήλη μόνο, η οποία λέγεται host_categories και αναφέρει κάθε δείγμα από ποιόν οργανισμό προήλθε. Τέλος ,πραγματοποιήθηκαν ανά δείγμα **Log2(X+1) transformation** και **z-score normalization** για να αφαιρεθεί η μεροληψία από το διαφορετικό βάθος ανάγνωσης.

3.2.1 Feature selection

0) <u>Θεωρία</u>

Σε δεδομένα όπως αυτά που συλλέχθηκαν, τα οποία έχουν πολλές πληροφορίες και συγκεκριμένα πολλά χαρακτηριστικά (features/ variables) καλό είναι εα εφαρμοστεί feature selection. Feature selection ή επιλογή χαρακτηριστικών είναι η διαδικασία επιλογής ενός υποσυνόλου χαρακτηριστικών κρατώντας αυτά που είναι πιο σημαντικά στη πορεία της ανάλυσης. Υπάρχουν πολλοί λόγοι για να εφαρμόσει κανείς feature selection. Σε γενικές γραμμές η μείωση των χαρακτηριστικών βοηθάει να μειωθεί το υπολογιστικό κόστος ή/και να παραχθούν καλύτερα αποτελέσματα.

Το feature selection εφαρμόζεται πριν την εφαρμογή του αλγορίθμου μηχανικής μάθησης για να προετοιμάσει τα δεδομένα. Δεν είναι υποχρεωτικό να εφαρμοστεί διότι τα μοντέλα από μόνα τους μπορούν να ασχοληθούν με χαρακτηριστικά που χρειάζονται, αλλά πρακτικά μπορεί να εμφανιστούν προβλήματα. Για αρχή ένας απλός αλλά σημαντικός παράγοντας είναι ο χρόνος. Όσο περισσότερα χαρακτηριστικά έχουμε τόσο περισσότερο χρόνο θα πάρει να εκπαιδεύσουμε το μοντέλο. Ανάλογα το μέγεθος των δεδομένων η διαφορά αυτή μπορεί να είναι λίγα λιγότερα λεπτά, ώρες ή ακόμα και μέρες υπολογισμών. Άλλος παράγοντας είναι η παρουσία χαρακτηριστικών που δεν είναι χρήσιμα για τη πρόβλεψη (irrelevant features) ή χαρακτηριστικών που μοιράζονται την ίδια πληροφορία (redundant features). Αυτό προκαλεί το μοντέλο να εκπαιδευτεί γύρω από αυτά με τόση ακρίβεια

που μπορεί να μην καταφέρει να κατηγοριοποιήσει σωστά ή με ακρίβεια νέα δεδομένα. Αυτό ονομάζεται υπεργενίκευση ή υπερμοντελοποίηση (overfitting). Επίσης πολλά χαρακτηριστικά σημαίνει μεγάλη πολυπλοκότητα που επίσης προκαλεί overfitting. Επομένως με τη μείωση των μεταβλητών μειώνεται η περιπλοκότητα , ο χρόνος εκπαίδευσης και οι πιθανότητες υπεργενίκευσης του μοντέλου μεταξύ άλλων.

Υπάρχουν πολλοί μέθοδοι που μπορούμε να χρησιμοποιήσουμε, χωρίζονται όμως κατά βάση σε 3 βασικές κατηγορίες, Filter, Wrapper, και Embedded. Οι κατηγορίες αυτές υποκύπτουν στη κατηγορία της επιβλεπόμενης εκπαίδευσης (supervised learning) σε αντίθεση με την μη επιβλεπόμενη (unsupervised) που δεν χρησιμοποιεί τη μεταβλητή στόχο. Unsupervised methods χρησιμοποιούνται για να ελέγξουν τα δεδομένα καθ' αυτά και τη συσχέτιση μεταξύ τους. Για παράδειγμα features που έχουν μικρή διακύμανση(low variance) δεν προσφέρουν πολλή πληροφορία στο μοντέλο. Επίσης είναι χρήσιμο, όπως αναφέρθηκε ήδη, η αναζήτηση των redundant features, μεταβλητών δηλαδή που μοιράζονται την ίδια πληροφορία ή έχουν μεγάλη στατιστική συσχέτιση (high correlation) με άλλα και η επιλογή ενός εξ αυτών.

Filter methods είναι αυτοί που βασίζονται σε αρχές στατιστικής ανάλυσης, συγκρίνουν τα features με το χαρακτηριστικό στόχο που μας ενδιαφέρει να προβλέψουμε με σκοπό να βρει τα features που δεν έχουν στατιστικά σημαντική συσχέτιση με το target. Υπάρχει πολύ μεγάλη ποικιλία τεστ που υπάγονται σε αυτή τη κατηγορία, καθένα προσφέρει άλλη προσέγγιση στο πρόβλημα. Το ποιο είναι το σωστό εργαλείο βασίζεται στις γνώσεις του χρήστη στη στατιστική ανάλυση και το τύπο των δεδομένων. Στη παρούσα εργασία δίνεται ως είσοδος το πλήθος των k-mer σε κάθε δείγμα επομένως είναι αριθμητικά δεδομένα (numeric/ interval) και ως έξοδο είναι ο οργανισμός προέλευσης των δειγμάτων άρα κατηγορική μεταβλητή (categorical/ nominal). Τυπικά παραδείγματα στατιστικών μεθόδων που θα μπορούσαν να χρησιμοποιηθούν είναι τα Pearson's correlation, ANOVA f-score, Kendall rank coefficient, Spearman's rank correlation. Οι μέθοδοι filter είναι γρήγοροι, δεν χρειάζονται εκπαίδευση μοντέλου εφόσον βασίζονται σε στατιστικές μεθόδους αλλά βρίσκουν μόνο τη συσχέτιση των δεδομένων εισόδου με τη μεταβλητή στόχο, αγνοώντας τις σχέσεις των μεταβλητών μεταξύ τους.

Οι Wrapper methods χρησιμοποιούν τη τεχνική των αλγορίθμων μηχανικής μάθησης, εκπαιδεύουν δηλαδή μοντέλα χρησιμοποιώντας διάφορα υποσύνολα των δεδομένων για να βρουν το καλύτερο (cross-validation). Διαφορετικοί wrapper

μέθοδοι βασίζονται σε διαφορετικά μοντέλα μηχανικής μάθησης και έτσι μπορούν να προετοιμάσουν με μεγάλη ακρίβεια τα δεδομένα για τη χρήση του αντίστοιχου μοντέλου. Όμως όπως είναι καλύτερα για τα συγκεκριμένα μοντέλα, είναι αντίστοιχα αδύναμοι κριτές για την εύρεση του κατάλληλου υποσυνόλου όταν θα χρησιμοποιηθούν άλλα μοντέλα για τη κατηγοριοποίηση. Το υποσύνολο δηλαδή που θα κρίνουν ως το καλύτερο θα είναι πολύ καλό για το αντίστοιχο μοντέλο του οποίου τη τεχνική χρησιμοποιούν αλλά όχι τόσο καλό για άλλα. Επίσης οι wrappers για να βγάλουν τέτοια αποτελέσματα χρειάζεται να δημιουργήσουν πολλά μοντέλα και αυτό μπορεί να πάρει πολύ χρόνο και υπολογιστική δύναμη. Οι wrapper methods μπορούν να αναλυθούν περεταίρω ως greedy και non greedy με παραδείγματα τα simulated annealing και genetic algorithms. Γνωστοί μέθοδοι είναι οι Forward feature selection, Backwards feature selection και Recursive Feature elimination.

Η τρίτη κατηγορία των supervised feature selection methods είναι τα Embedded methods. Λέγονται Embedded διότι ενσωματώνουν την επιλογή των χαρακτηριστικών μέσα στο ίδιο το μοντέλο ταξινόμησης. Πρακτικά είναι μια μέση λύση της χρήσης Filters και Wrappers διότι είναι πιο γρήγορο και λιγότερο υπολογιστικά δαπανηρό από τα Wrappers αλλά ταυτόχρονα κρατάει υπ' όψιν τις αλληλεπιδράσεις των μεταβλητών σε αντίθεση με τα Filter. Επομένως ουσιαστικά επιλέγοντας κάποιον αλγόριθμο για ταξινόμηση επιλέγει και απορρίπτει από μόνο του features κατά την εκπαίδευση. Γνωστοί αλγόριθμοι είναι τα Random Forest, Lasso and Ridge regression (L1, L2 regularization) και το Elastic Net που είναι ο συνδυασμός τους.

Γνωρίζοντας τη θεωρία αυτή παρουσιάζεται πλέον η μεθοδολογία για την επιλογή των χαρακτηριστικών. Όπως φαίνεται και στα Βήματα οι μέθοδοι που χρησιμοποιήθηκαν είναι οι Boruta, Recursive Feature Elimination, Correlation και Variance. Οι πρώτοι δύο είναι Supervised, Wrapper methods και οι επόμενοι μέσω του τρόπου που χρησιμοποιήθηκαν υπόκεινται στη κατηγορία των Unsupervised. Με λίγα λόγια επιλέχθηκε για αρχή ο Boruta για την αντιμετώπιση του μεγάλου όγκου μεταβλητών, στη συνέχεια ο RFE για την σύγκριση και αξιολόγηση των αποτελεσμάτων αυτών. Με την επιλογή του κατάλληλου υποσυνόλου έγινε τέλος αναζήτηση για μεταβλητές με υψηλή συσχέτιση (high correlation) και χαμηλή διακύμανση (low variance) και αφαίρεση αυτών. Παρακάτω αναλύεται η μεθοδολογία του feature selection.

1) Boruta

Για αρχή χρησιμοποιείται μια Wrapper μέθοδος που λέγεται Boruta και βασίζεται στο Random Forest. Το Boruta πρακτικά δημιουργεί ένα δεύτερο σύνολο δεδομένων το οποίο είναι σαν το πρώτο αλλά έχει τυχαία ανακατεμένα όλα τα features. Αυτά ονομάζονται χαρακτηριστικά σκιά ή shadow features. Χρησιμοποιώντας Random Forest στο σύνολο των δεδομένων συμπεραίνει τη σημαντικότητα (importance) κάθε feature και συγκρίνει αυτές των αρχικών χαρακτηριστικών με αυτές των χαρακτηριστικών σκιάς. Δημιουργεί ένα όριο κατώφλι το οποίο χαρακτηρίζεται ως η υψηλότερη σημαντικότητα των χαρακτηριστικών σκιάς. Κάθε feature που έχει σημαντικότητα υψηλότερη από το κατώφλι παίρνουν πόντο, ενώ αυτά που δεν έχουν δεν παίρνουν. Η διαδικασία αυτή επαναλαμβάνεται πολλές φορές, κάθε φορά με νέα shadow features. Στο τέλος καταλήγουν κάποια χαρακτηριστικά ως επιβεβαιωμένα σημαντικά, κάποια ως επιβεβαιωμένα μη σημαντικά και κάποια στο ενδιάμεσο, ονομαζόμενα “Tentative”. Η απόφαση αυτή βασίζεται σε μια διωνυμική κατανομή που γίνεται βάση των πόντων των χαρακτηριστικών.

Η επιλογή του Boruta αυτή θεωρήθηκε αναγκαία λόγω του μεγέθους των δεδομένων που συλλέχθηκαν εξ’ αρχής. Το αρχείο έχοντας πλήθος χαρακτηριστικών ίσο με 131.072 k-mers (1.5 GB) κατέστησε αδύνατη τη συστηματική μελέτη πολλών μεθόδων καθώς δεν ήταν δυνατή η δοκιμή τους πρακτικά στο πρόγραμμα. Εξού και η ανάγκη για την εφαρμογή του feature selection.

Το αρχικό σύνολο που θα καταλήξει το Boruta θα αναλυθεί περαιτέρω χρησιμοποιώντας ένα δεύτερο, μικρότερης ισχύος, τεστ με σκοπό να αντιμετωπιστούν τα Tentatives. Έτσι το αποτέλεσμα θα έχει 2 σύνολα διαφορετικού μεγέθους.

2) ***Recursive Feature Elimination***

Τα δεδομένα αυτά στη συνέχεια τα αναλύθηκαν χρησιμοποιώντας μια επιπλέον Wrapper μέθοδο, την Recursive Feature Elimination ή RFE. Η μέθοδος RFE χρησιμοποιεί κάποιο αλγόριθμο ταξινόμησης, που ορίζεται από τον χρήστη και φτιάχνει ένα μοντέλο. Υπολογίζει για κάθε χαρακτηριστικό τη σημαντικότητά του βάση το μέτρο υπολογισμού σημαντικότητας του αλγορίθμου ταξινόμησης που του ορίστηκε. Κρατώντας ένα υποσύνολο των πιο σημαντικών χαρακτηριστικών επαναλαμβάνει την δημιουργία του μοντέλου και την κατάταξη της σημαντικότητας των χαρακτηριστικών αυτών. Η διαδικασία επαναλαμβάνεται μέχρι να φτάσει σε ένα συγκεκριμένο αριθμό χαρακτηριστικών. Τέλος συγκρίνονται τα υποσύνολα που δημιουργήθηκαν και επιλέγεται το υποσύνολο με τα καλύτερα στατιστικά επίδοσης.

Η μέθοδος θα συγκρίνει τα αποτελέσματα της μεθόδου Boruta με σκοπό την επιλογή του καλύτερου συνόλου και θα τα συγκρίνει με τυχαία υποσύνολα για την επαλήθευση της ακρίβειας των αποτελεσμάτων των συνόλων.

3) ***Correlation - Variance***

Ως τελευταίο βήμα στην προετοιμασία των δεδομένων θεωρήθηκε σημαντική η μελέτη των χαρακτηριστικών καθαυτών και πιο συγκεκριμένα η διακύμανσή τους (variance) και τη γραμμική συσχέτιση μεταξύ τους (correlation).

Σε αντίθεση με τη συσχέτιση στις μεθόδους Filter, εδώ δεν γίνεται έλεγχος για τις αλληλεπιδράσεις μεταβλητών και μεταβλητής στόχου, όπου απορρίπτονται τα χαρακτηριστικά που έχουν μικρή συσχέτιση με τον στόχο. Στη προκειμένη περίπτωση γίνεται έλεγχος για τη συσχέτιση των χαρακτηριστικών μεταξύ τους, δηλαδή k-mer προς k-mer, όπου απορρίπτονται αυτά που έχουν μεγάλη συσχέτιση μεταξύ τους. Η μέθοδος αυτή ανήκει στη κατηγορία των μη επιβλεπόμενων μεθόδων (unsupervised).

Στην ανάλυση αυτή δημιουργούνται 3 σύνολα. Το πρώτο είναι τα αποτελέσματα της ανάλυσης της συσχέτισης, το δεύτερο τα αποτελέσματα της ανάλυσης της διακύμανσης και το τρίτο είναι ο συνδυασμός. Για το πρώτο, δημιουργείται ένας πίνακας συσχετίσεων βάση του οποίου επιλέγονται και απορρίπτονται μεταβλητές. Το όριο κατώφλι ορίζεται το 0,9 όπου κάθε k-mer με τιμή συσχέτισης μεγαλύτερη από αυτό θα απορρίπτεται ως redundant. Για το δεύτερο σύνολο μελετώνται τα

κμερς, ως προς τη διακύμανση τους ανά δείγμα. Ελέγχονται για μηδενική διακύμανση και για διακύμανση μικρότερη από 0.015. Το τρίτο σύνολο δημιουργείται με η εφαρμογή του ελέγχου της διακύμανσης στα αποτελέσματα της ανάλυσης της συσχέτισης. Αυτά τα 3 σύνολα συν το αρχικό συγκρίνονται μεταξύ τους χρησιμοποιώντας ξανά τη μέθοδο RFE με σκοπό την επιλογή του καλύτερου.

Εκ των αποτελεσμάτων αποφασίστηκε να χρησιμοποιηθούν δύο σύνολα για τη πορεία της ανάλυσης, το σύνολο των 470, το οποίο είναι αυτό που έκρινε σημαντικό η μέθοδος Boruta, και το σύνολο των 284, όπου είναι το ίδιο σύνολο αφού επεξεργάστηκε βάση συσχέτισης και διακύμανσης. Σε κάθε από τα μοντέλα που θα χρησιμοποιηθούν παρακάτω θα γίνεται η ανάλυση και στα δύο σύνολα και στο τέλος θα συγκριθούν και θα συνδυαστούν τα αποτελέσματά τους για την εύρεση των σημαντικών k-mer. Επομένως, τα δεδομένα, που στην αρχή είχαν μέγεθος 131.073 features, και 680 δείγματα, κατέληξαν στα μεγέθη των 470 και 284 features.

3.2.2 Dimensionality reduction

Το Dimensionality reduction ή στα ελληνικά, μείωση διαστάσεων είναι η διαδικασία προβολής των δεδομένων σε δύο διαστάσεις. Σε δύο διαστάσεις αναπαριστώνται κανονικά τα δεδομένα που έχουν δύο χαρακτηριστικά. Αυτά που έχουν τρία αναπαριστώνται στο χώρο και ότι είναι παραπάνω από αυτό δεν μπορεί να αναπαρασταθεί. Με τις μεθόδους της μείωσης των διαστάσεων γίνονται υπολογισμοί και βάση των αποτελεσμάτων διατυπώνονται στο επίπεδο. Αυτό στη περίπτωση της ταξινόμησης μπορεί να αναδείξει τις αλληλεπιδράσεις και συσχετίσεις των επιμέρους κατηγοριών.

Η αναπαράσταση μέσω μείωσης των διαστάσεων εφαρμόζεται σε διάφορα μέρη της ανάλυσης και συγκεκριμένα στην αρχή μέσω των μεθόδων PCA και LDA και κατά την ταξινόμηση για σύγκριση των μεθόδων με χρήση της MDA.

3.3 Classification

3.3.0 Θεωρία

Αρχικά, όπως και στο feature selection, έτσι και εδώ υπάρχουν κατηγορίες οι οποίες αντιστοιχούν στον τύπο του προβλήματος. Τρεις βασικές κατηγορίες εκπαίδευσης είναι η επιβλεπόμενη (supervised), η μη επιβλεπόμενη (unsupervised) και η ενισχυμένη (reinforcement). Πρακτικά η επιβλεπόμενη εκπαιδεύει μοντέλα δίνοντας τους παραδείγματα, δίνει τιμές εισόδου και την αντίστοιχη μεταβλητή εξόδου/ πρόβλεψης ενώ η μη επιβλεπόμενη δεν χρησιμοποιεί μεταβλητές στόχους και μελετάει τα ίδια τα δεδομένα μεταξύ τους. Τέλος η ενισχυμένη ψάχνει τη καλύτερη πορεία δράσης μέσω δοκιμής και σφάλματος (trial & error) για να πετύχει το «μεγαλύτερο κέρδος». Είδη μη επιβλεπόμενης εκπαίδευσης είναι η ομαδοποίηση (clustering), εκτίμηση πυκνότητας (density estimation) και απεικόνιση (visualization) και είδη ενισχυτικής είναι η ανάθεση ευσήμων (credit assignment) και τα εξερεύνηση και εκμετάλλευση. Η ταξινόμηση (classification) ανήκει στην επιβλεπόμενη εκπαίδευση μαζί με τη παλινδρόμηση (regression).

Η ταξινόμηση έχει σκοπό να ταξινομήσει δεδομένα σε μία ή περισσότερες κατηγορίες. Υπάρχουν 4 κατηγορίες ταξινόμησης, η δυαδική (binary), η πολλαπλών κλάσεων (multiclass), η πολλαπλών ετικετών (multi-label) και η μη ισορροπημένη (imbalanced). Η δυαδική ταξινομεί δεδομένα μεταξύ δυο αμοιβαία αποκλειόμενες κατηγορίες. Αντίστοιχα η παρουσία περισσότερων από δύο κατηγοριών είναι η ταξινόμηση πολλών κλάσεων. Η ταξινόμηση πολλαπλών ετικετών είναι μια άλλη εκδοχή κατά την οποία οι κατηγορίες δεν είναι αμοιβαία αποκλειόμενες δηλαδή η ταξινόμηση σε μία κατηγορία δεν απορρίπτει την ταξινόμηση σε άλλη. Τέλος η μη ισορροπημένη είναι όταν τα δείγματα δεν είναι ισάξια κατανεμημένα σε κάθε κατηγορία προκαλώντας το μοντέλο να είναι προκατειλημμένο ως προς την συχνότερη κατηγορία.

Βάση των δεδομένων, η παρουσία των 8 ξενιστών αποκλείει την χρήση δυαδικής ταξινόμησης. Η διαφορά μεταξύ των multi-label και multi-class είναι μικρή και αν σκοπός ήταν απλά η πρόβλεψη του ξενιστή, τότε και τα δυο μπορούν να βγάλουν αξιόπιστα αποτελέσματα. Όμως στην εργασία αυτή η ταξινόμηση γίνεται με σκοπό να εντοπιστούν ποια k-mers κάνουν καλύτερα την διαφοροποίηση. Δεν μπορεί ένα δείγμα να ανήκει σε περισσότερους από έναν ξενιστές και η ταξινόμηση χωρίς

αμοιβαία απόκλιση των ξενιστών μπορεί να βγάλει διαφορετικά κ-μερή από αυτά που αναζητούνται. Επομένως επιλέγεται η multi-class ταξινόμηση.

Τέλος ένα πρόβλημα ταξινόμησης μπορεί να χρειάζεται έλεγχο ισορροπίας των δεδομένων παρά των προηγούμενων κατηγοριών. Για τη μη ισορροπημένη ταξινόμηση υπάρχουν δύο εκδοχές. Η μία είναι τα δεδομένα να έχουν ελαφριά ανισορροπία όπου η ταξινόμηση κλίνει ως προς τη μεγαλύτερη κατηγορία δεδομένων και η άλλη είναι η σοβαρή ανισορροπία όπου η διαφορά είναι τόσο μεγάλη που οι περιπτώσεις των μικρών κατηγοριών είναι τόσο λίγες που μπορεί να αντιμετωπιστούν και ως θόρυβο από το μοντέλο ταξινόμησης. Σε αυτές τις περιπτώσεις υπάρχουν τεχνικές εξισορρόπησης των δεδομένων. Είτε υπάρχει μία κατηγορία που είναι αρκετά μεγαλύτερη από τις υπόλοιπες είτε μία που είναι αρκετά μικρότερη γίνεται με συγκεκριμένες τεχνικές να μειώσεις ή να αυξήσεις το μέγεθος τους με τυχαία αφαίρεση ή προσθήκη δειγμάτων αντίστοιχα. Ή μπορούν να εφαρμοστούν αλγόριθμοι που λαμβάνουν υπόψιν το κόστος εσφαλμένης κατηγοριοποίησης και αποφεύγεται έτσι να αγνοηθούν οι μικρές κατηγορίες.

3.3.1 Multi-class Random Forest

Υπάρχουν πολλοί αλγόριθμοι που μπορούν να κάνουν ταξινόμηση πολλών κλάσεων. Μεταξύ αυτών είναι τα k-Nearest Neighbors, Decision Trees, Naive Bayes, Random Forest. Για αρχή δοκιμάζεται ο αλγόριθμος Random Forest. Ο Random Forest είναι της κατηγορίας Ensemble, το οποίο είναι ο συνδυασμός μεθόδων ταξινόμησης. Υπάρχουν 3 είδη τεχνικών Ensemble, το Bagging, το Boosting και το Stacking. Ο Random Forest συνδυάζει δέντρα απόφασης (decision trees) χρησιμοποιώντας τη τεχνική Bagging.

Τα δέντρα απόφασης χρησιμοποιούν τα χαρακτηριστικά των δεδομένων ως ρίζα (root node) και κλαδιά (internal nodes) του δέντρου για να πάρει αποφάσεις οι οποίες οδηγούν σε άλλα κλαδιά ή στα φύλλα (leaf nodes) τα οποία δίνουν τη τελική απόφαση η οποία είναι η κατηγοριοποίηση της μεταβλητής στόχου. Έτσι κάθε νέο δείγμα, ακολουθώντας τα μονοπάτια του δέντρου, μπορεί να κατηγοριοποιηθεί. Για να κριθεί ποιο χαρακτηριστικό θα πάρει τη θέση κάθε κλαδιού χρησιμοποιείται η τιμή «καθαρότητάς» τους το οποίο συνήθως υπολογίζεται χρησιμοποιώντας τη μέθοδο Gini Impurity. Όσο πιο καθαρό είναι ένα χαρακτηριστικό, δηλαδή όσο πιο καλά πετυχαίνει τη πρόβλεψη, τόσο μικρότερη τιμή θα πάρει και τόσο πιο πιθανό είναι να επιλεχθεί έναντι άλλων.

Ο Random forest χρησιμοποιεί τη τεχνική των decision trees. Για αρχή δημιουργεί ένα σύνολο δεδομένων βασισμένο στο αρχικό που ονομάζεται Bootstrap dataset. Αυτό έχει το ίδιο μέγεθος με το αρχικό και δομείται από τυχαία δείγματα του αρχικού συνόλου, όπου κάθε δείγμα μπορεί να καταχωρηθεί πάνω από μια φορές. Μετά δημιουργεί ένα δέντρο απόφασης βασισμένο στο Bootstrap dataset. Για κάθε node δημιουργείται ένα τυχαίο υποσύνολο των αρχικών μεταβλητών και η καλύτερη από αυτές ορίζεται στο node. Η παρουσία μιας μεταβλητής σε ένα node δεν αποκλείει τη παρουσία της σε άλλο. Η διαδικασία αυτή επαναλαμβάνεται εκατοντάδες φορές.

Ο Random Forest χωρίζει με αυτόν τον τρόπο τα δεδομένα σε train και test set και δεν χρειάζεται cross-validation. Τα δεδομένα που χρησιμοποιήθηκαν για να δημιουργήσουν τα δέντρα απόφασης, δηλαδή τα Bootstrap datasets, χρησιμοποίησαν κάποια από τα δείγματα τα οποία πρακτικά είναι το train set του κάθε δέντρου. Τα δείγματα που μένουν χρησιμοποιούνται για τον έλεγχο του κάθε δέντρου που δεν τα χρησιμοποίησε άρα δουλεύουν ως test set. Αυτά ονομάζονται Out-of-Bag (oob) dataset. Η τελική κατηγοριοποίηση κάθε oob δείγματος είναι η συχνότερη απόφαση

από το σύνολο των αποφάσεων των δέντρων στα οποία δοκιμάστηκε. Μέσω κατηγοριοποίησης όλων των oob δειγμάτων μπορεί να βγει η συχνότητα σωστής και λάθος κατηγοριοποίησης τους, δηλαδή το πόσο ακριβές είναι το Random Forest. Αυτό που μετράει ο Random Forest είναι το ποσοστό λάθος κατηγοριοποίησης των oob δειγμάτων το οποίο ονομάζεται Out-of-Bag error και η ακρίβεια εμφανίζεται με τη μορφή του Out-of-Bag error rate. Τέλος, η διαδικασία δημιουργίας Bootstrap δεδομένων και η λήψη απόφασης βάση του συνόλου (aggregate) των αποφάσεων ονομάζεται Bagging.

Έτσι δημιουργείται ο Random Forest. Ένα τελευταίο βήμα στη διαδικασία είναι η βελτιστοποίηση των παραμέτρων. Οι παράμετροι που μπορούν να αλλάξουν στον Random Forest είναι ο αριθμός των δέντρων που θα δημιουργηθούν (ntree) και ο αριθμός των μεταβλητών που παίρνει σε κάθε node, ή αλλιώς split, στη δημιουργία των δέντρων (mtry). Από προεπιλογή τα δεδομένα αυτά είναι τα 500 δέντρα και ως αριθμό των μεταβλητών είναι η τετραγωνική ρίζα του συνόλου των μεταβλητών. Θα μελετηθούν τρεις ομάδες δέντρων για το καλύτερο ntree, τα 500, 1000 και 1500 και για το καλύτερο mtry δοκιμάζεται ίσος αριθμός μονάδων μεγαλύτερους και μικρότερους από το προκαθορισμένο. Η διαδικασία αυτή γίνεται δύο φορές, μία για κάθε σύνολο δεδομένων που μελετάται. Η μόνη διαφορά μεταξύ τους είναι ο αριθμός των mtry που δοκιμάζονται, συγκεκριμένα στο σύνολο των 470 δοκιμάζονται από το 1 μέχρι το 42 με προκαθορισμένη το 21 και στο σύνολο των 284 δοκιμάζονται από το 1 μέχρι το 32 με προκαθορισμένη το 16.

Η διαδικασία δημιουργίας του μοντέλου αναπαρίσταται με confusion matrix και διαγράμματα και MDS διάγραμμα μείωσης διαστάσεων στα αποτελέσματα. Τέλος, αποθηκεύονται τα 50 πιο σημαντικά k-mers για τα επόμενα βήματα της μεθοδολογίας.

3.3.2 One vs. All Random Forest

Υπάρχουν αλγόριθμοι που ενώ δεν είναι φτιαγμένοι για πολλαπλή ταξινόμηση, μπορούν να εφαρμοστούν χρησιμοποιώντας μία από τις παρακάτω τεχνικές. Αυτές είναι η «ένα ενάντια ενός» (One vs. One) και η «ένα ενάντια όλων» (One vs. All). Αυτές οι τεχνικές ουσιαστικά μετατρέπουν το πρόβλημα από πολλαπλών τάξεων σε δυαδικό παίρνοντας είτε ζευγάρια κατηγοριών κάθε φορά ή ένα ενάντια όλων των υπολοίπων αντίστοιχα.

Σκοπός της εργασίας είναι η εύρεση των k-mers που θεωρήθηκαν πως είναι πιο σημαντικά στη ταξινόμηση με τη λογική πως αυτά ίσως χαρακτηρίζουν την κατηγοριοποίηση των οργανισμών. Με τον Random Forest στη προηγούμενη μοντελοποίηση αποθηκεύτηκαν αυτά που ήταν πιο σημαντικά. Χρησιμοποιώντας όμως την τεχνική “One vs All” μπορεί να αναγνωριστεί συγκεκριμένα για κάθε οργανισμό ποια k-mers συμβάλλουν στην ταξινόμηση τους ενάντια σε όλους τους άλλους. Αυτό παρουσιάσει τα κμερς που έχουν πιο πολλή σημασία για τον κάθε οργανισμό και δίνει περισσότερη πληροφορία από το γενικό Random Forest.

Για τη δημιουργία του OvA Random Forest ακολουθείται η ίδια διαδικασία με πριν. Δημιουργία μοντέλου με τις προκαθορισμένες ρυθμίσεις, αναζήτηση των εν δυνάμει βέλτιστων παραμέτρων και εκπαίδευση τελικού μοντέλου με τη χρήση αυτών. Η διαδικασία αυτή επαναλαμβάνεται τόσες φορές όσοι είναι οι οργανισμοί και ως χαρακτηριστικό στόχος δεν είναι όλοι οι οργανισμοί αλλά ο ένας οργανισμός που μελετάται κάθε φορά ενάντια σε μία τιμή που περιέχει όλους τους υπόλοιπους. Οι παράμετροι που δοκιμάζονται είναι για αρχή, ο αριθμός των μεταβλητών που χρησιμοποιούνται σε κάθε node split, ή αλλιώς το mtry. Για το σύνολο των 470 μεταβλητών δοκιμάζονται οι αριθμοί από το 1 μέχρι το 42, για ένα ισάξιο αριθμό δοκιμών πάνω και κάτω του προκαθορισμένου που ήταν το 21. Για το σύνολο δεδομένων των 284, που έχει ως προκαθορισμένη τιμή το 16, δοκιμάζονται οι αριθμοί από το 1 μέχρι το 32. Η δεύτερη παράμετρος είναι ο αριθμός των δέντρων και για κάθε μοντέλο θα δοκιμαστούν πρώτα τα 500, 1000 και 1500 και στα δύο σύνολα. Αν θεωρηθεί πως τα 1500 δεν προσφέρουν σημαντικά καλύτερα αποτελέσματα, τότε θα παραλειφθούν από την ανάλυση για λόγους ταχύτητας.

Τέλος δημιουργείται αναπαράσταση των δειγμάτων μέσω της μεθόδου μείωσης διαστάσεων MDS για κάθε οργανισμό σε σύγκριση με τους υπόλοιπους.

3.4 Variable importance

Με το τέλος της μοντελοποίησης συλλέγονται τα πιο σημαντικά k-mers που αποφασίστηκαν σε κάθε μέθοδο. Αυτά αναπαρίστανται με διαγράμματα bar-plot για κάθε μέθοδο και οργανισμό. Γίνεται σύγκριση μεταξύ αυτών των συνόλων κμερς με σκοπό την αναζήτηση των σημαντικότερων. Αρχικά, συγκρίνονται τα k-mers της μεθόδου One vs All των 470 με των 284. Ελέγχεται κάθε οργανισμός του ενός συνόλου με τον αντίστοιχο οργανισμό του άλλου για παρουσία κοινών κμερς μεταξύ τους. Τα κμερς αυτά επιλέγονται. Στη συνέχεια τα κμερς που έμειναν σε κάθε οργανισμό συγκρίνονται μεταξύ τους. Αυτό γίνεται γιατί σκοπός είναι η αναζήτηση μοναδικών k-mer για κάθε οργανισμό, επομένως τα κοινά μεταξύ οργανισμού – οργανισμού αφαιρούνται. Τα κμερς που μένουν θα χρησιμοποιηθούν στη συνέχεια της ανάλυσης επιλέγοντας τα 3 πιο σημαντικά. Έμφαση θα δοθεί στα κμερς αυτά και στα κμερς που είναι κοινά μεταξύ των συνόλων της μεθόδου Random Forest. Τέλος για την αναπαράστασή τους δημιουργούνται πίνακες Venn.

3.5 BLAST

Τα τελικά 3 κμερς κάθε οργανισμού, που βρέθηκαν κοινά μεταξύ των συνόλων δεδομένων που μελετήθηκαν και των επιμέρους μεθόδων που χρησιμοποιήθηκαν είναι αυτά που αναλύονται χρήση του εργαλείου BLAST. Επιλέγεται «nucleotide BLAST» όπου ορίζεται ως όρισμα στο «Enter Query Sequence» το k-mer το οποίο εξετάζεται κάθε φορά. Στο «Choose Search Set» επιλέγεται στο «Database» το «RefSeq Representative genomes» και στο «Organism» ορίζεται το όνομα του βακτηρίου, «Klebsiella pneumoniae (taxid: 573)». Επιλέγεται στο «Program Selection» το «megablast (highly similar sequences)» και στα «Algorithm parameters» ορίζεται το «word size» ίσο με 16. Τέλος παραμένει επιλεγμένο το «Automatically adjust parameters for short input sequences» ώστε να κάνει ότι αλλαγές χρειάζεται για να τρέξει το εργαλείο.

Στα αποτελέσματα γίνεται αναζήτηση των γονιδίων που ανήκει το k-mer. Λαμβάνονται υπόψη μόνο οι καταχωρήσεις που έχουν «Identities» 9/9(100%) και σκορ 18.3 bits. Στην επιλογή «Graphics» κάθε καταχώρησης βρίσκονται οι πληροφορίες που θα συλλεχθούν για κάθε k-mer. Αυτές είναι η τοποθεσία του k-mer, δηλαδή αν βρίσκεται σε πλασμίδιο ή στο χρωμόσωμα η συχνότητα συνάντησής του στο γονιδίωμα και σε ποια γονίδια ανήκει.

Το βακτήριο Klebsiella pneumoniae αποτελείται από ένα χρωμόσωμα μεγέθους 5.3 Mb (όπου 1 Mb είναι 1.000.000 βάσεις) , 3 πολυανθεκτικά στα φάρμακα πλασμίδια με μέγεθος γύρω στα 110 kb (όπου 1kb είναι 1.000 βάσεις) και 3 μικρά πλασμίδια μεγέθους περίπου 3 kb. [17] Αναλυτικότερα τα πλασμίδια είναι κυκλικά, όπως και το χρωμόσωμα, και ονομάζονται με τη σειρά pKPHS1, pKPHS2, pKPHS3, pKPHS4, pKPHS5, pKPHS6. [18]

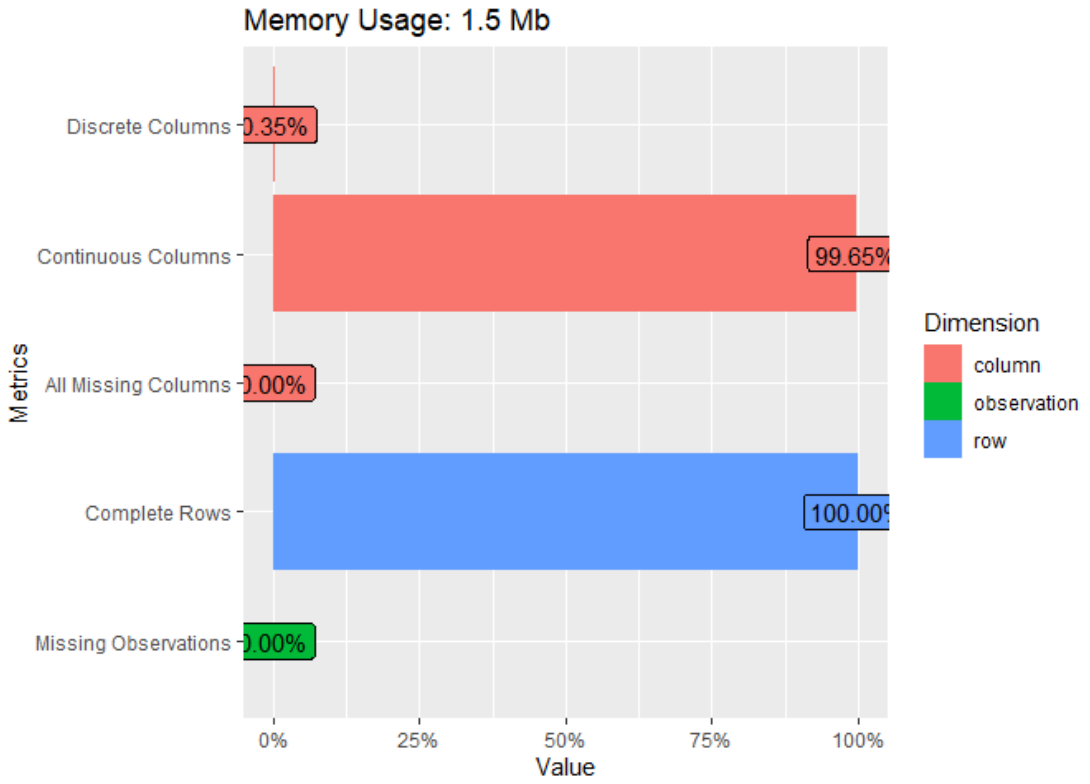
4. Αποτελέσματα

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα κάθε βήματος της μεθοδολογίας.

4.1 Εξερεύνηση δεδομένων

Για αρχή γίνεται μια εξερεύνηση των δεδομένων ή αλλιώς data exploration. Αυτό γίνεται για να κατανοηθούν καλύτερα τα δεδομένα και να εφαρμοστούν πιο σωστά οι μέθοδοι και οι τεχνικές.

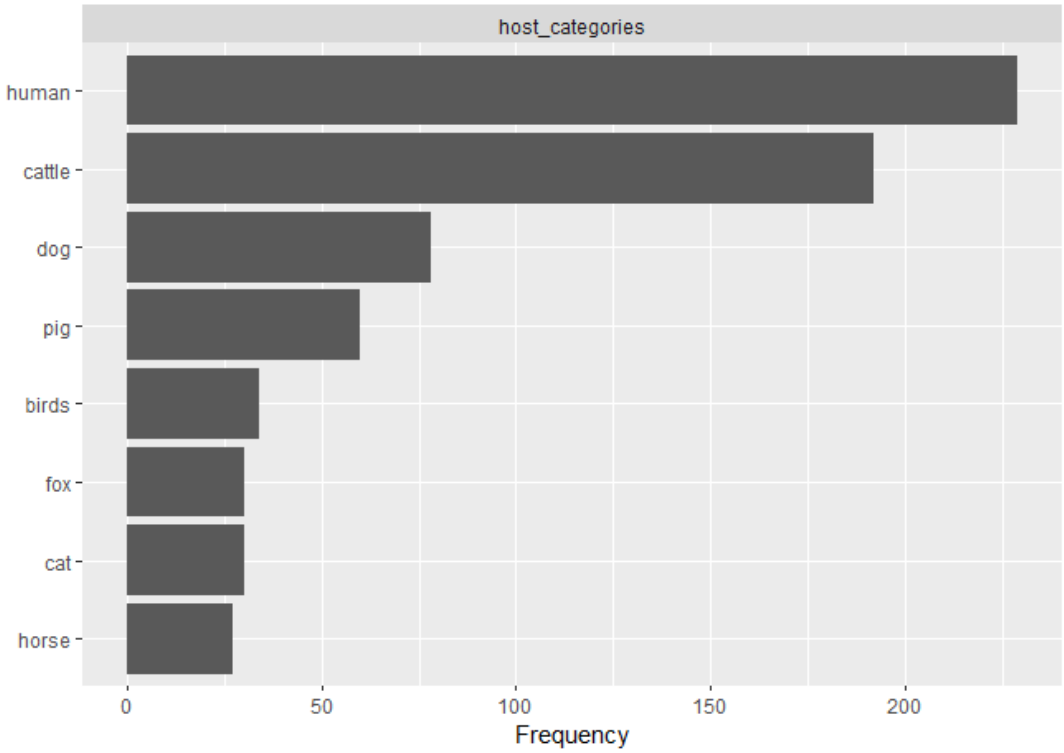
Πρώτα ελέγχεται αν υπάρχουν κενές παρατηρήσεις ή αλλιώς NA's. Το σύνολο των δεδομένων δεν έχει δείγματα με έλλειψη τιμών και αυτό φαίνεται παρακάτω στο διάγραμμα.



Εικόνα 1 Ποιοτική ανάλυση δεδομένων

Το 0.35% των διακριτών στηλών είναι η μία στήλη των ξενιστών οργανισμών και οι υπόλοιπες είναι συνεχείς στήλες, με την κανονικοποιημένη συχνότητα των 9-μερών. Οι γραμμές που είναι τα δείγματα είναι 100% και αντίστοιχα δεν υπάρχουν «χαμένες» στήλες ή τιμές στα δείγματα.

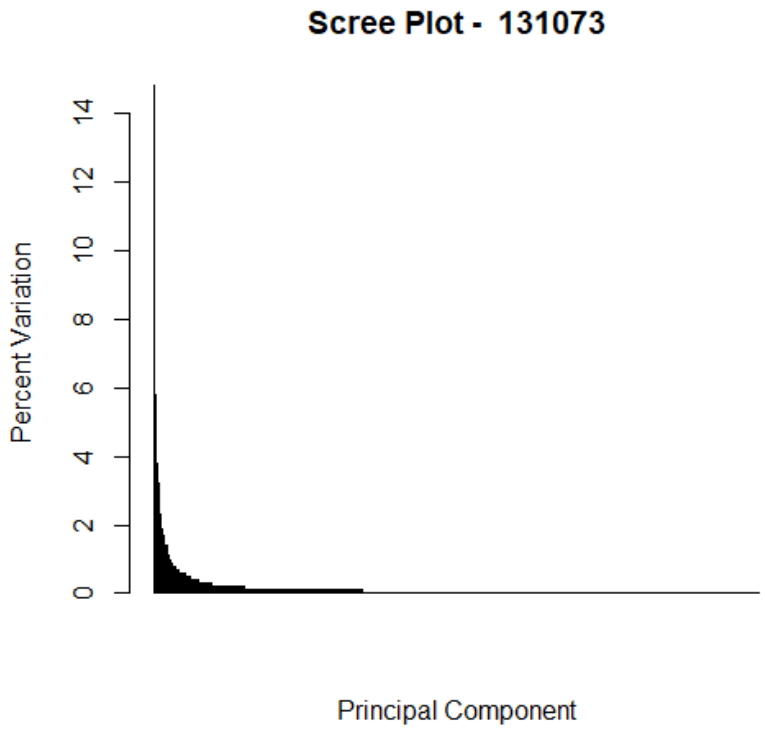
Συνεχίζοντας ελέγχονται τα δεδομένα για την ισορροπία των δειγμάτων των οργανισμών. Η αναλογία τους φαίνεται στο παρακάτω διάγραμμα.



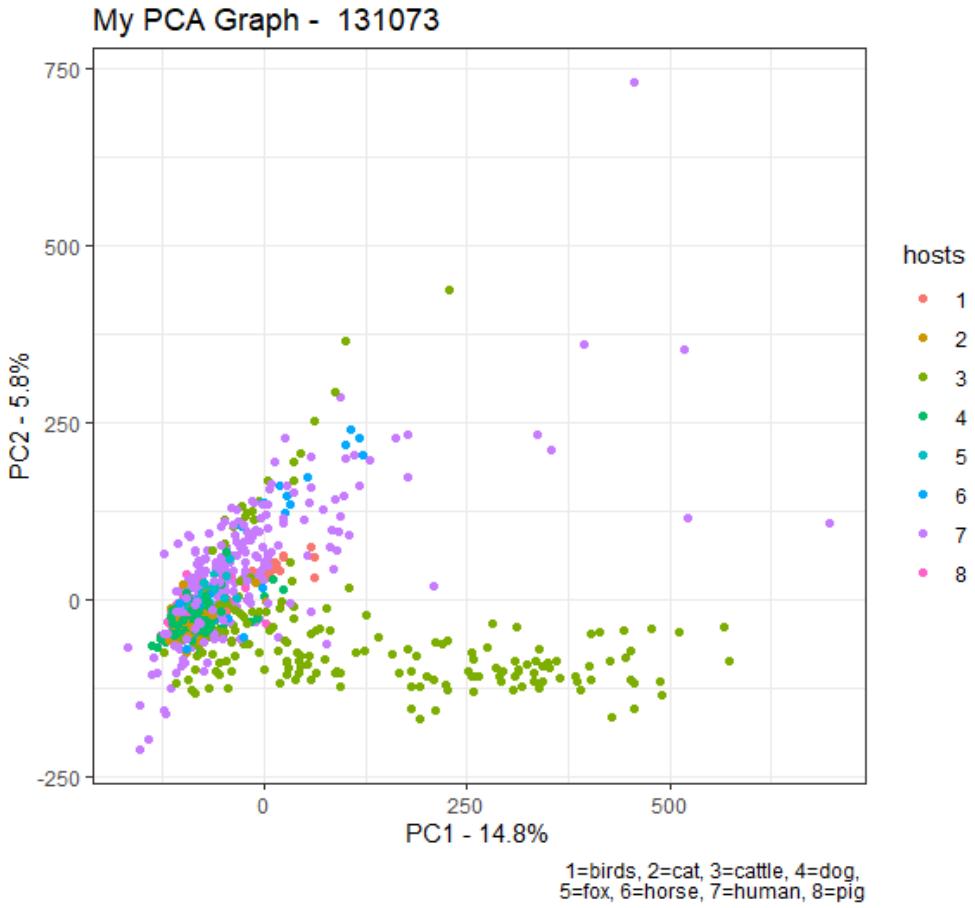
Εικόνα 2 Συχνότητα εμφάνισης οργανισμών στο σύνολο δεδομένων

Φαίνεται να έχουν ανισορροπία δεδομένων. Η παρουσία δειγμάτων για τον άνθρωπο και τα βοοειδή είναι πολύ μεγαλύτερη από τα υπόλοιπα με τα σκυλιά να είναι στη τρίτη θέση αλλά τουλάχιστον 100 δείγματα πίσω. Όμως βρίσκοντας τις αναλογίες τους, η διαφορά τους είναι μέσα στις δεκάδες (3/10, 1/10, 4/10…) και μπορούν να θεωρηθούν ως ελαφριά ανισορροπία. Σε αυτή τη περίπτωση είναι αποδεκτή η μη χρήση τεχνικών εξισορρόπησης των δεδομένων. Στα επόμενα βήματα της ανάλυσης η μικρή αυτή λοξότητα δεν θα επηρεάσει σε σημαντικό βαθμό τα αποτελέσματα.

Τέλος αναπαριστάται το σύνολο δεδομένων με PCA και LDA.



Εικόνα 3 Αναπαράσταση των Principal Components – αρχικά δεδομένα

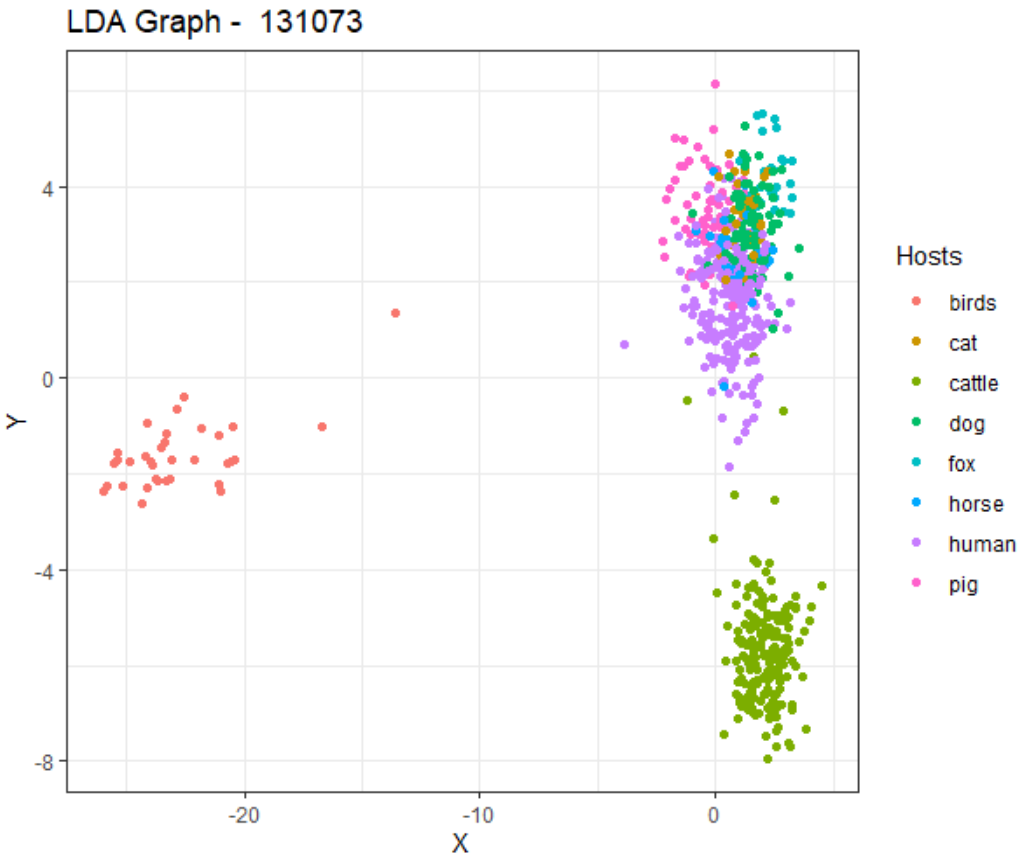


Εικόνα 4 Αναπαράσταση δεδομένων με PCA - αρχικά δεδομένα

```
> top_kmers ## show the names of the top 100 kmers
  [1] "CCGCTGGTC" "ACGTCGCGC" "CGATGCGCC" "AGATCGGCG" "CGCCGAGGA" "GATCGTCGA" "CGCTGGACG"
  [8] "ACGCCGATC" "AGCGCGTCG" "GATCGGCGA" "CGTCGCGCC" "GCGCGCCAA" "CCATCGCGC" "CCTTCGGCG"
 [15] "CTGCAGGTC" "CCTGGGCGC" "CGCGGCGTA" "ACCGGCGTG" "CCAGGCGTC" "GATGGCGCC" "CCGACGCGC"
 [22] "GACGATCGC" "GCGCGGTGA" "CGCGGATCA" "CACCAGGTC" "AGGGTGCCG" "CCGCGACGC" "GCCGCGCGA"
 [29] "CCAGCGACG" "CATCGCGCG" "ACGGCGCGC" "CCCAGCTGC" "GCCGGTCAC" "CGCGCGCCA" "CATCGCGGC"
 [36] "CGACCATCG" "GCGCTGGAC" "CTTCGCGCA" "AACGGCACC" "GCGCCTGGA" "CCACCTGGC" "ACCGCGCGC"
 [43] "ACCGCATCG" "CGTCGCCAC" "GGCGATCAC" "AGGCGATCG" "GCTCGGCGA" "CCGCGCGCA" "CGATCGTCA"
 [50] "ACGCGCCGC" "CTCGACGGC" "TGCCGCCCA" "GCGCGACGA" "GGTCAGCAC" "ACCGACAGC" "CAGCTGGGC"
 [57] "CGCGCGGCA" "ACGGCCTGC" "CGCATCGAC" "AACGGCGCG" "CGGCGTGGC" "CGCCGAGCA" "GCGCGTCGA"
 [64] "CGCCTGATC" "GCCGATCAC" "AGATGGCGC" "CGCGAGCGC" "CGCGCTGCA" "CGCGACGCG" "AGCGTGCCG"
 [71] "CGGTGGTCA" "ACTGCAGCG" "CAGGGTGCC" "GCGCTTCGA" "ATGCGCGCC" "CGCTGGGCC" "CTGGACGGC"
 [78] "CGCGCCGAA" "ACCCTGGCG" "CCTGACCGC" "GAGCTGCGC" "CGCCGATCA" "CGATGGGCG" "CCAGCTGGG"
 [85] "CTGGGCGTC" "GCCAGCGAC" "CTGGCCCAG" "AGGGCGCGC" "ACGACGCCG" "AGCTGGAGC" "GCTCGCCGA"
 [92] "ACGATCAGC" "CCGAGCAGC" "CGGCGTACC" "CGATGCCGC" "ATGGCCGGC" "ATCACCGCG" "GATGGGCGA"
 [99] "CTTCGGCGA" "AGCGTCGCG"
```

Εικόνα 5 Τα πιο σημαντικά κμερς βάση PCA – αρχικά δεδομένα

Και με το LDA:



Εικόνα 6 Αναπαράσταση δεδομένων με LDA - αρχικά δεδομένα

4.2 Boruta

Το Boruta έτρεξε με τις προκαθορισμένες ρυθμίσεις κάνοντας 100 επαναλήψεις και από τα 131.072 k-mer κράτησε 729, από τα οποία τα 81 θεωρήθηκαν σημαντικά και τα 648 Tentative. Τα 130343 θεωρήθηκαν μη σημαντικά.

```
> boruta_output
Boruta performed 99 iterations in 1.991866 hours.
 81 attributes confirmed important: AAAAAAAAA, AAAAAAAAG, AAAAAAAAT,
AAAAAAAGA, AAACCCCCC and 76 more;
 130343 attributes confirmed unimportant: AAAAAAACA, AAAAAAACC, AAAAAAACG,
AAAAAAACT, AAAAAAAGC and 130338 more;
 648 tentative attributes left: AAAAAAAAC, AAAAAAAGG, AAAAAAATG, AAAAAAATT,
AAAAAAGAA and 643 more;
```

Εικόνα 7 Τα αποτελέσματα της feature selection μεθόδου Boruta

Το Boruta πακέτο της R δίνει την επιλογή να αντιμετωπίσει τα Tentative χαρακτηριστικά με ένα επιπλέον, μικρότερης ισχύος, τεστ. Αυτό επιλέγει τι να κάνει τα Tentative και κράτησε 469 features. Με τα αποτελέσματα αυτά δημιουργήθηκαν δυο νέα δεδομένα, δύο πίνακες, ο ένας με τα 729+host και ο άλλος με τα 469+host.

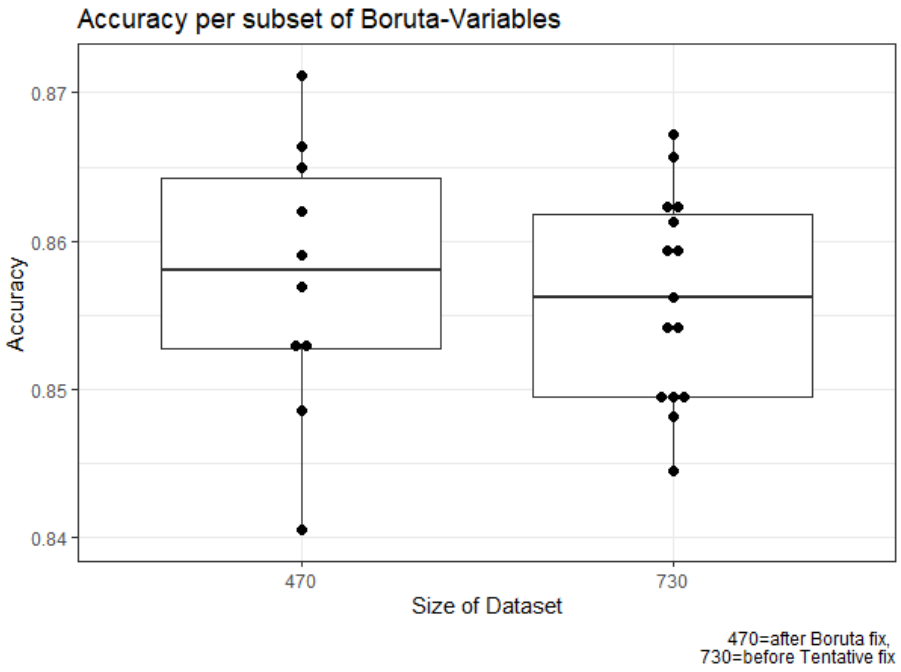
Τυπικά παρακάτω είναι τα 6 πιο σημαντικά χαρακτηριστικά που έκρινε το Boruta κατά την διαδικασία έγκρισης και απόρριψης χαρακτηριστικών.

	meanImp	decision
AATGATACG	4.489340	Confirmed
GGGGGGCAC	3.953945	Confirmed
GCCGAGACC	3.778185	Confirmed
GGGGGGAAA	3.760377	Confirmed
AAAAAAAAG	3.675143	Confirmed
GGGGGGGAA	3.636510	Confirmed

Εικόνα 8 Τα 6 πιο σημαντικά features που έκρινε το Boruta

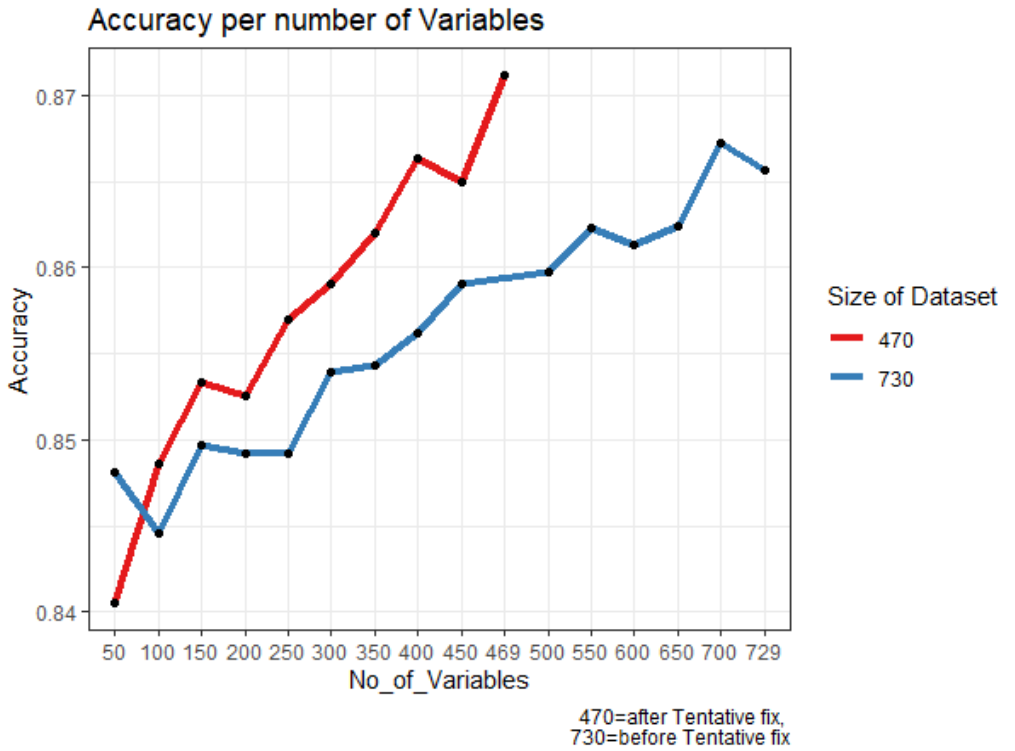
4.3 Recursive Feature Elimination

Στην RFE δίνονται για αξιολόγηση οι δύο πίνακες, τα σύνολα δεδομένων (datasets) που έβγαλε το Boruta με σκοπό την επιλογή ενός εξ αυτών. Χρησιμοποιώντας ως βάση τον αλγόριθμο Random Forest και ορίζοντας για το cross-validation 10-fold και 5 επαναλήψεις δημιουργείται το μοντέλο το οποίο εξετάζει επαναληπτικά υποσύνολα από το 50 μέχρι το μέγεθος του κάθε συνόλου. Παρακάτω βλέπουμε την ακρίβεια (Accuracy) για κάθε υποσύνολο του κάθε συνόλου δεδομένων.



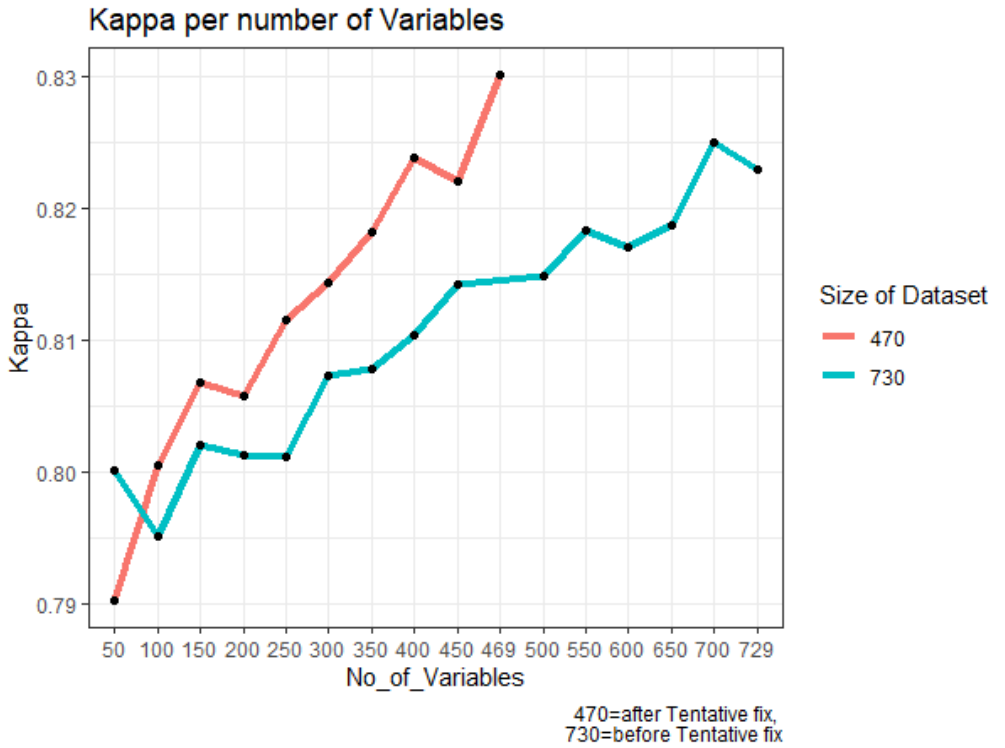
Εικόνα 9 Ακρίβεια υποσυνόλων χαρακτηριστικών για κάθε σύνολο δεδομένων του Boruta - Boxplot

Κάθε στήλη είναι τα datasets του Boruta και κάθε σημείο είναι η Ακρίβεια κάθε υποσυνόλου. Συνολικά φαίνεται πως η ακρίβεια των δύο συνόλων δεν διαφέρει πολύ, με το σύνολο των υποσυνόλων να βρίσκονται μεταξύ 0,84 μέχρι 0,87 τα οποία αυτά κάθε αυτά είναι υψηλός βαθμός ακρίβειας. Λεπτομερέστερα το σύνολο των 470 φαίνεται να έχει λίγο υψηλότερα το μέσο βαθμό ακρίβειας και κάποιο υποσύνολό του να έχει την υψηλότερη ακρίβεια όλων αλλά ταυτόχρονα κάποιο έχει τη χαμηλότερη. Το παρακάτω διάγραμμα δείχνει αναλυτικότερα την διακύμανση της ακρίβειας των υποσυνόλων κάθε dataset.



Εικόνα 10 Ακρίβεια υποσυνόλων χαρακτηριστικών για κάθε σύνολο δεδομένων του Boruta - line plot

Όμοια και το Kappa (ή Cohen's kappa) :

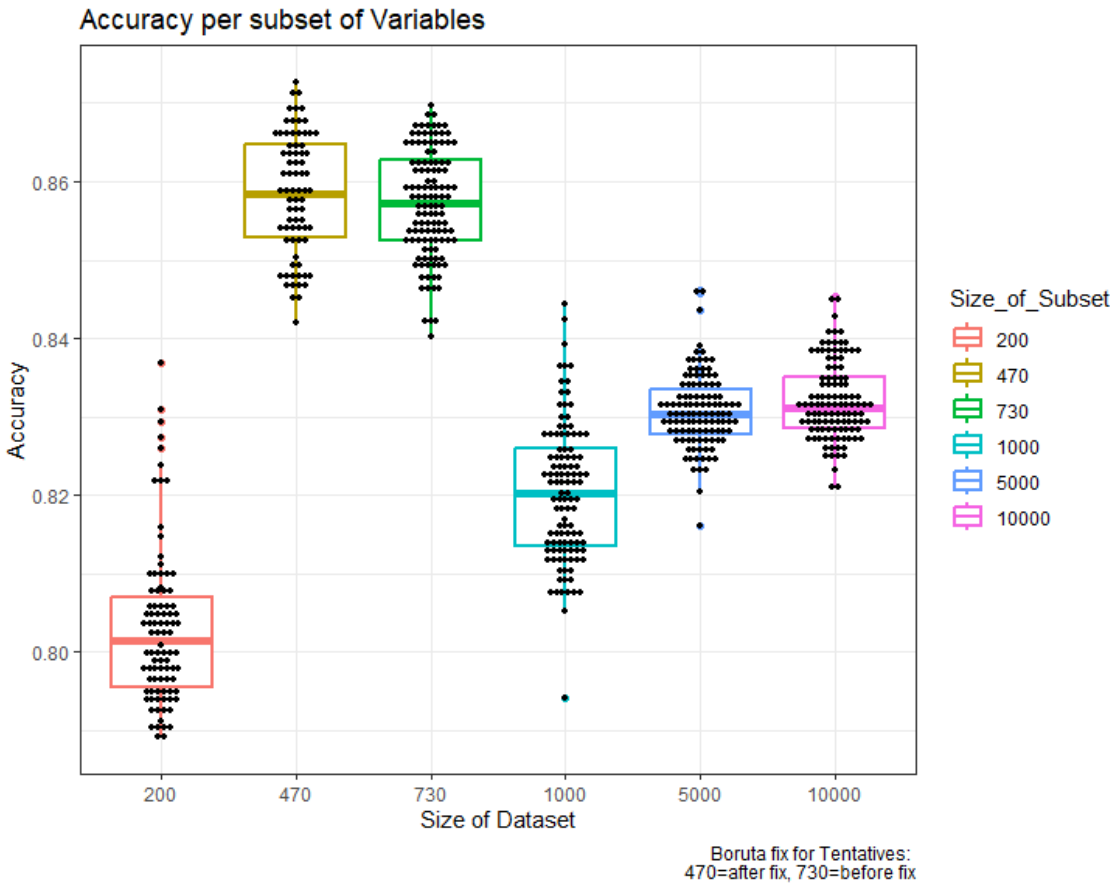


Εικόνα 11 Kappa υποσυνόλων χαρακτηριστικών για κάθε σύνολο δεδομένων του Boruta - line plot

Φαίνεται καλύτερα πως η ακρίβεια σχεδόν κάθε υποσυνόλου του 470 είναι καλύτερη από των 730. Η μέγιστη τιμή βρίσκεται στις 469 μεταβλητές και τα σύνολα

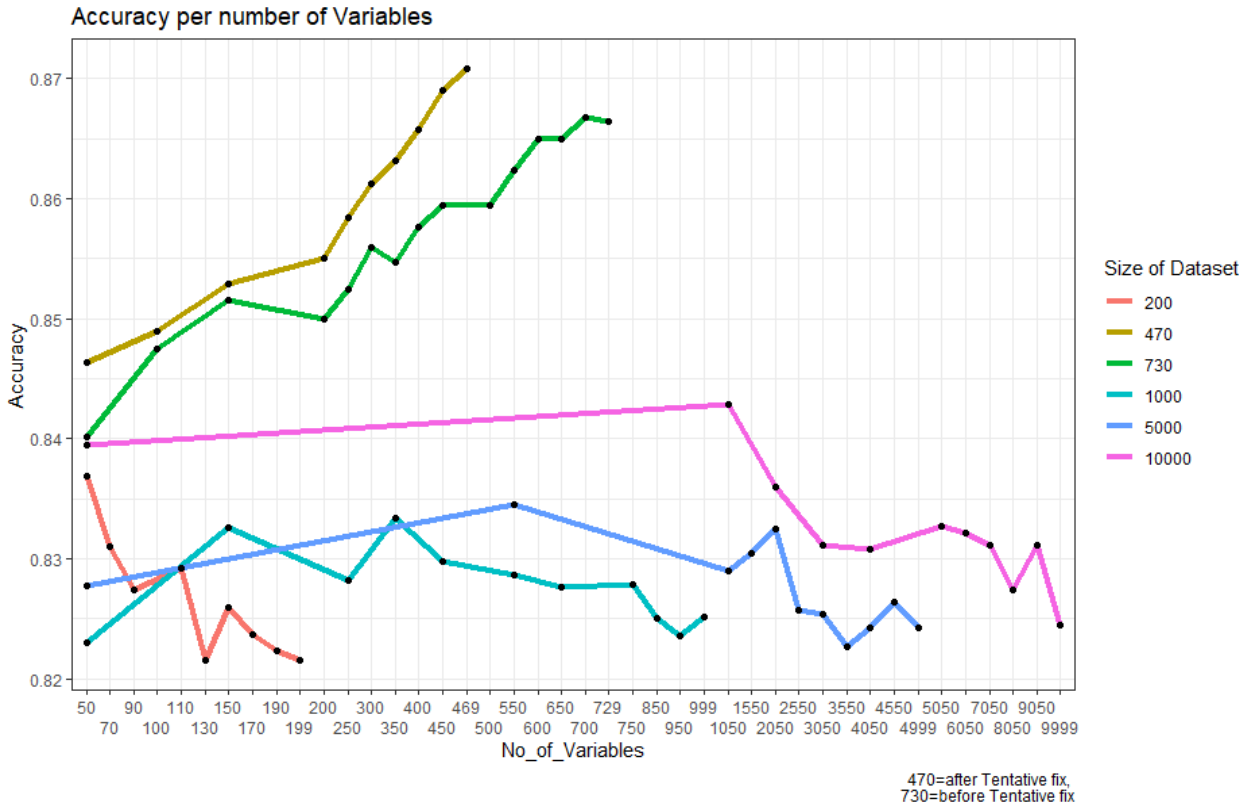
έχουν γενικά μια αύξουσα πορεία τιμών ακρίβειας. Επομένως, ενώ οι διαφορές δεν είναι πολύ μεγάλες, θα επιλεχθεί το μικρότερο σύνολο, το σύνολο των 470.

Ως επιβεβαίωση πως τα σύνολα του Boruta, και συγκεκριμένα το σύνολο που επιλέξαμε, δίνουν την μεγαλύτερη ακρίβεια επαναλήφθηκε η διαδικασία αυτή με νέα σύνολα. Επιπρόσθετα των συνόλων 470 και 730 δημιουργήθηκαν άλλα 4 σύνολα τα οποία προήλθαν από το αρχικό σύνολο των 131.072 μεταβλητών. Τα dataset αυτά είναι του μεγέθους των 200, 1.000, 5.000 και 10.000 τα οποία προέκυψαν ορίζοντας στο καθένα τυχαίες μεταβλητές από το αρχικό. Κάθε σύνολο επαναλήφθηκε 10 φορές. Επομένως τελικά υπάρχουν 10 υποσύνολα των 200, 1.000, 5.000 και 10.000. εξετάζοντας αυτά τα υποσύνολα μαζί με τα σύνολα του boruta στο rfe θα δοθεί μια πιο καθαρή εικόνα της ακρίβειας των χαρακτηριστικών του boruta σε σύγκριση με τυχαία υποσύνολα του αρχικού. Καθένα από τα νέα υποσύνολα εξετάστηκε συνολικά περίπου 100 φορές, κάτι που επιτεύχθηκε ορίζοντας για το καθένα, τα υποσύνολα που θα μελετήσει το RFE να είναι το 1/10 του αρχικού τους μεγέθους. Δηλαδή το 200 μελετούσε ανά 20 χαρακτηριστικά, το 1000 ανά 100 κ.ο.κ. αντίστοιχα για την εξισορρόπηση των δοκιμών τα σύνολα του Boruta μελετήθηκαν ανά 10 χαρακτηριστικά επιπλέον των δοκιμών τους σε κάθε ένα από τα υποσύνολα. Η αναπαράστασή τους φαίνεται παρακάτω.



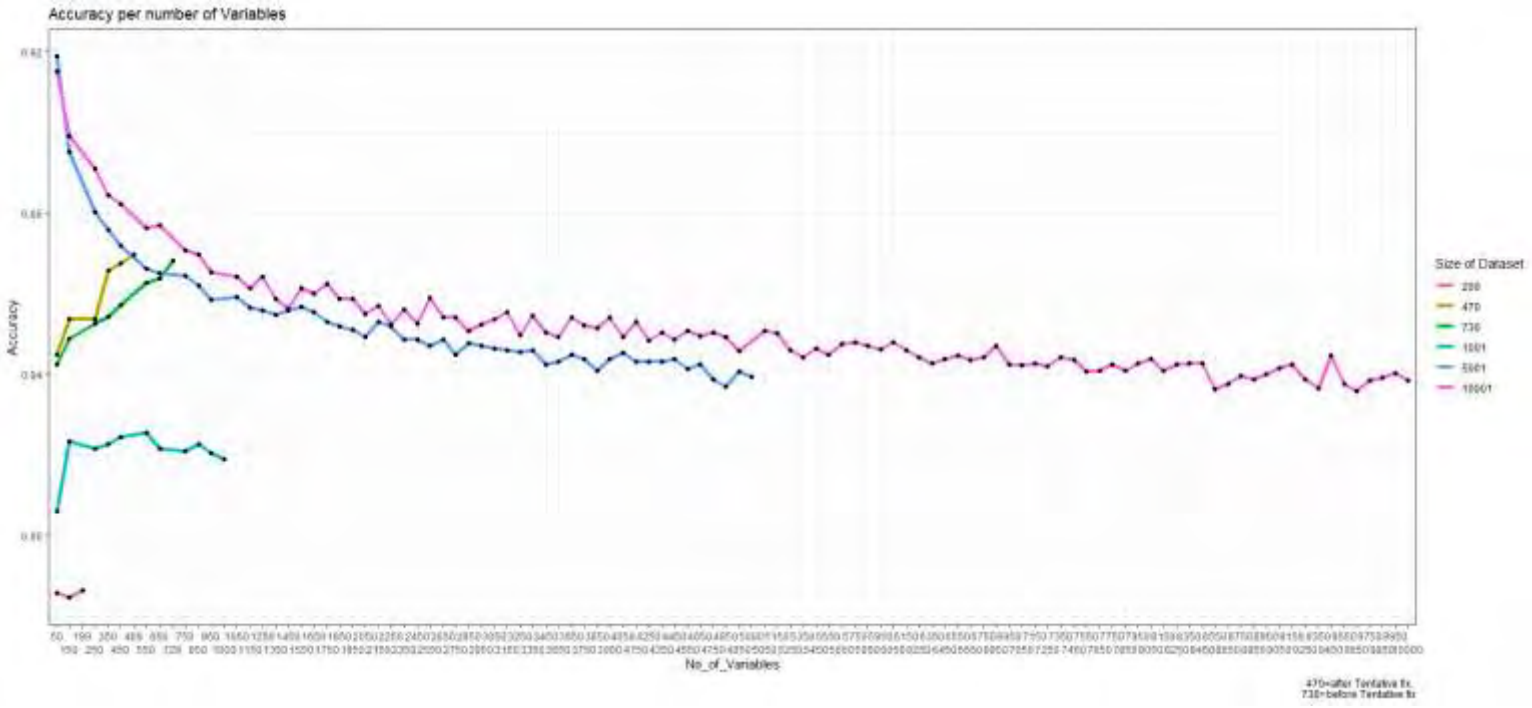
Εικόνα 12 Σύγκριση ακρίβειας συνόλων Boruta με τυχαία υποσύνολα του αρχικού- Box plot

Και η αναπαράστασή τους για κάθε υποσύνολο που εξέτασε το RFE:



Εικόνα 13 Σύγκριση ακρίβειας συνόλων Boruta με τυχαία υποσύνολα του αρχικού - line plot

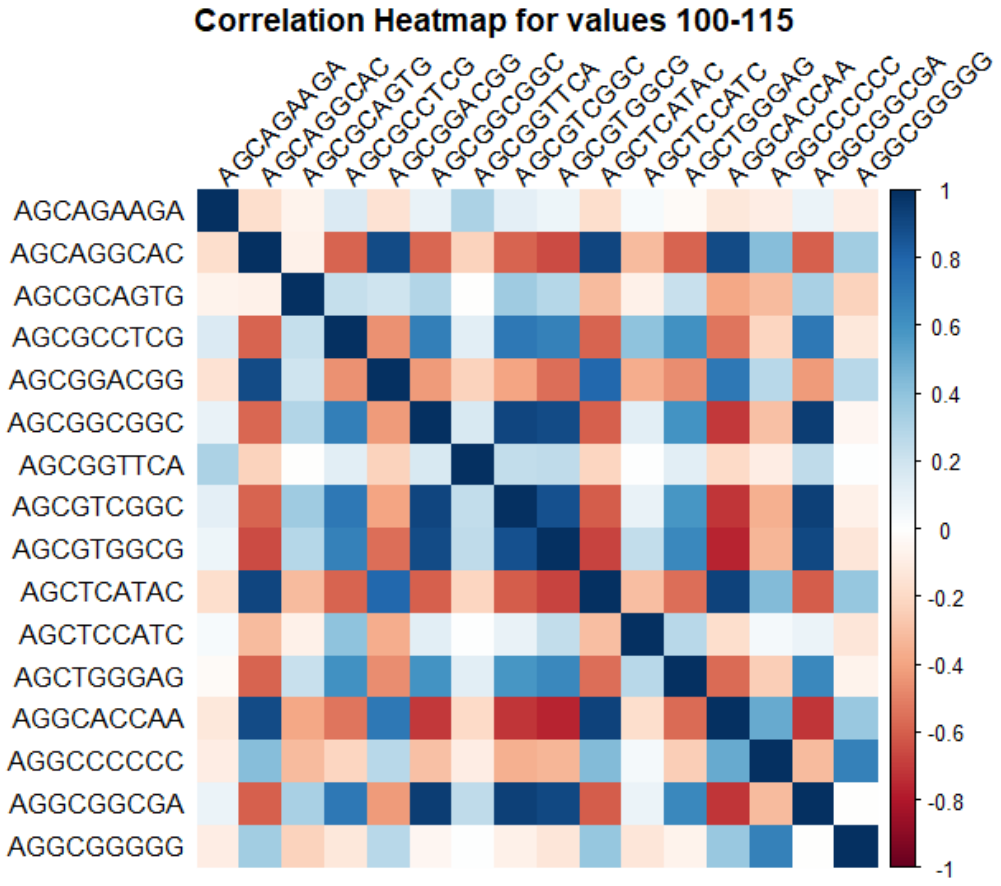
Μια επανάληψη της μεθόδου με όλα τα υποσύνολα για κάθε 100 χαρακτηριστικά αναπαριστάται παρακάτω.



Εικόνα 14 Σύγκριση ακριβείας συνόλων για κάθε 100 μεταβλητές - line plot

4.4 Correlation – variance

Για τη μελέτη της συσχέτισης των χαρακτηριστικών δημιουργείται αρχικά ένας πίνακας με τις συσχετίσεις (correlation matrix), ο οποίος πρακτικά δεν μπορεί να αναπαρασταθεί στο συνολικό του μέγεθος λόγω του όγκου των μεταβλητών. Ένα στιγμιότυπο 15 μεταβλητών στις σειρές 100-115 αναδεικνύεται παρακάτω.



Εικόνα 15 πίνακας αναπαράστασης συσχέτισης μεταβλητών

Το διάγραμμα δείχνει τις αλληλεπιδράσεις των k-mers, με μπλε αυτά που έχουν θετική συσχέτιση και κόκκινο αυτά που έχουν αρνητική συσχέτιση. Η αρίθμηση πάει από το +1 μέχρι το -1 αντίστοιχα με το 0 να σημαίνει μηδενική συσχέτιση. Οι συσχετίσεις της διαγωνίου είναι 1 διότι αντιστοιχούν στις αλληλεπιδράσεις του κάθε χαρακτηριστικού με τον εαυτό του και αυτό αντιστοιχεί σε απόλυτη γραμμική συσχέτιση.

Τα δεδομένα που επιλέγονται είναι αυτά που ξεπερνάνε ένα όριο ποσοστού συσχέτισης. Το όριο αυτό ορίστηκε ως 0.9 δηλαδή όσα κμερς έχουν συσχέτιση

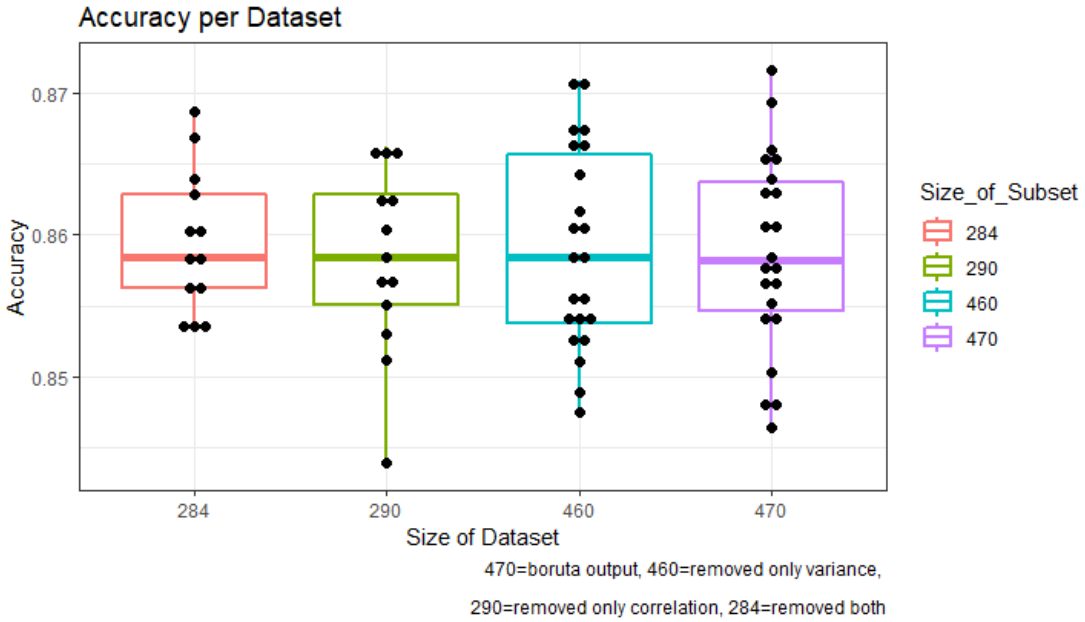
μεγαλύτερη από 90% επιλέγονται ως redundant, ότι δεν προσφέρουν επιπλέον πληροφορία στο μοντέλο και ως αποτέλεσμα 180 μεταβλητές αφαιρέθηκαν από το σύνολο δεδομένων των 470 δημιουργώντας ένα σύνολο με 290 μεταβλητές.

Στη συνέχεια μελετήθηκε το σύνολο των 470 για την διακύμανσης των χαρακτηριστικών. Κανένα από τα χαρακτηριστικά δεν είχε μηδενική διακύμανση αλλά βρέθηκαν 10 που είχαν διακύμανση μικρότερη από 0.015. Τα κμερς αυτά απορρίφτηκαν και το σύνολο που δημιουργείται έχει μέγεθος 460 μεταβλητών.

Τέλος, για την ανάλυση βάση και των δύο μεθόδων, χρησιμοποιώντας το σύνολο δεδομένων των 290 μεταβλητών, όπου έχουν αφαιρεθεί τα κμερς που είχαν μεγάλη συσχέτιση μεταξύ τους, έγινε έλεγχος για τη διακύμανση. Κανένα από τα χαρακτηριστικά δεν είχε μηδενική διακύμανση αλλά βρέθηκαν 6 που είχαν διακύμανση μικρότερη από 0.015. Τα κμερς αυτά απορρίφτηκαν δημιουργώντας ένα σύνολο με 284 μεταβλητές.

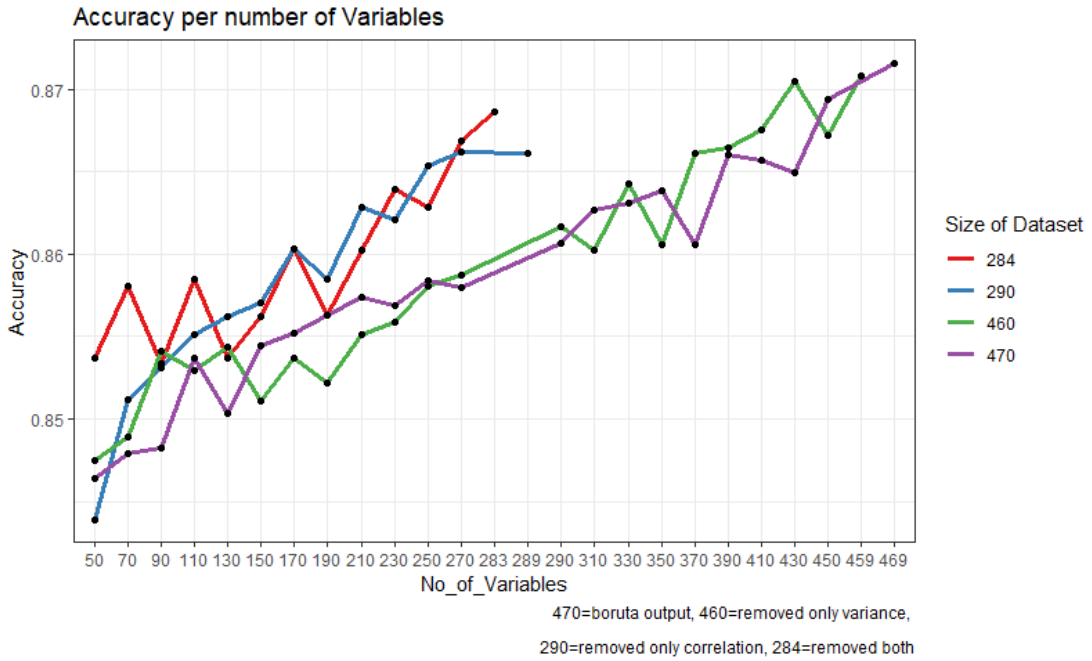
4.5 Τελικά δεδομένα

Ως επιβεβαίωση δοκιμάζεται ξανά το RFE για 3 αυτά σύνολα συν του αρχικού. Τα δεδομένα που συγκρίνονται είναι το τελικό σύνολο των 284 μεταβλητών, το σύνολο των 470 μεταβλητών που είναι αυτό που έβγαλε το Boruta και δύο σύνολα των οποίων το ένα έχει αφαιρέσει μόνο τα correlated features(290) και το άλλο έχει αφαιρέσει μόνο τα low variance(460). Τα αποτελέσματα είναι τα εξής.



Εικόνα 16 Ακρίβεια δεδομένων με correlation, variance και τα δυο και κανένα

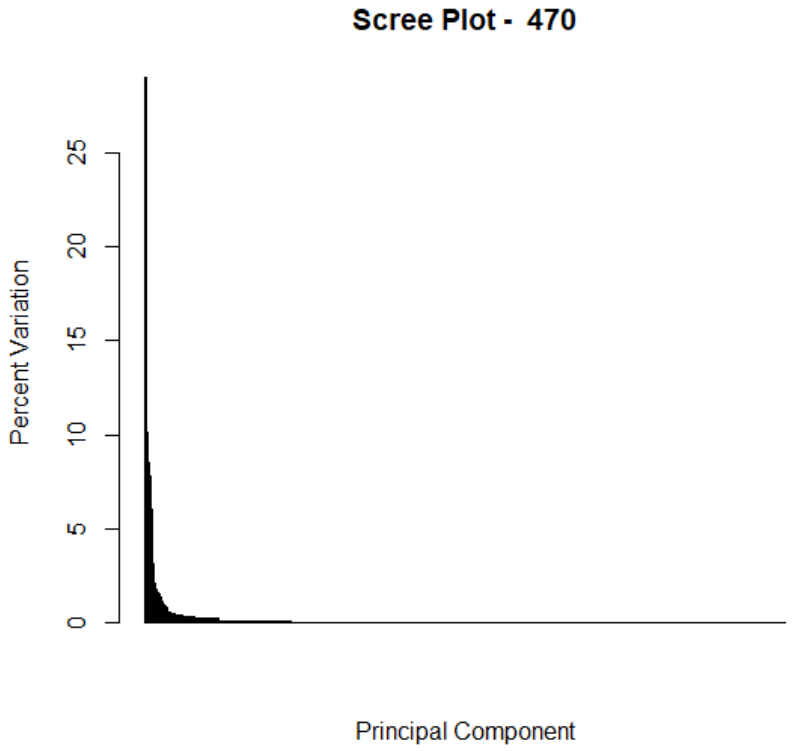
Και αντίστοιχα το διάγραμμα με την ακρίβεια ανά υποσύνολο μεταβλητών σε κάθε σύνολο δεδομένων.



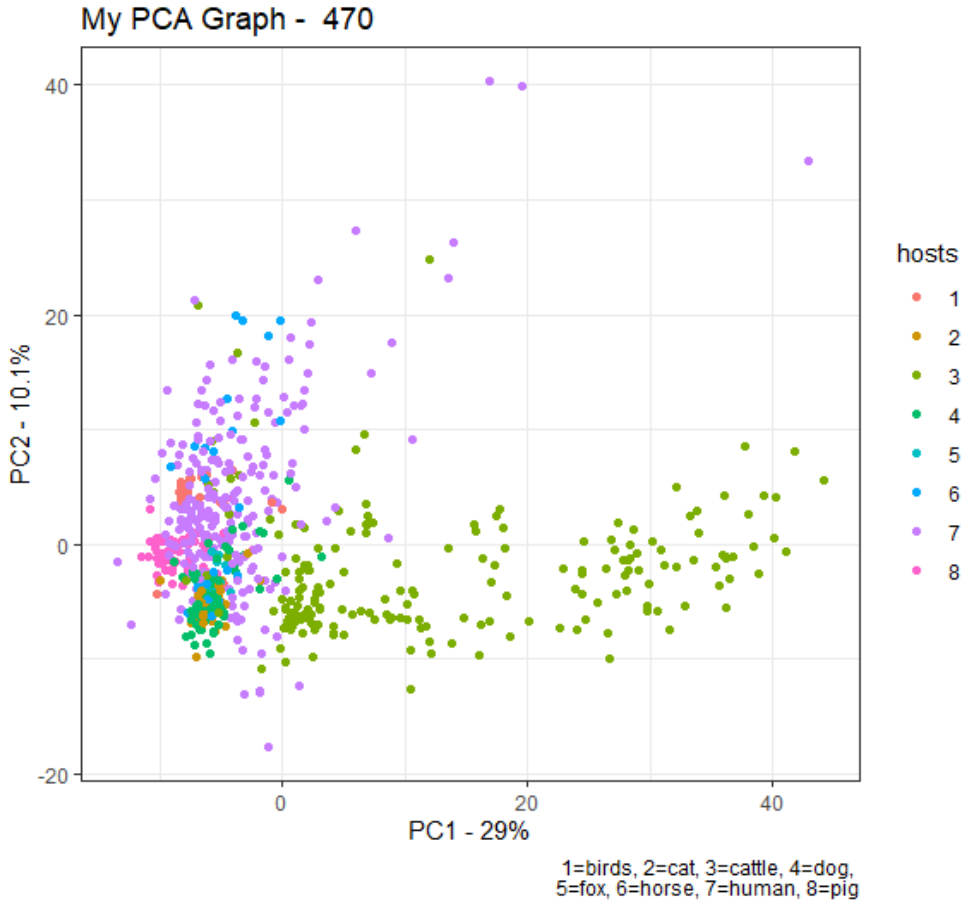
Εικόνα 17 Ακρίβεια δεδομένων ανά υποσύνολο μεταβλητών για correlation, variance και τα δύο και κανένα

Τα αποτελέσματα δεν δείχνουν πολύ καθαρά τη διαφορά, με το σύνολο των 284 να βγάζει κατά βάση την υψηλότερη ακρίβεια στο σύνολο των δεδομένων του αλλά το σύνολο των 470 να βγάζει τελικά την υψηλότερη ακρίβεια γενικά. Τα σύνολα γενικά έχουν πολύ μικρές διαφορές, με ένα εύρος τιμών περίπου από 0.80 μέχρι 0.87. Τα σύνολα που επιλέγονται για την ανάλυση είναι το 284 και το 470.

Η αναπαράσταση των συνόλων αυτών με τεχνικές μείωσης των διαστάσεων PCA και LDA κάθε συνόλου βρίσκεται στη συνέχεια.



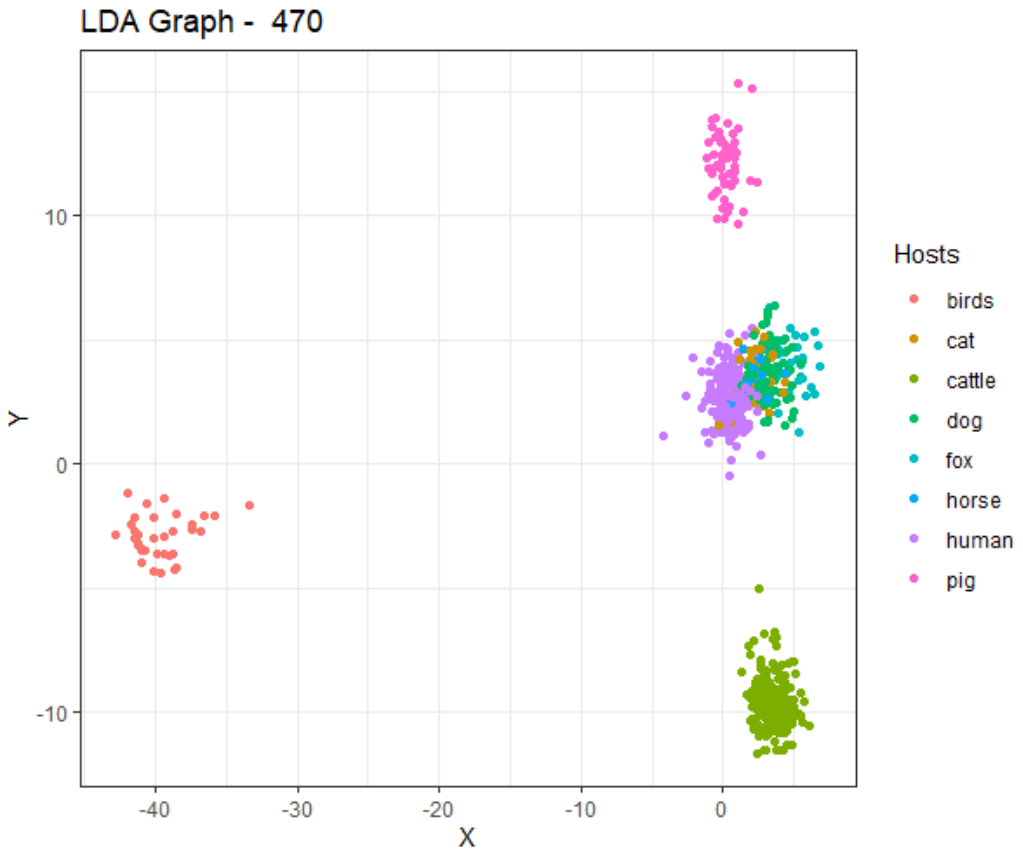
Εικόνα 18 Αναπαράσταση των Principal Components – 470



Εικόνα 19 Αναπαράσταση δεδομένων με PCA - 470

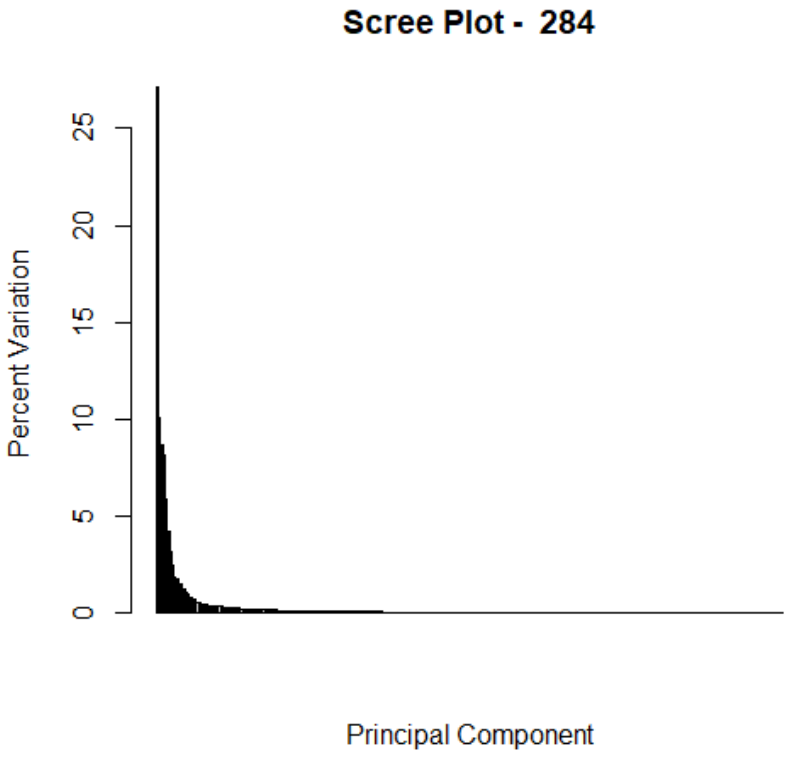
```
> top_kmers ## show the names of the top 100 kmers
  [1] "AGGCACCAA" "ATTTTCGCA" "ACTGATCGA" "AAACTCGGT" "GTCAAACCA" "ACGTGGATC" "AGACGTTCA"
  [8] "GTCCACTGA" "AGTCCTGAA" "AATGGCCCA" "CGGGGGGCA" "GGGGGGCAC" "ATCCGGATA" "AGGGACTGA"
 [15] "CTCCTGTAC" "GGGACTGAC" "CCATTTTCG" "AGCTCATAC" "CGCGTAAGA" "CTCAGAGTC" "CCGGGTCCC"
 [22] "CGGGGATAC" "GCGATGACA" "CTGCGATCC" "CATATCGAC" "GACATGCTC" "CATACAGGG" "GGACGGGGA"
 [29] "CCTCCTTGC" "CTGGGCGTA" "AGCGTGGCG" "AAATGGGCC" "GCCGACGCC" "CCAGGGTAC" "CGACGCTGC"
 [36] "GCGGCCAGC" "GTGCATCCA" "CCGGACCCC" "GACGCGCAC" "ATCAGTCCC" "GGGCCTACA" "CGACGCCGC"
 [43] "CAGCTCATA" "CCTGATTGC" "GCCGCCGAC" "CGGCAGAAC" "CGGCGGCGC" "CAGACCGAG" "GCGCCACGC"
 [50] "GACGCCGCC" "GGCGGCGAC" "ACCCGCTAT" "CAGCGTGGC" "CGATTTCAC" "CCGTCGGCG" "CCTCCCGAC"
 [57] "CCGCCGACG" "CGCCACCCG" "CGTCGGCGC" "AGCGTCGGC" "CCAGCCGGC" "AATTCCGGA" "AGGCGGCGA"
 [64] "CCTCGGCGC" "AGCAGGCAC" "AGCGGCGGC" "CCCGGCAGC" "GCACGGCGC" "GCTGCCGCC" "GCGCCGCGC"
 [71] "CCCGACCGG" "ACGCGGGTC" "AATTTGCCG" "GCGCCTGCC" "GCCGCTGCC" "GCGGCGACC" "CAGCCGCCA"
 [78] "CAGGGCAAA" "ACCCGCCGC" "CAGAGCAAG" "GCGTCGGCC" "CCCGCCGTA" "GGCAGCCCA" "CGCGCGGGC"
 [85] "GCGGGCGCA" "GGCGGGCGA" "CCTCGGCGA" "ATGGCGAAA" "GCCGGGAGA" "CGGCAGCGC" "CGGCGGGTA"
 [92] "AGGGCAAAA" "CGTGGCGCC" "CGCTGCCGC" "GACGCTGCC" "ACGCTGCCG" "CACGGTTAC" "AAGGGCCCC"
 [99] "CGGATCCCC" "CTGCCGACC"
```

Εικόνα 20Τα πιο σημαντικά κμερς βάση PCA – 470

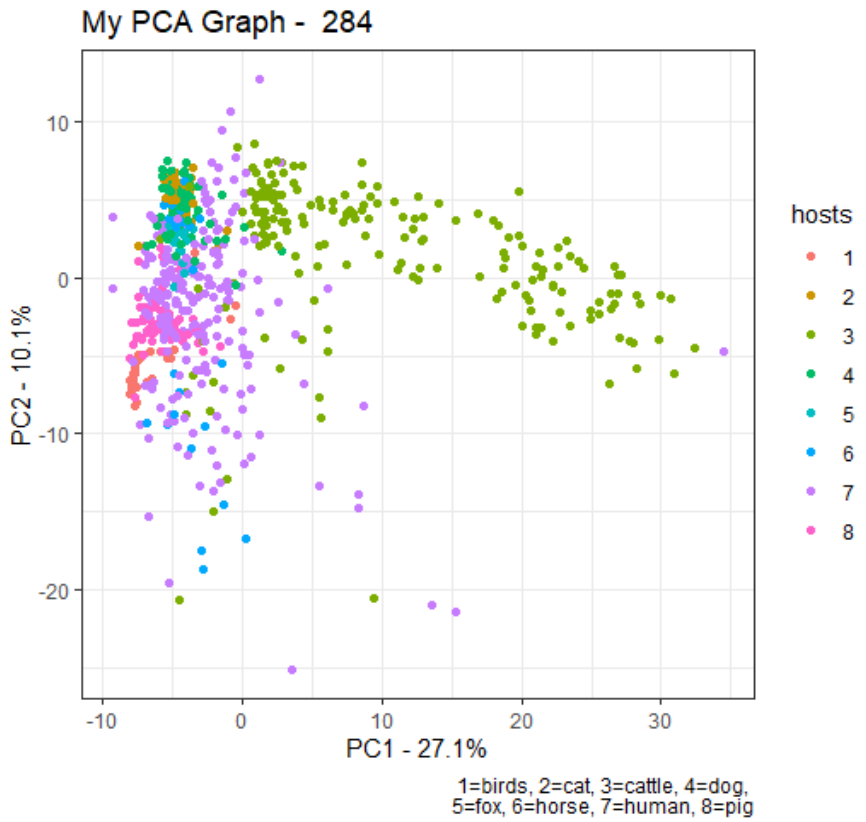


Εικόνα 21 Αναπαράσταση δεδομένων με LDA - 470

Και το σύνολο των 284:



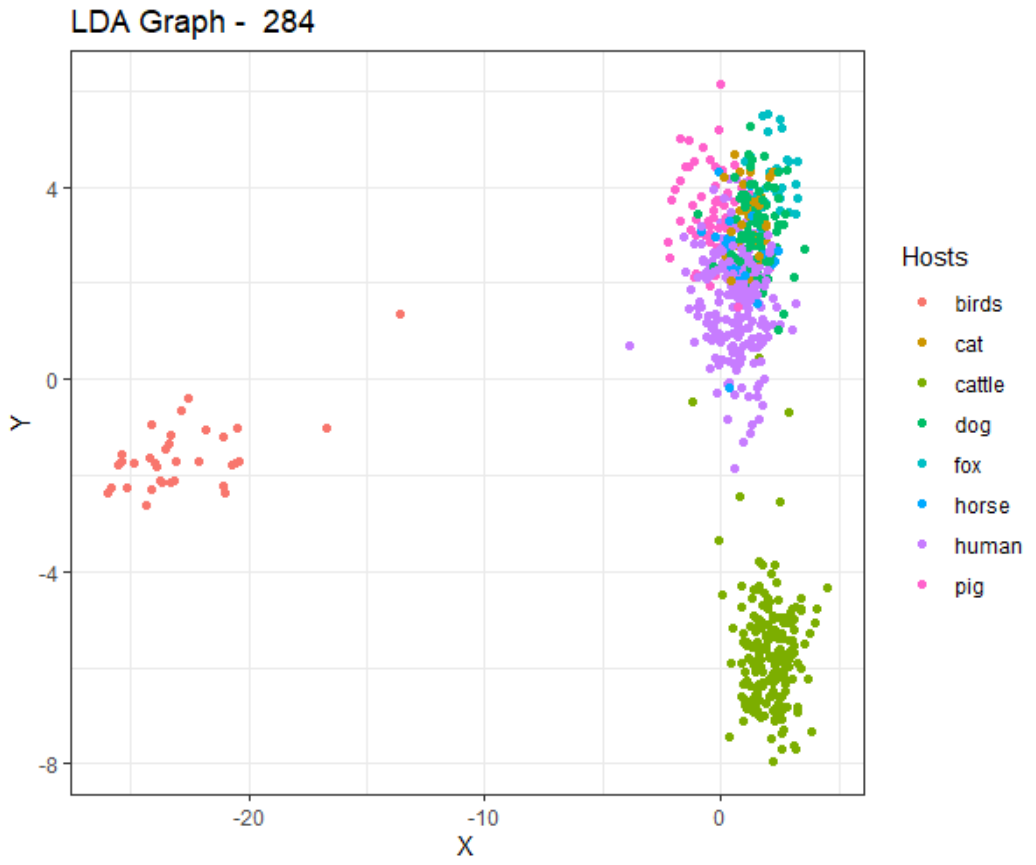
Εικόνα 22 Αναπαράσταση των Principal Components – 284



Εικόνα 23 Αναπαράσταση δεδομένων με PCA - 284

```
> top_kmers ## show the names of the top 100 kmers
  [1] "AGGCACCAA" "ATTTTCGCA" "GTCAAACCA" "ACTGATCGA" "AAACTCGGT" "AGACGTTCA" "ACGTGGATC" "CGGGGGGCA"
  [9] "GTCCACTGA" "AGTCCTGAA" "AATGGCCCA" "ATCCGGATA" "AGGGACTGA" "CATATCGAC" "GGGACTGAC" "CCATTTTCG"
 [17] "CTCCTGTAC" "AGCTCATAC" "GGACGGGGA" "CTCAGAGTC" "CTGGGCGTA" "CGCGTAAGA" "CCGGACCCC" "GTGCATCCA"
 [25] "CAGCTCATA" "CGGCAGAAC" "CGATTTCAC" "CGACGCTGC" "CAGACCGAG" "GCGCCACGC" "CGGCGGCGC" "CCGTCGGCG"
 [33] "CGTCGGCGC" "AATTTGCCG" "CAGGGCAAA" "GCCGCTGCC" "CCCGGCAGC" "AGGCGGCGA" "CAGAGCAAG" "GCACGGCGC"
 [41] "GGCAGCCCA" "GCGCCTGCC" "ACCCGCCGC" "GCGTCGGCC" "ATGGCGAAA" "AAGGGCCCC" "AGGGCAAAA" "GGCGGGCGA"
 [49] "CGGCGGGTA" "GCGGGCGCA" "CGCGCGGGC" "GACGCTGCC" "GCAGCGGCC" "ACGCTGCCG" "GCGGCGTCC" "CGCTGCCGC"
 [57] "CGCCCGCCA" "CGTCCCCCC" "GCGCGGGCC" "ACCTCGGCG" "CGCCCCGGC" "CGCCGCCCG" "CGGGGATCC" "CCCTCGGCG"
 [65] "GCGCCATAC" "GATCAGGCC" "ACCCTCGGC" "AGGGTACGC" "CGTCGGCTA" "ATGGGGATC" "AGGGGGGAC" "AAAAAAATG"
 [73] "CCTGCATCC" "CGGTCGCCC" "AGCGCCTCG" "GCTCTCCAC" "GGGGGGGAC" "AAACCCCCC" "GGGGGGGAA" "CAATGCCCG"
 [81] "GGGGGTTCA" "CCCCAGCAC" "AGAGGGGGG" "CGCTGCCGA" "GGGGGGAAA" "GGCCCCCCC" "CTACGGCAC" "CAGGGGGGG"
 [89] "GGTGCCGAA" "CGGGCCCCC" "GGGTCCTAA" "CCCCCCGGG" "ACGGGGGGG" "ATGGCGACC" "CCGGGAGAC" "ACCCCCCGG"
 [97] "AAAAAGGTA" "GGACCCCCC" "CGGGGGGGC" "GATGCCGTA"
```

Εικόνα 24 Τα πιο σημαντικά κμερς βάση PCA – 284



Εικόνα 25 Αναπαράσταση δεδομένων με LDA - 284

4.6 Random Forest - 470

Για αρχή εκτελείται το μοντέλο με τις προκαθορισμένες ρυθμίσεις.

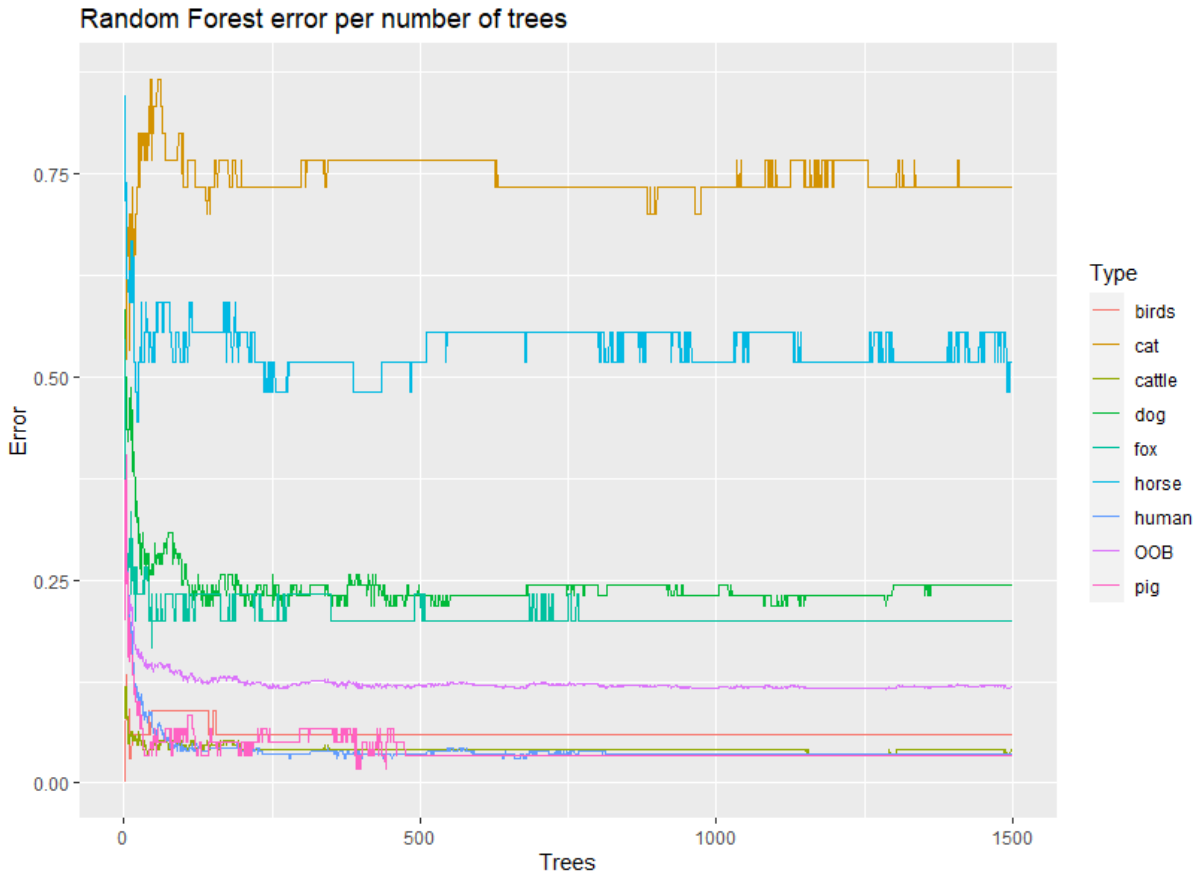
```
Call:
 randomForest(formula = host_categories ~ ., data = k_9_fix, proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 12.79%
Confusion matrix:
       birds cat cattle dog fox horse human pig class.error
birds     32   0      0   0   0     0     2   0  0.05882353
cat        0   9      0  17   0     0     4   0  0.70000000
cattle     0   0    184   2   0     0     6   0  0.04166667
dog        0   5      1  56   0     0    16   0  0.28205128
fox        0   0      0   0  23     0     7   0  0.23333333
horse      0   1      0   9   0    12     5   0  0.55555556
human      0   0      7   2   0     0   219   1  0.04366812
pig        0   0      0   0   0     0     2  58  0.03333333
```

Εικόνα 26 Random Forest με default settings - 470

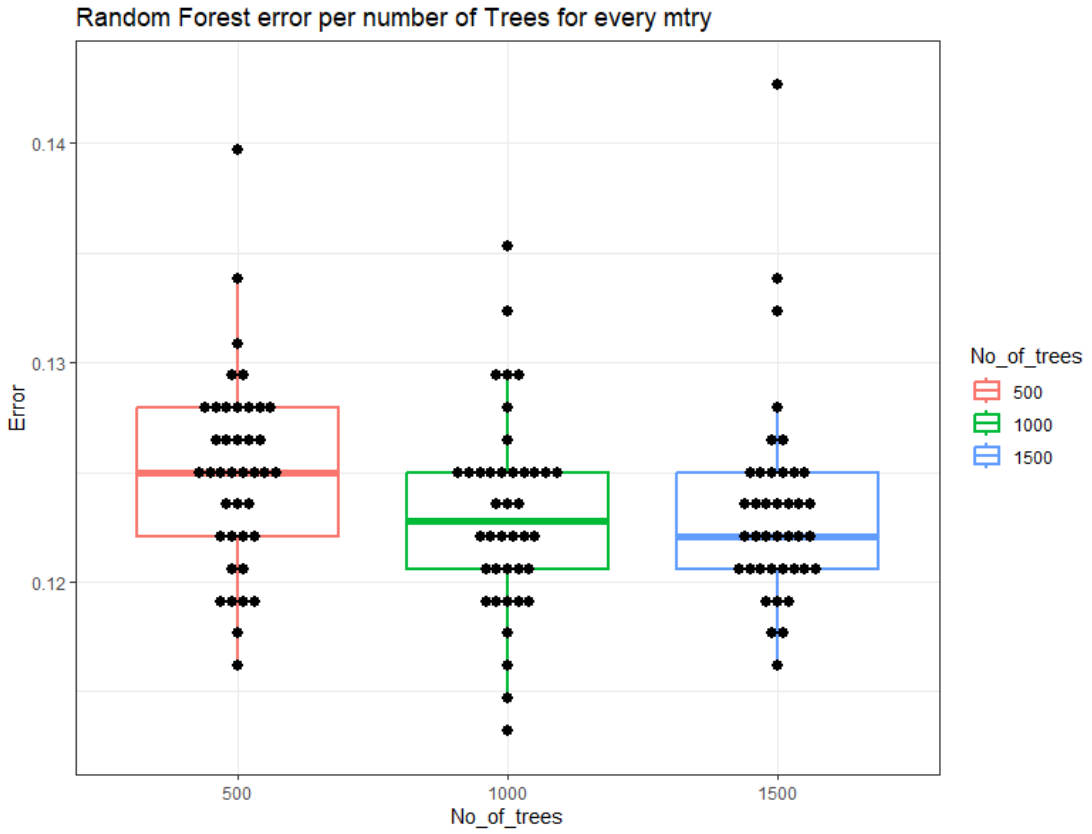
Η εικόνα δείχνει πρώτα τον κώδικα που εκτελέστηκε, τον τύπο ανάλυσης του Random Forest και τις τιμές στις παραμέτρους που χρησιμοποίησε. Μετά δίνει το ποσοστό εσφαλμένης κατηγοριοποίησης των oob δειγμάτων το οποίο είναι 12.79% επομένως έχει ποσοστό ακρίβειας ίσο με 87.21%. Τέλος δίνει το confusion matrix, στο οποίο αναγράφεται αναλυτικά η κατηγοριοποίηση των δειγμάτων σε οργανισμούς. Σε αυτόν τον πίνακα βλέποντας κάθε γραμμή φαίνεται που κατηγοριοποιήθηκαν τα δείγματα του κάθε οργανισμού και στο τέλος το ποσοστό λάθους της κάθε κλάσης.

Για την αναζήτηση του καλύτερου πλήθους δέντρων γίνεται αναπαράσταση του ποσοστού λάθους του κάθε ξενιστή σε κάθε πλήθος δέντρων. Το μοντέλο εκτελέστηκε στα 1500 δέντρα ώστε η αναπαράσταση να περιέχει τα 500, 1000 και 1500.



Εικόνα 27 Error ανά αριθμό δέντρων στο Random Forest

Με την αύξηση στο πλήθος δέντρων φαίνονται κάποια να αυξάνονται και κάποια να μειώνονται και ο μέσος όρος του λάθους είναι σταθερός επομένων φαίνεται πως δεν έχει μεγάλη διαφορά ποιο σύνολο δέντρων θα χρησιμοποιηθεί. Θα μπορούσε να επιλεχθεί ένα αλλά προτιμήθηκε μαζί με τον έλεγχο της επόμενης παραμέτρου να δοκιμαστούν και τα δέντρα συνδυαστικά. Το διάγραμμα με κάθε σύνολο να είναι ο αριθμός των δέντρων και κάθε σημείο είναι τα mtry από το 1 μέχρι το 40.



Εικόνα 28 Error των mtry για κάθε σύνολο δέντρων - 470

Το διάγραμμα δείχνει πως το mtry που δίνει το μικρότερο error είναι στα 1000 δέντρα. Τα αποτελέσματα και των άλλων δύο συνόλων δέντρων δεν διαφέρει πολύ με τα 500 να έχουν την υψηλότερη μέση τιμή λάθους και το 1500 να έχει τη μεγαλύτερη τιμή λάθους αλλά το μικρότερο μέσο όρο. Το αποτέλεσμα που έβγαλε η βελτιστοποίηση είναι πως το μικρότερο error ίσο με 0.1132 το έχει το mtry ίσο με 35 το οποίο είναι στα 1000 δέντρα.

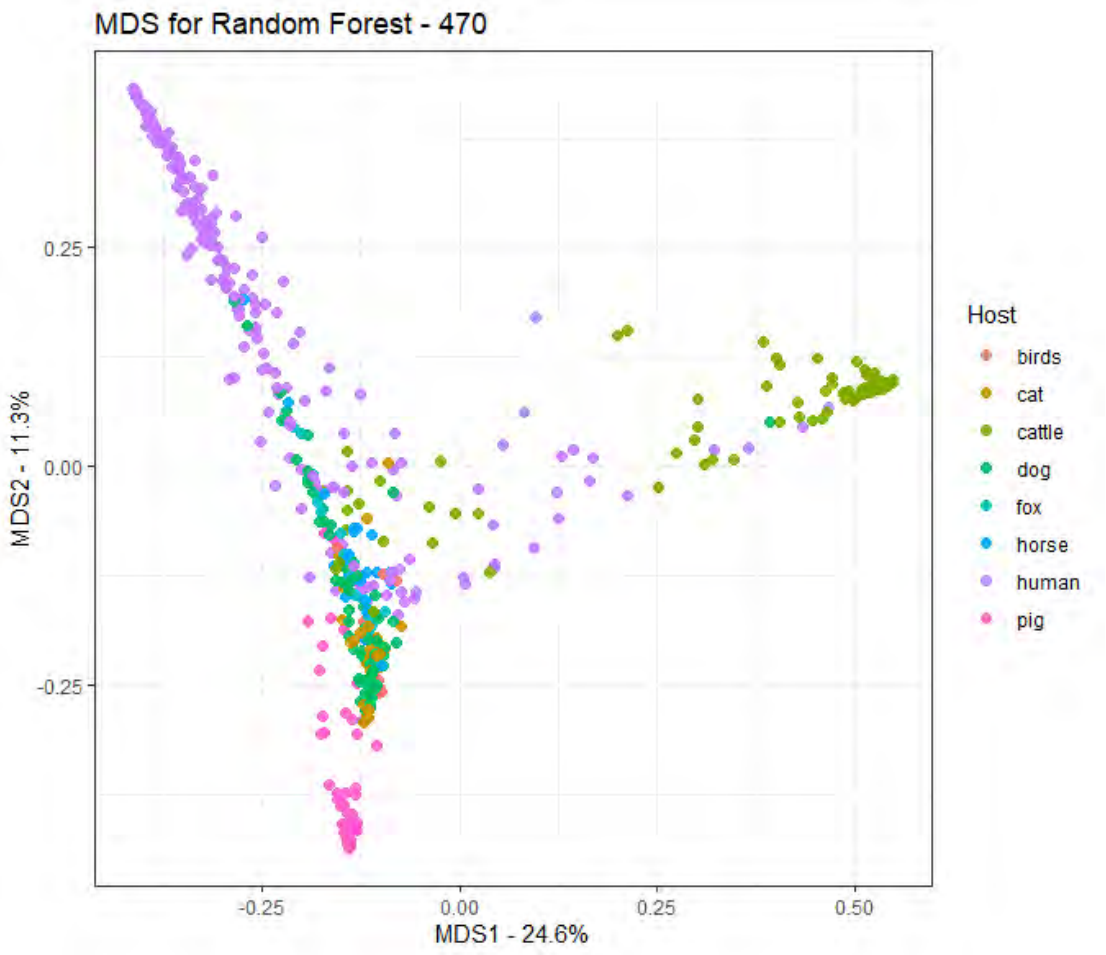
```
Call:
 randomForest(formula = host_categories ~ ., data = k_9_fix, ntree = best_trees,
proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 35

        OOB estimate of  error rate: 11.91%
Confusion matrix:
       birds cat cattle dog fox horse human pig class.error
birds     32   0      0   0   0     0     2   0  0.05882353
cat        0   8      0  17   0     0     5   0  0.73333333
cattle     0   0    184   2   0     0     6   0  0.04166667
dog        0   3      1  60   0     0    14   0  0.23076923
fox        0   0      0   0  24     0     6   0  0.20000000
horse      0   0      0   9   0    12     6   0  0.55555556
human      0   0      6   2   1     0   219   1  0.04366812
pig        0   0      0   0   0     0     0  60  0.00000000
>
```

Εικόνα 29 Random Forest με βελτιστοποιημένες παραμέτρους - 470

Τα νέα αποτελέσματα δίνουν ποσοστό εσφαλμένης κατηγοριοποίησης των oob δειγμάτων ίσο με 11.91% , το οποίο δηλώνει μείωση του error, σε σύγκριση με το αρχικό μοντέλο που είχε τις προκαθορισμένες ρυθμίσεις, κατά 0.88%. Το τελικό accuracy του μοντέλου είναι 88.09%.

Η αναπαράσταση των αποτελεσμάτων με χρήση μεθόδου MDS για μείωση των διαστάσεων φέρνει αυτό το διάγραμμα.



Εικόνα 30 MDS Random Forest - 470

Τέλος αποθηκεύεται μια λίστα με τα κορυφαία 50 k-mers που συνέβαλλαν στην κατηγοριοποίηση του μοντέλου. Το μέτρο σύγκρισης είναι το Mean Decrease Gini. Ενδεικτικά τα πρώτα 5 είναι τα παρακάτω και η αναπαράστασή τους βάση Mean Decrease Gini.

```
GGGGGGGAA GGACCCCCC AGAGGGGGG GGGGGGAAA GCCGAGACC
13.755754 12.443557 10.151826  9.035521  8.054945
```

Εικόνα 31 Πρώτα 5 kmers και η βαθμολογία τους βαση Mean Decrease Gini - 470

4.7 Random Forest - 284

Για το σύνολο δεδομένων των 284 μεταβλητών το μοντέλο του random forest με τις προκαθορισμένες ρυθμίσεις, με αριθμό δέντρων ίσο με 500 και αριθμό μεταβλητών που δοκιμάζονται σε κάθε split των δέντρων απόφασης είναι το εξής.

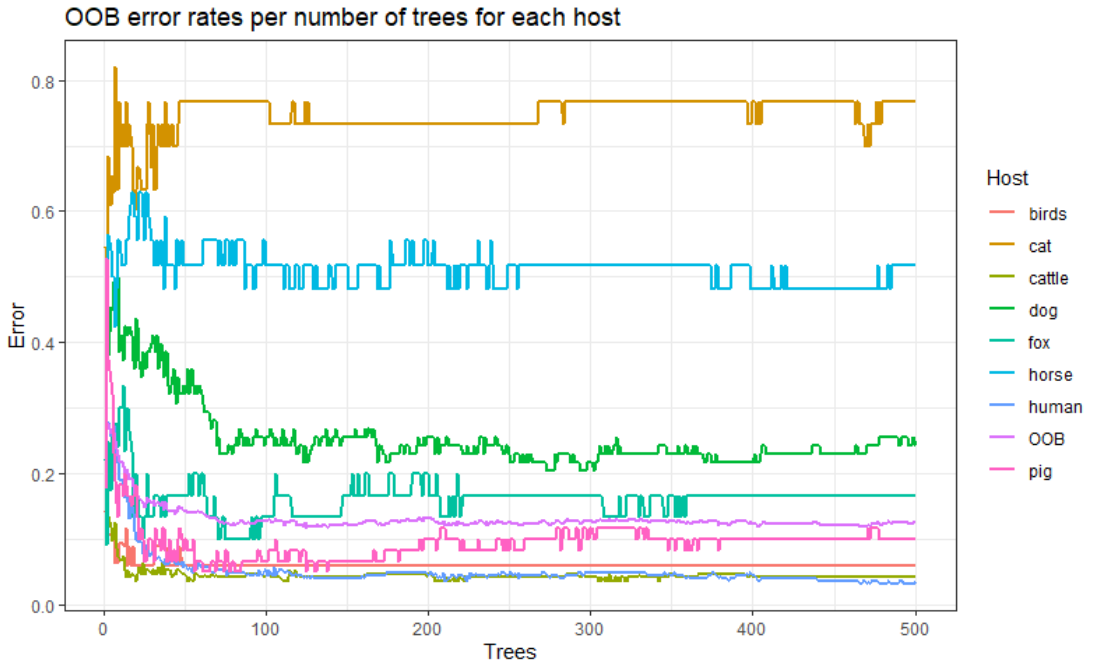
```
Call:
 randomForest(formula = host_categories ~ ., data = fix, proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 12.79%
Confusion matrix:
       birds cat cattle dog fox horse human pig class.error
birds     32   0      0   0   0     0     2   0  0.05882353
cat        0   6      0  19   0     0     5   0  0.80000000
cattle     0   0    184   2   0     0     6   0  0.04166667
dog        0   2      1  59   0     0    16   0  0.24358974
fox        0   0      0   0  24     0     6   0  0.20000000
horse      0   0      0   9   0    13     5   0  0.51851852
human      0   0      6   2   1     0   219   1  0.04366812
pig        0   0      0   0   0     0     4  56  0.06666667
```

Εικόνα 32 Random Forest default settings - 284

Το ποσοστό λάθους είναι ίσο με 12.79% .Ο βαθμός αποτυχημένης πρόβλεψης των επιμέρους οργανισμών αναγράφεται στη τελευταία στήλη του confusion matrix και σε κάποιους οργανισμούς είναι πολύ υψηλό όπως στη γάτα και σε άλλους πολύ χαμηλό όπως στα βοοειδή.

Τα δέντρα ανά οργανισμό βγάζουν αυτό το ποσοστό λάθους ανά σύνολο δέντρων κατά τη μοντελοποίηση.



Εικόνα 33 Error ανά αριθμό δέντρων στο Random Forest - 284

Τα αποτελέσματα αυτά δεν δείχνουν αύξηση ή μείωση του ποσοστού λάθους στα σύνολα 500, 1000 και 1500 επομένως γίνεται και η επόμενη αναπαράσταση.

Εικόνα 34 Error των mtry για κάθε σύνολο δέντρων - 284

Όμοια και εδώ τα αποτελέσματα έχουν όμοιο μέσο ποσοστό λάθους με το 500 και το 1000 να έχουν τη μικρότερη τιμή, το 500 να έχει το μεγαλύτερο μέσο όσο ποσοστού λάθους και το 1500 το μικρότερο και τέλος το 1000 να έχει τη μικρότερη

διακύμανση. Επιλέγεται το μικρότερο mtry το οποίο είναι το 19, στα 500 δέντρα το οποίο βγάζει error 0.1176. Ορίζοντας αυτές τις παραμέτρους στο μοντέλο βγάζει τα παρακάτω αποτελέσματα.

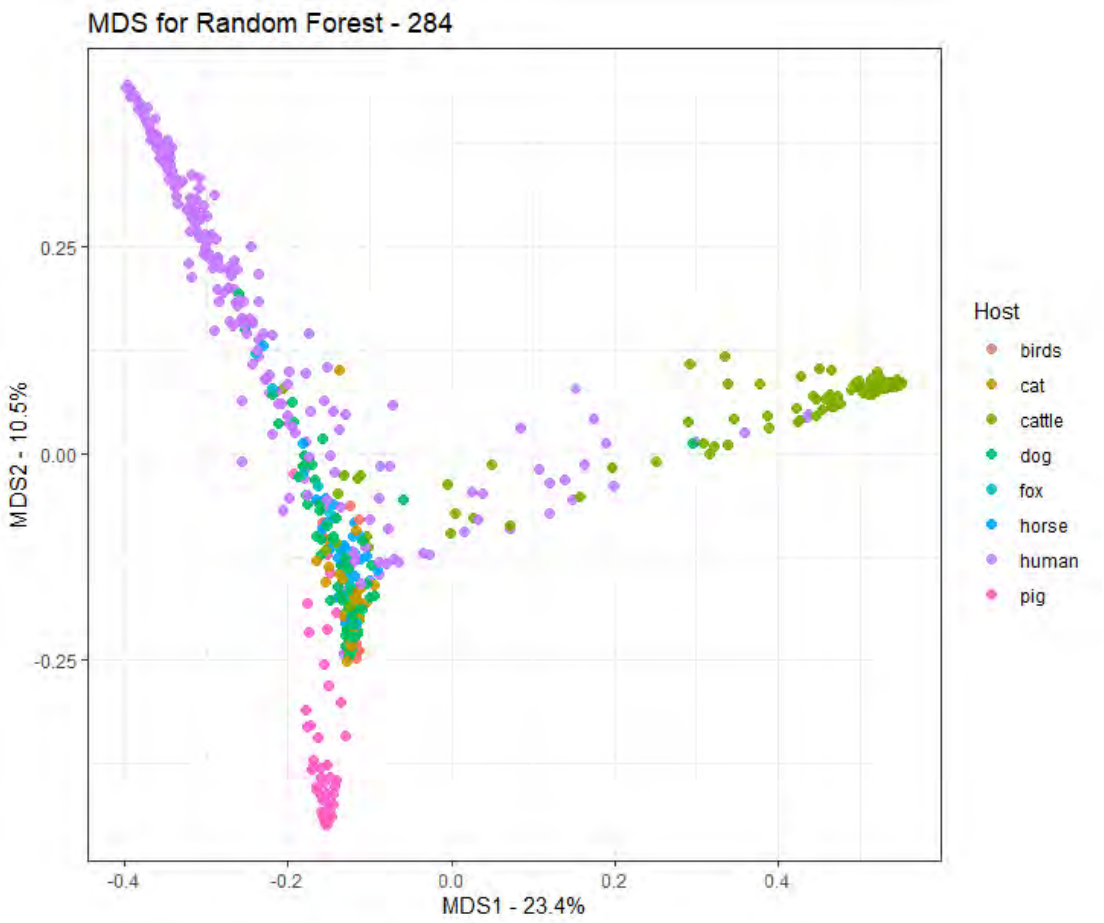
```
Call:
 randomForest(formula = host_categories ~ ., data = fix, ntree = best_trees,
proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 19

        OOB estimate of  error rate: 12.06%
Confusion matrix:
       birds cat cattle dog fox horse human pig class.error
birds     32   0      0   0   0     0     2   0  0.05882353
cat        0   7      0  19   0     0     4   0  0.76666667
cattle     0   0    184   2   0     0     6   0  0.04166667
dog        0   2      1  63   0     0    12   0  0.19230769
fox        0   0      0   0  25     0     5   0  0.16666667
horse      0   0      0  11   0    12     4   0  0.55555556
human      0   0      7   2   1     0   219   0  0.04366812
pig        0   0      0   0   0     0     4  56  0.06666667
```

Εικόνα 35 Random Forest με βελτιστοποιημένες παραμέτρους - 284

Τα νέα αποτελέσματα δίνουν ποσοστό εσφαλμένης κατηγοριοποίησης των oob δειγμάτων ίσο με 12.06% , το οποίο δηλώνει μείωση του error, σε σύγκριση με το αρχικό μοντέλο που είχε τις προκαθορισμένες ρυθμίσεις, κατά 0.73%. Το τελικό accuracy του μοντέλου είναι 87.94%.

Η αναπαράσταση των δειγμάτων με MDS.



Εικόνα 36 MDS Random Forest - 284

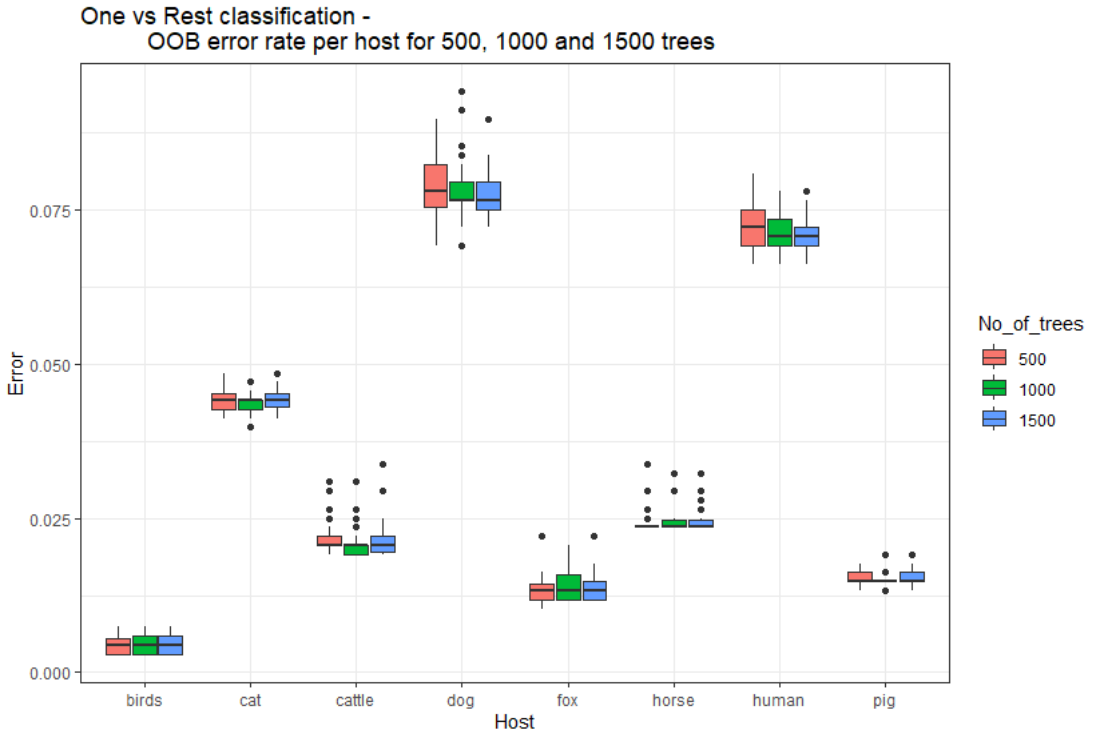
Τέλος, αποθηκεύονται τα 50 πιο σημαντικά κμερς της κατηγοριοποίησης των οργανισμών του Random Forest. Το μέτρο σύγκρισης είναι το Mean Decrease Gini. Ενδεικτικά αυτά είναι τα 5 πρώτα.

```
GGGGGGGAA GGACCCCCC GGGGGGAAA GGGGGGGGA AGAGGGGGG
10.218348  9.667253  9.154491  6.782530  6.725892
```

Εικόνα 37 Πρώτα 5 k-mers και η βαθμολογία τους βάση Mean Decrease Gini - 284

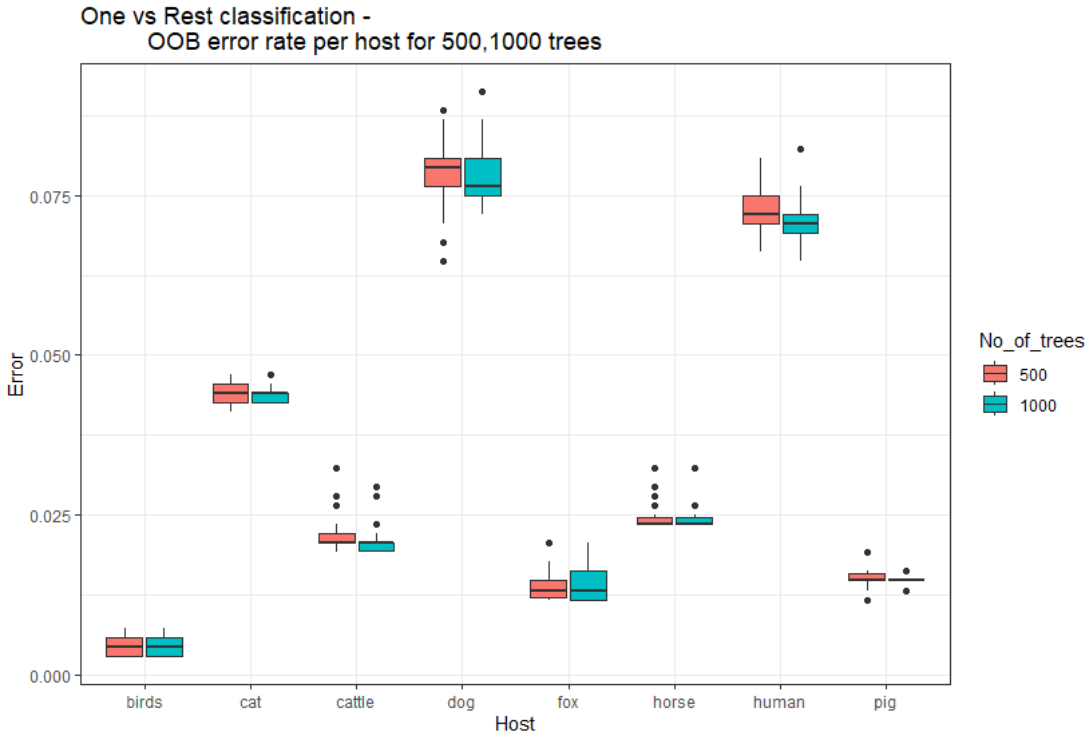
4.8 One vs All – 470

Για αρχή έγινε η δοκιμή της μοντελοποίησης για όλους τους οργανισμούς με 500, 1000, και 1500 δέντρα ώστε να αποφασιστεί αν τα 1500 χρειάζονται στην ανάλυση. Ο λόγος είναι γιατί η χρήση αυτών επιβραδύνει πολύ τη ταχύτητα των αποτελεσμάτων. Όμοια με το Random Forest ο πίνακας δείχνει τη πιθανότητα λάθους για κάθε σύνολο δέντρων, αυτή τη φορά για κάθε οργανισμό.



Εικόνα 38 Σύγκριση error για κάθε σύνολο δέντρων μεταξύ 500, 1000 και 1500 - 470

Παρακάτω είναι ο ίδιος πίνακας για τα 500 και 1000 δέντρα.



Εικόνα 39 Σύγκριση error για κάθε σύνολο δέντρων μεταξύ 500 και 1000 - 470

Χρησιμοποιώντας επομένως τις παραμέτρους ntrees για 500 και 1000 δέντρα και mtry για τους αριθμούς από το 1 μέχρι το 42, βγαίνουν αυτά τα αποτελέσματα στο μοντέλο. Οι επόμενες εικόνες δείχνουν το μοντέλο με τις προκαθορισμένες ρυθμίσεις (ntrees = 500, mtry = 21) και τα βελτιστοποιημένα αποτελέσματα για κάθε οργανισμό.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 7.21%
Confusion matrix:
      human other class.error
human   190    39  0.17030568
other    10   441  0.02217295
>
```

Εικόνα 40 One vs All Random Forest - Human - default parameters - 470

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 16

        OOB estimate of  error rate: 6.91%
Confusion matrix:
      human other class.error
human   189    40  0.17467249
other     7   444  0.01552106
>
```

Εικόνα 41 One vs All Random Forest - Human - tuned parameters - 470

Για τον άνθρωπο το μοντέλο έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 7.21% και ορίζοντας το mtry 16 και το ntree 1000 το ποσοστό πέφτει 0.3% στο 6.91%. Το ποσοστό επιτυχίας του μοντέλου είναι 93.09%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 2.06%
Confusion matrix:
       cattle other class.error
cattle    184     8  0.04166667
other       6   482  0.01229508
>
```

Εικόνα 42 One vs All Random Forest - Cattle - default parameters - 470

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 9

        OOB estimate of  error rate: 1.76%
Confusion matrix:
       cattle other class.error
cattle    184     8 0.041666667
other       4   484 0.008196721
>
```

Εικόνα 43 One vs All Random Forest - Cattle - tuned parameters - 470

Για τα βοοειδή έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 2.06% και ορίζοντας το mtry 9 και το ntree 500 το ποσοστό πέφτει 0.3% στο 1.76%. Το ποσοστό επιτυχίας του μοντέλου είναι 98.24%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 0.44%
Confusion matrix:
      birds other class.error
birds    31     3  0.08823529
other     0   646  0.00000000
>
```

Εικόνα 44 One vs All Random Forest - Birds - default parameters - 470

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 22

        OOB estimate of  error rate: 0.29%
Confusion matrix:
      birds other class.error
birds    32     2  0.05882353
other     0   646  0.00000000
>
```

Εικόνα 45 One vs All Random Forest - Birds - tuned parameters - 470

Για τα πουλιά έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 0.44% και ορίζοντας το mtry 22 και το ntree 500 το ποσοστό πέφτει 0.15% στο 0.29%. Το ποσοστό επιτυχίας του μοντέλου είναι 99.71%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 1.62%
Confusion matrix:
      other pig class.error
other   619   1 0.001612903
pig      10  50 0.166666667
>
```

Εικόνα 46 One vs All Random Forest - Pig - default parameters - 470

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 12

        OOB estimate of  error rate: 1.32%
Confusion matrix:
      other pig class.error
other   620   0        0.00
pig       9  51        0.15
>
```

Εικόνα 47 One vs All Random Forest - Pig - tuned parameters - 470

Για το γουρούνι έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 1.62% και ορίζοντας το mtry 12 και το ntree 500 το ποσοστό πέφτει 0.2% στο 1.32%. Το ποσοστό επιτυχίας του μοντέλου είναι 98.68%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 2.35%
Confusion matrix:
      horse other class.error
horse    11    16   0.5925926
other     0   653   0.0000000
>
```

Εικόνα 48 One vs All Random Forest - Horse - default parameters - 470

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 42

        OOB estimate of  error rate: 2.21%
Confusion matrix:
      horse other class.error
horse    12    15   0.5555556
other     0   653   0.0000000
>
```

Εικόνα 49 One vs All Random Forest - Horse - tuned parameters - 470

Για το άλογο έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 2.35% και ορίζοντας το mtry 42 και το ntree 500 το ποσοστό πέφτει 0.14% στο 2.21%. Το ποσοστό επιτυχίας του μοντέλου είναι 97.79%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 7.94%
Confusion matrix:
      dog other class.error
dog    39    39  0.50000000
other  15   587  0.02491694
>
```

Εικόνα 50 One vs All Random Forest - Dog - default parameters - 470

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 26

        OOB estimate of  error rate: 7.06%
Confusion matrix:
      dog other class.error
dog    45    33  0.42307692
other  15   587  0.02491694
>
```

Εικόνα 51 One vs All Random Forest - Dog - tuned parameters - 470

Για τον σκύλο το μοντέλο έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 7.94% και ορίζοντας το mtry 26 και το ntree 500 το ποσοστό πέφτει 0.88% στο 7.06%. Το ποσοστό επιτυχίας του μοντέλου είναι 92.94%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 4.41%
Confusion matrix:
      cat other class.error
cat     2    28 0.933333333
other   2   648 0.003076923
>
```

Εικόνα 52 One vs All Random Forest - Cat - default parameters - 470

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 29

        OOB estimate of  error rate: 4.12%
Confusion matrix:
      cat other class.error
cat     5    25 0.833333333
other   3   647 0.004615385
>
```

Εικόνα 53 One vs All Random Forest - Cat - tuned parameters - 470

Για τη γάτα έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 4.41% και ορίζοντας το mtry 29 και το ntree 500 το ποσοστό πέφτει 0.29% στο 4.12%. Το ποσοστό επιτυχίας του μοντέλου είναι 95.88%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 21

        OOB estimate of  error rate: 1.32%
Confusion matrix:
      fox other class.error
fox    21     9         0.3
other   0   650         0.0
>
```

Εικόνα 54 One vs All Random Forest - Fox - default parameters - 470

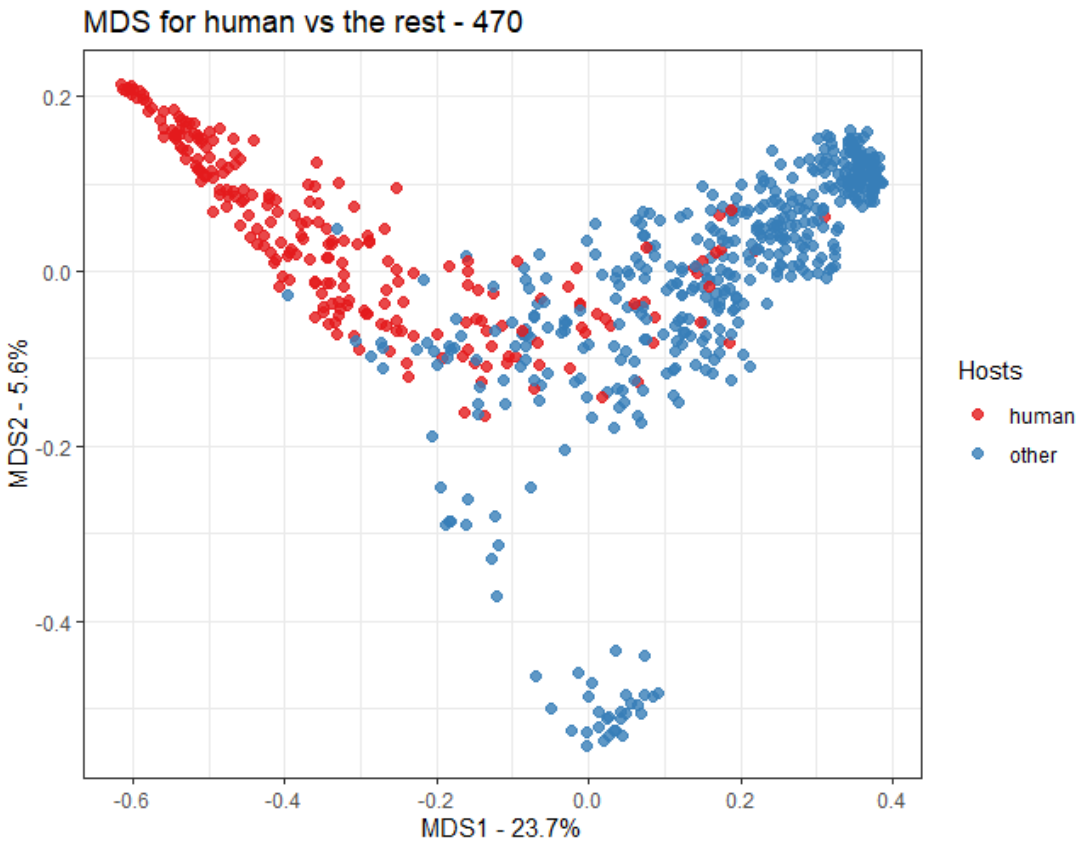
```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 31

        OOB estimate of  error rate: 1.03%
Confusion matrix:
      fox other class.error
fox    23     7   0.2333333
other   0   650   0.0000000
>
```

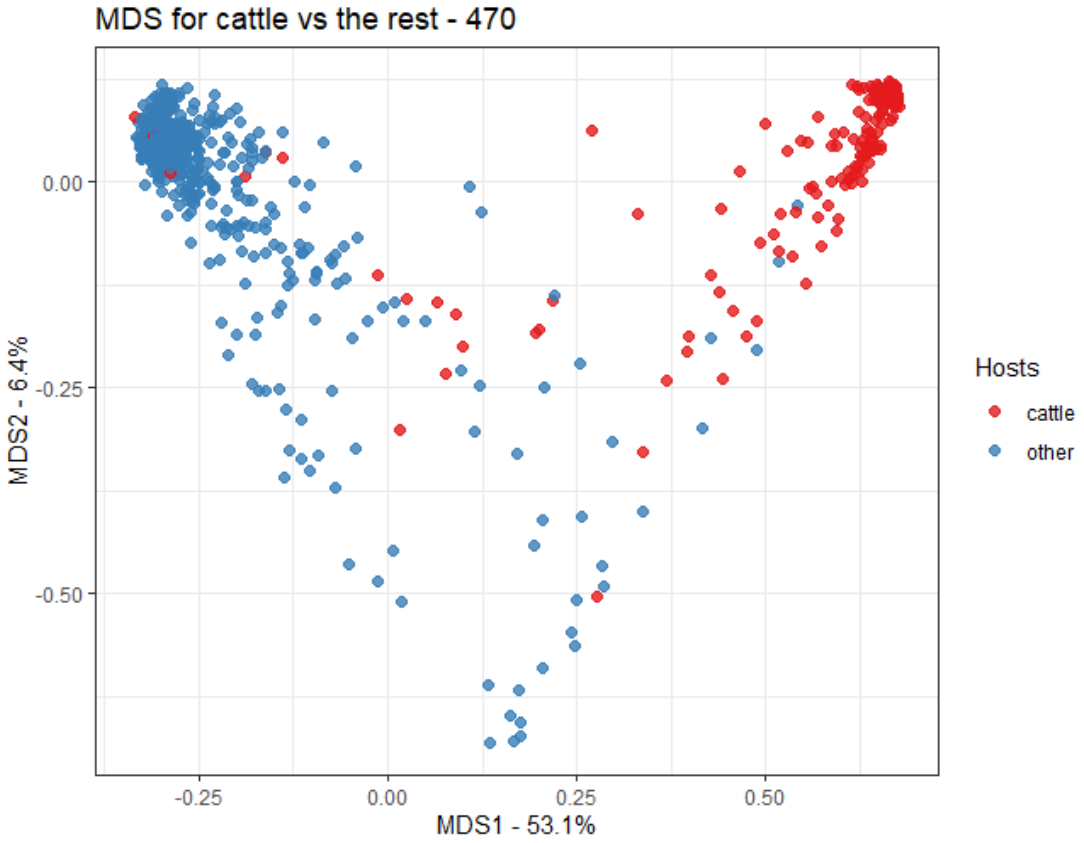
Εικόνα 55 One vs All Random Forest - Fox - tuned parameters - 470

Τέλος, για την αλεπού το μοντέλο έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 1.32% και ορίζοντας το mtry 31 και το ntree 500 το ποσοστό πέφτει 0.29% στο 1.03%. Το ποσοστό επιτυχίας του μοντέλου είναι 98.97%.

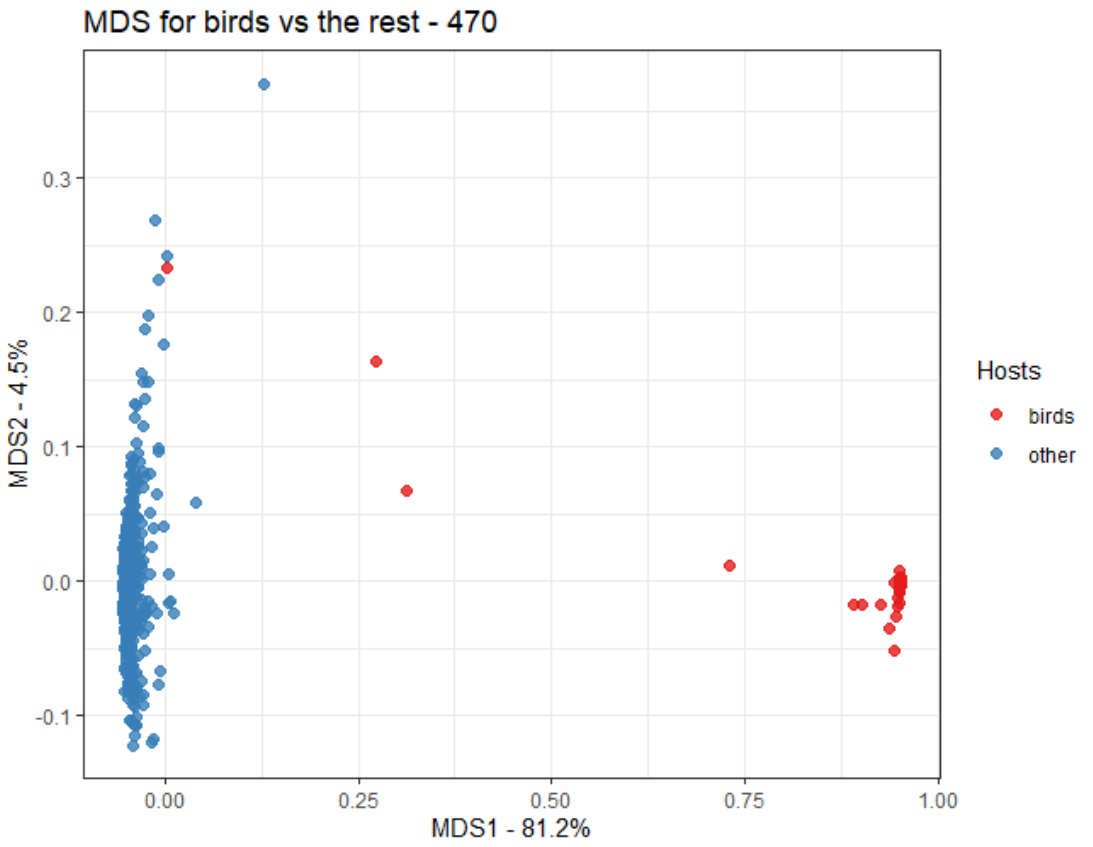
Όπως και στο Random Forest, έτσι και εδώ αναπαριστώνται οι αποστάσεις των δειγμάτων με διαγράμματα MDS. Αυτή τη φορά εκτελείται για κάθε οργανισμό ενάντια στους υπολοίπους. Στα διαγράμματα αναγράφεται το MDS1 και MDS2 και τα ποσοστά τους και ο κάθε οργανισμός χρωματίζεται με κόκκινο ενώ οι υπόλοιποι με μπλε. Σημείωση πως για το γουρούνι (Εικόνα 59) τα χρώματα είναι ανάποδα.



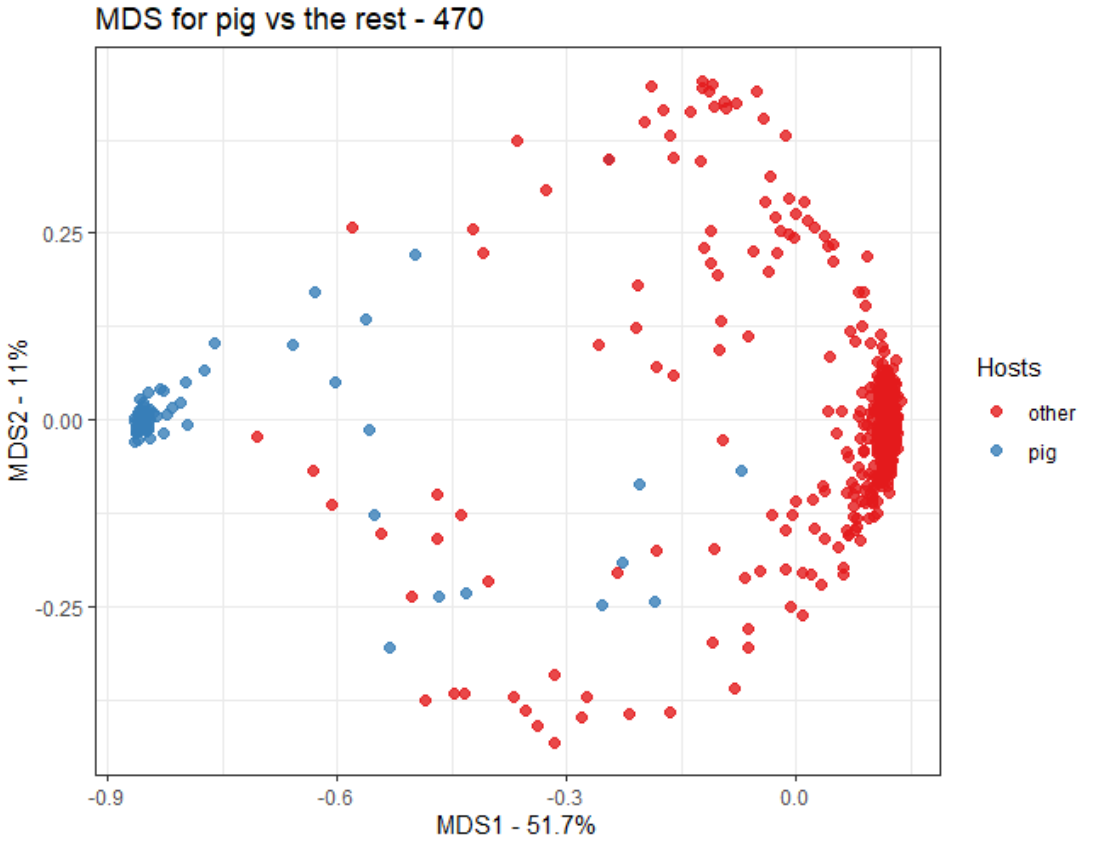
Εικόνα 56 MDS OvA Human - 470



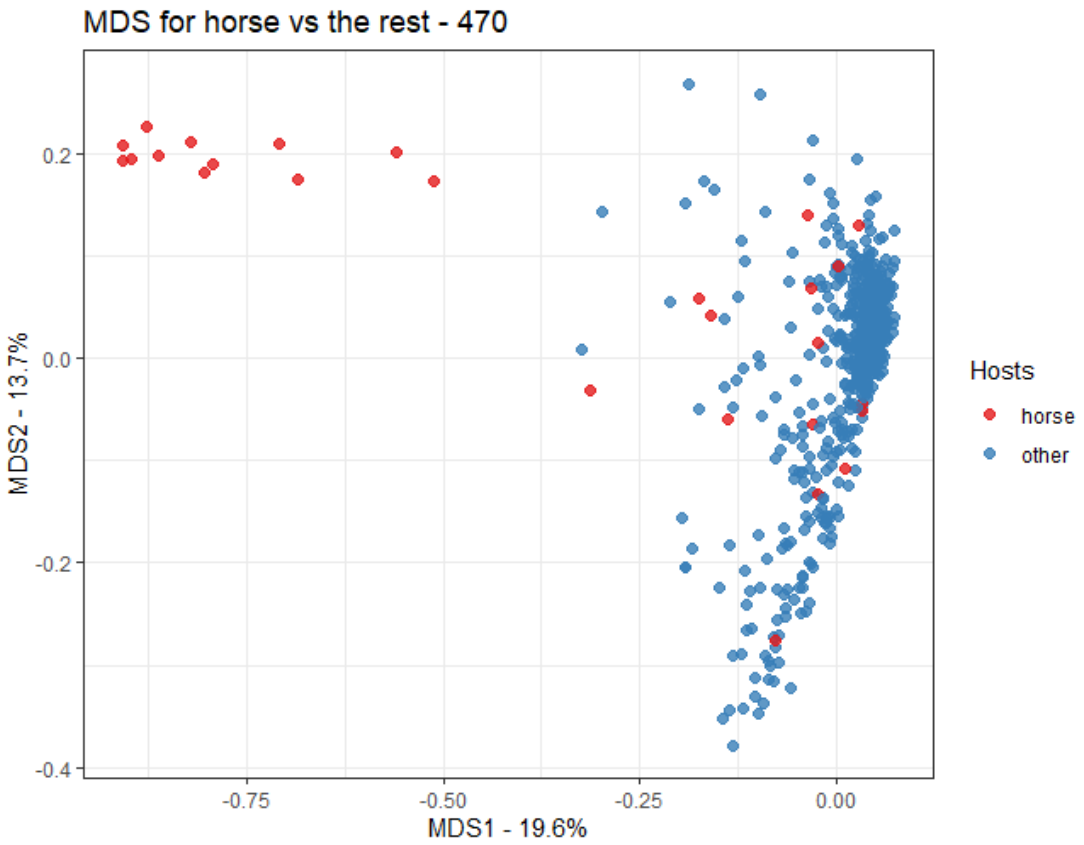
Εικόνα 57 MDS OvA Cattle - 470



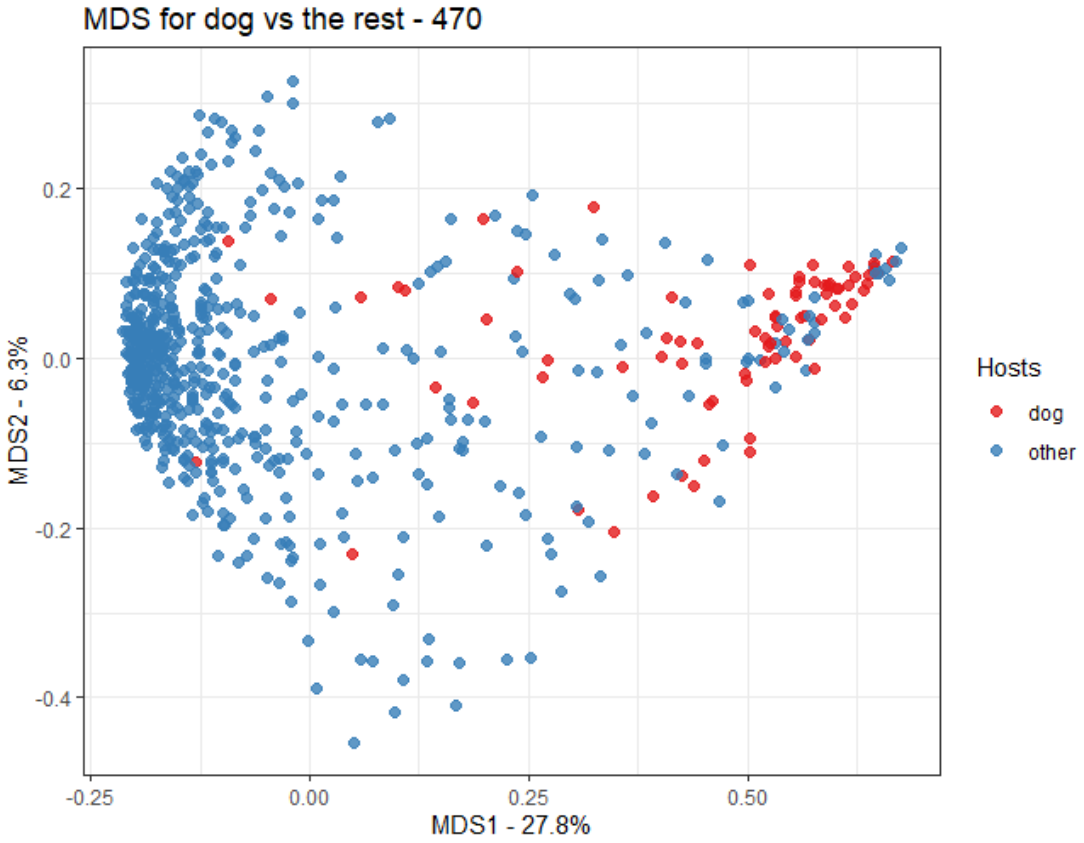
Εικόνα 58 MDS OvA Birds - 470



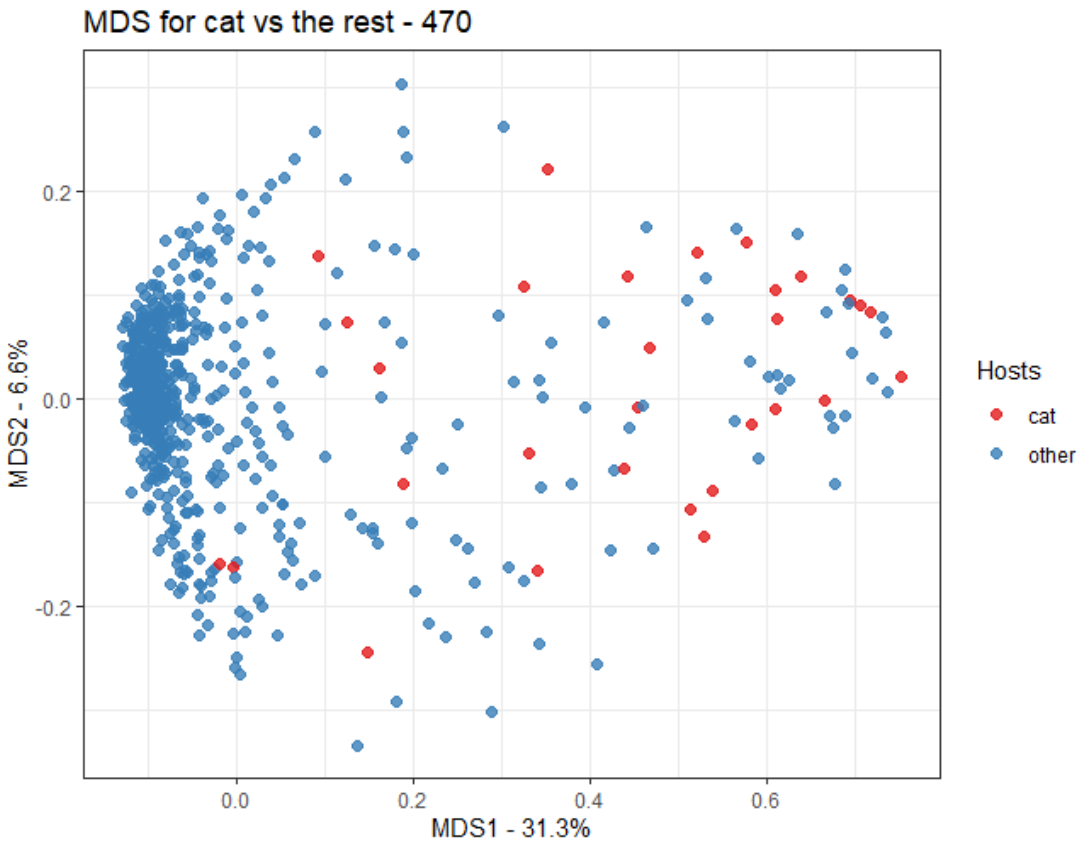
Εικόνα 59 MDS OvA Pig - 470



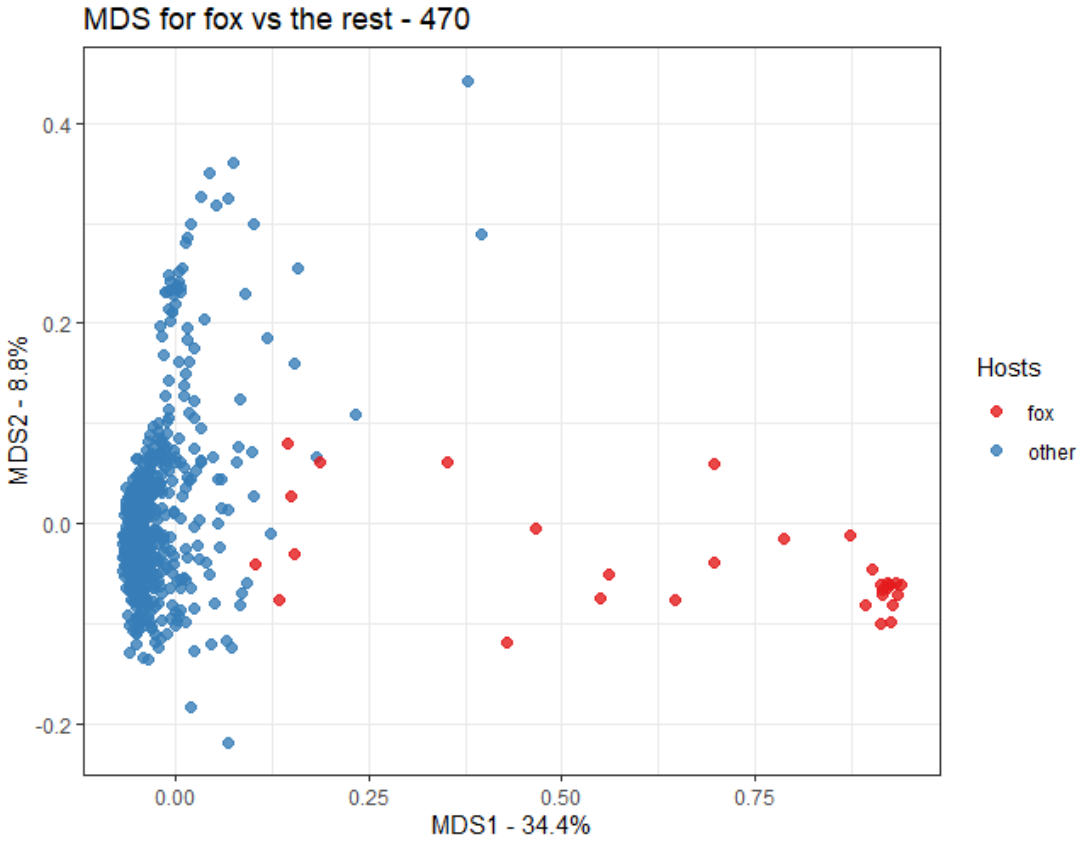
Εικόνα 60 MDS OvA Horse - 470



Εικόνα 61 MDS OvA Dog - 470



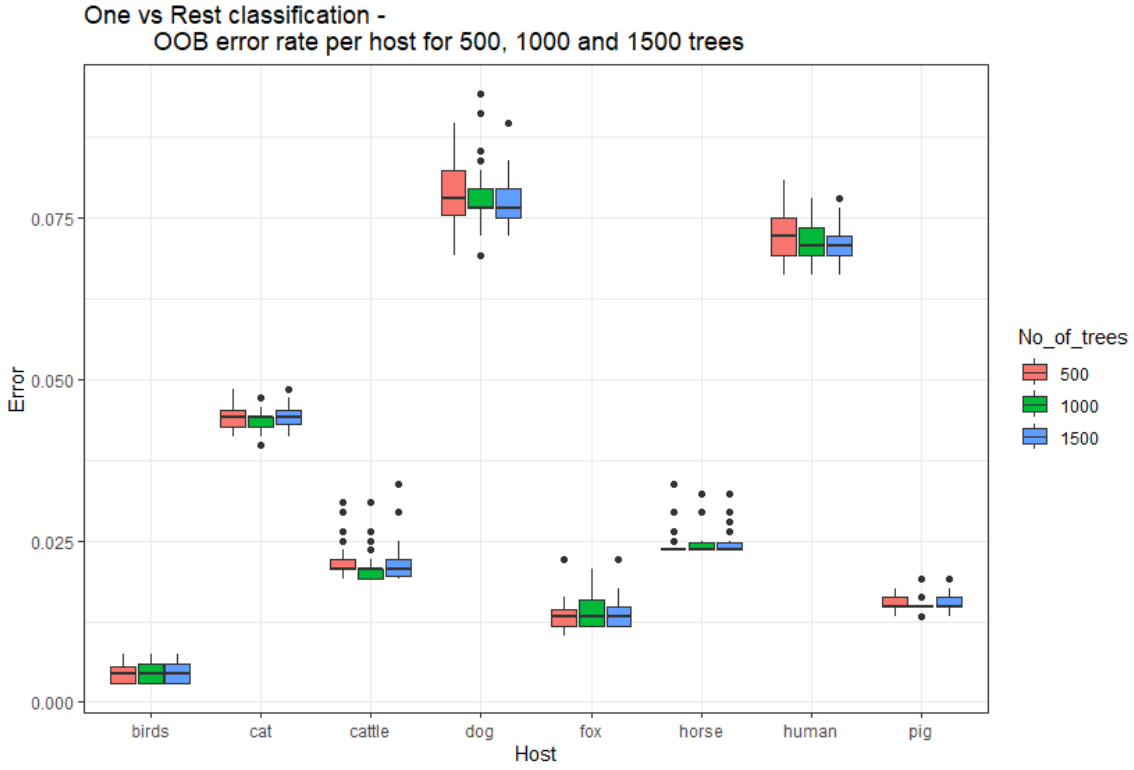
Εικόνα 62 MDS OvA Cat - 470



Εικόνα 63 MDS OvA Fox - 470

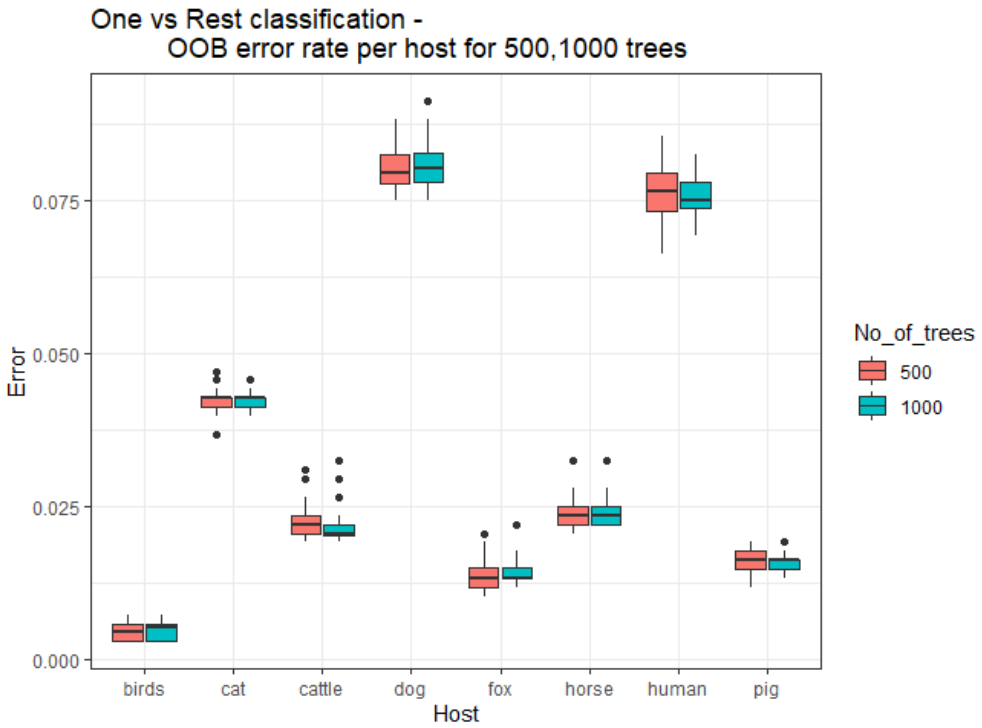
4.9 One vs All – 284

Όμοια για το σύνολο δεδομένων των 284 μεταβλητών γίνεται η μοντελοποίηση αρχικά για όλους τους οργανισμούς με 500, 1000, και 1500 δέντρα ώστε να αποφασιστεί αν τα 1500 χρειάζονται στην ανάλυση. Ο πίνακας δείχνει τη πιθανότητα λάθους για κάθε σύνολο δέντρων, για κάθε οργανισμό.



Εικόνα 64 Σύγκριση error για κάθε σύνολο δέντρων μεταξύ 500, 1000 και 1500 - 284

Και για τα σύνολα 500 και 1000:



Εικόνα 65 Σύγκριση error για κάθε σύνολο δέντρων μεταξύ 500, και 1000 - 284

Χρησιμοποιώντας επομένως τις παραμέτρους ntrees για 500 και 1000 δέντρα και mtry για τους αριθμούς από το 1 μέχρι το 32, εφόσον το 16 είναι το προκαθορισμένο. Βγαίνουν αυτά τα αποτελέσματα στο μοντέλο. Οι επόμενες εικόνες δείχνουν το μοντέλο με τις προκαθορισμένες ρυθμίσεις (ntrees = 500, mtry = 16) και τα βελτιστοποιημένα αποτελέσματα για κάθε οργανισμό.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 7.79%
Confusion matrix:
      human other class.error
human   188    41  0.17903930
other    12   439  0.02660754
> |
```

Εικόνα 66 One vs All Random Forest - Human - default parameters - 284

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 28

        OOB estimate of  error rate: 6.76%
Confusion matrix:
      human other class.error
human   193    36  0.15720524
other    10   441  0.02217295
>
```

Εικόνα 67 One vs All Random Forest - Human - tuned parameters - 284

Για τον άνθρωπο, το μοντέλο έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 7.79% και ορίζοντας το mtry 28 και το ntree 500 το ποσοστό πέφτει 1.03% στο 6.76%. Το ποσοστό επιτυχίας του μοντέλου είναι 93.09%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 2.06%
Confusion matrix:
       cattle other class.error
cattle    184     8  0.04166667
other       6   482  0.01229508
>
```

Εικόνα 68 One vs All Random Forest - Cattle - default parameters - 284

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 12

        OOB estimate of  error rate: 1.91%
Confusion matrix:
       cattle other class.error
cattle    184     8  0.04166667
other       5   483  0.01024590
>
```

Εικόνα 69 One vs All Random Forest - Cattle - tuned parameters - 284

Για τα βοοειδή έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 2.06% και ορίζοντας το mtry 12 και το ntree 500 το ποσοστό πέφτει 0.15% στο 1.91%. Το ποσοστό επιτυχίας του μοντέλου είναι 98.09%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 0.44%
Confusion matrix:
      birds other class.error
birds    31     3  0.08823529
other     0   646  0.00000000
>
```

Εικόνα 70 One vs All Random Forest - Birds - default parameters - 284

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 19

        OOB estimate of  error rate: 0.29%
Confusion matrix:
      birds other class.error
birds    32     2  0.05882353
other     0   646  0.00000000
>
```

Εικόνα 71 One vs All Random Forest - Birds - tuned parameters - 284

Για τα πουλιά έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 0.44% και ορίζοντας το mtry 19 και το ntree 500 το ποσοστό πέφτει 0.15% στο 0.29%. Το ποσοστό επιτυχίας του μοντέλου είναι 99.71%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 1.47%
Confusion matrix:
      other pig class.error
other   620   0   0.0000000
pig      10  50   0.1666667
>
```

Εικόνα 72 One vs All Random Forest - Pig - default parameters - 284

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 8

        OOB estimate of  error rate: 1.32%
Confusion matrix:
      other pig class.error
other   620   0        0.00
pig       9  51        0.15
>
```

Εικόνα 73 One vs All Random Forest - Pig - tuned parameters - 284

Για τα πουλιά έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 1.47% και ορίζοντας το mtry 8 και το ntree 500 το ποσοστό πέφτει 0.15% στο 1.32%. Το ποσοστό επιτυχίας του μοντέλου είναι 98.68%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 2.35%
Confusion matrix:
      horse other class.error
horse    11    16   0.5925926
other     0   653   0.0000000
>
```

Εικόνα 74 One vs All Random Forest - Horse - default parameters - 284

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 22

        OOB estimate of  error rate: 2.21%
Confusion matrix:
      horse other class.error
horse    12    15   0.5555556
other     0   653   0.0000000
>
```

Εικόνα 75 One vs All Random Forest - Horse - tuned parameters - 284

Για το άλογο έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 1.47% και ορίζοντας το mtry 8 και το ntree 500 το ποσοστό πέφτει 0.15% στο 1.32%. Το ποσοστό επιτυχίας του μοντέλου είναι 98.68%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 7.94%
Confusion matrix:
      dog other class.error
dog    38    40  0.51282051
other  14   588  0.02325581
>
```

Εικόνα 76 One vs All Random Forest - Dog - default parameters - 284

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 32

        OOB estimate of  error rate: 7.21%
Confusion matrix:
      dog other class.error
dog    44    34  0.43589744
other  15   587  0.02491694
>
```

Εικόνα 77 One vs All Random Forest - Dog - tuned parameters - 284

Για τον σκύλο έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 7.94% και ορίζοντας το mtry 32 και το ntree 1000 το ποσοστό πέφτει 0.73% στο 7.21%. Το ποσοστό επιτυχίας του μοντέλου είναι 92.79%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 4.41%
Confusion matrix:
      cat other class.error
cat     1    29 0.966666667
other   1   649 0.001538462
>
```

Εικόνα 78 One vs All Random Forest - Cat - default parameters - 284

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 9

        OOB estimate of  error rate: 3.97%
Confusion matrix:
      cat other class.error
cat     4    26 0.866666667
other   1   649 0.001538462
>
```

Εικόνα 79 One vs All Random Forest - Cat - tuned parameters - 284

Για τη γάτα έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 4.41% και ορίζοντας το mtry 32 και το ntree 1000 το ποσοστό πέφτει 0.44% στο 3.97%. Το ποσοστό επιτυχίας του μοντέλου είναι 96.03%.

```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 16

        OOB estimate of  error rate: 1.32%
Confusion matrix:
      fox other class.error
fox    21     9         0.3
other   0   650         0.0
>
```

Εικόνα 80 One vs All Random Forest - Fox - default parameters - 284

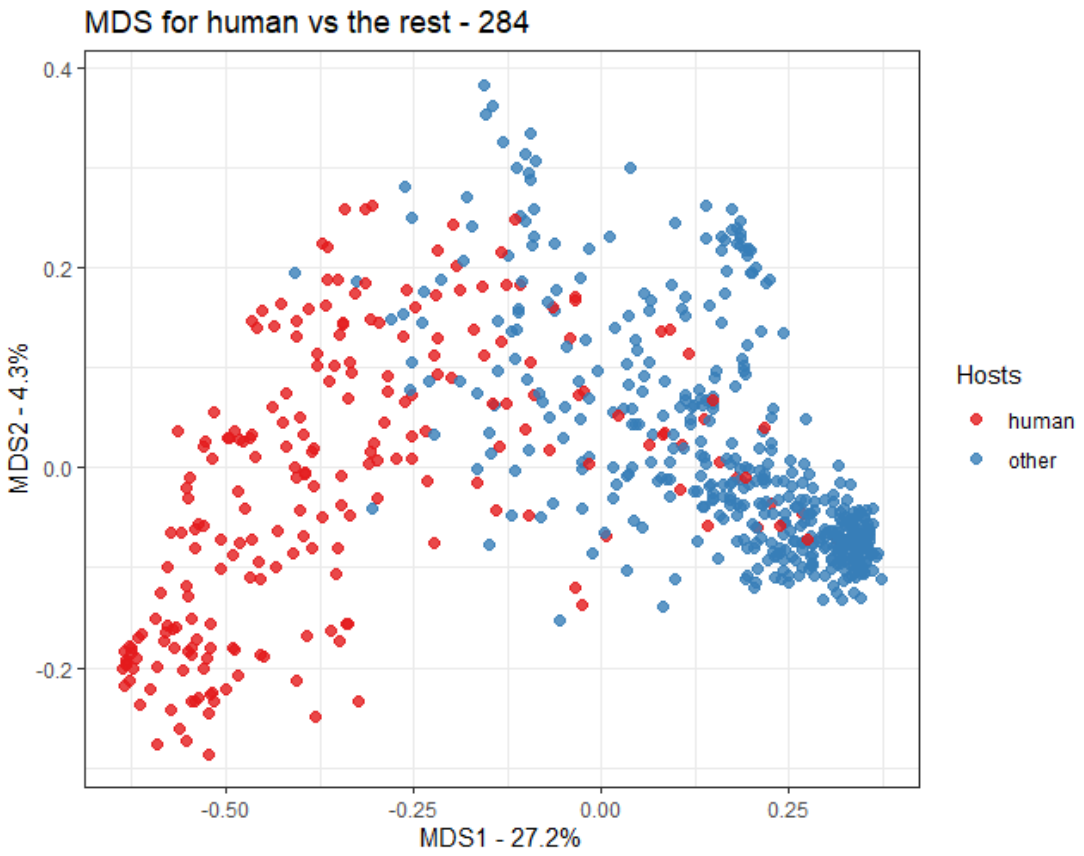
```
Call:
 randomForest(formula = host_categories ~ ., data = temp_fix,
ntree = best_trees, proximity = TRUE, mtry = min_mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 18

        OOB estimate of  error rate: 1.18%
Confusion matrix:
      fox other class.error
fox    22     8   0.2666667
other   0   650   0.0000000
>
```

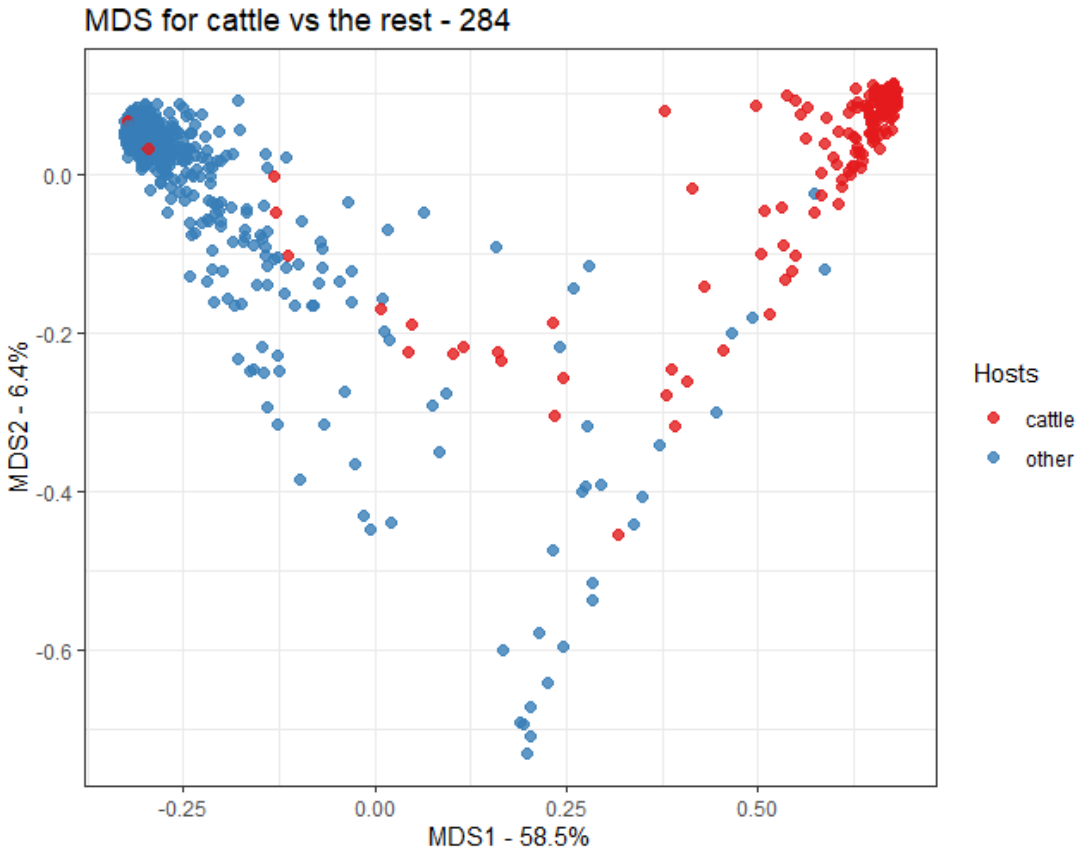
Εικόνα 81 One vs All Random Forest - Fox - tuned parameters - 284

Για την αλεπού έχει ποσοστό λάθους με προκαθορισμένες παραμέτρους ίσο με 1.32% και ορίζοντας το mtry 18 και το ntree 500 το ποσοστό πέφτει 0.44% στο 1.18%. Το ποσοστό επιτυχίας του μοντέλου είναι 98.68%.

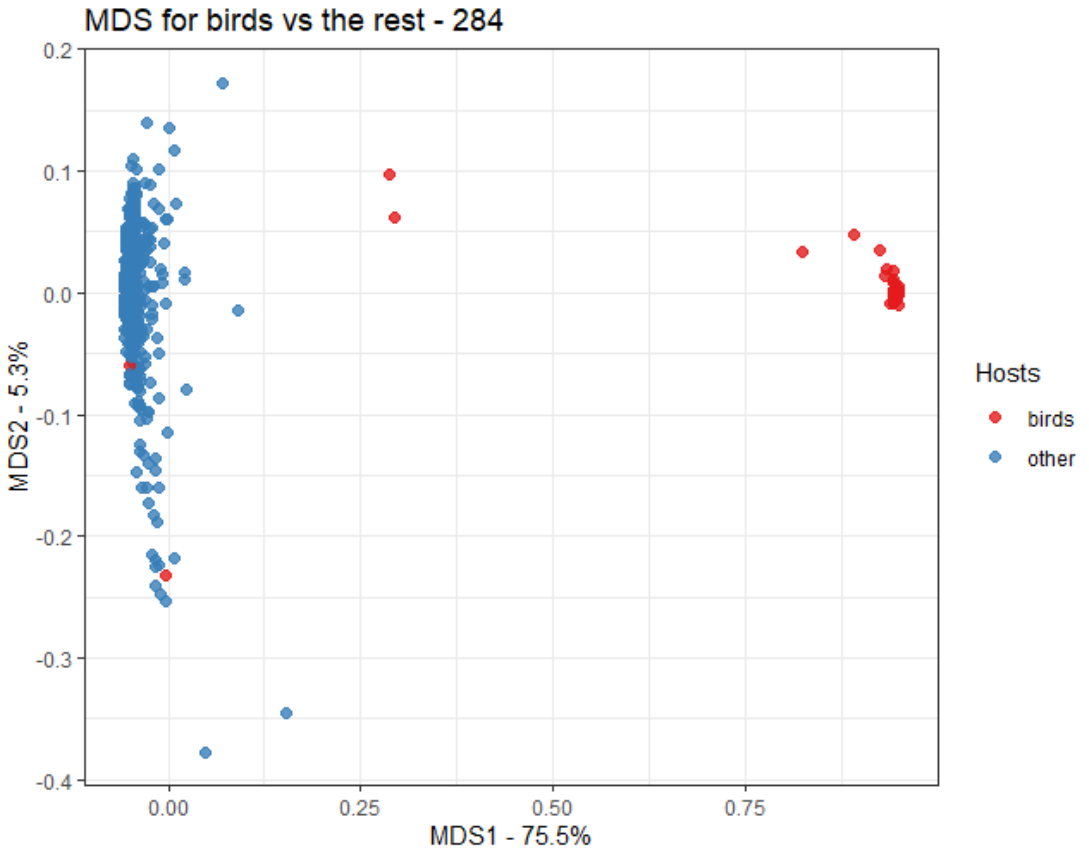
Όπως και στο Random Forest, έτσι και εδώ αναπαριστώνται οι αποστάσεις των δειγμάτων με διαγράμματα MDS για κάθε οργανισμό ενάντια στους υπολοίπους. Στα διαγράμματα αναγράφεται το MDS1 και MDS2 και τα ποσοστά τους και ο κάθε οργανισμός χρωματίζεται με κόκκινο ενώ οι υπόλοιποι με μπλε. Σημείωση πως για το γουρούνι (Εικόνα 85) τα χρώματα είναι ανάποδα.



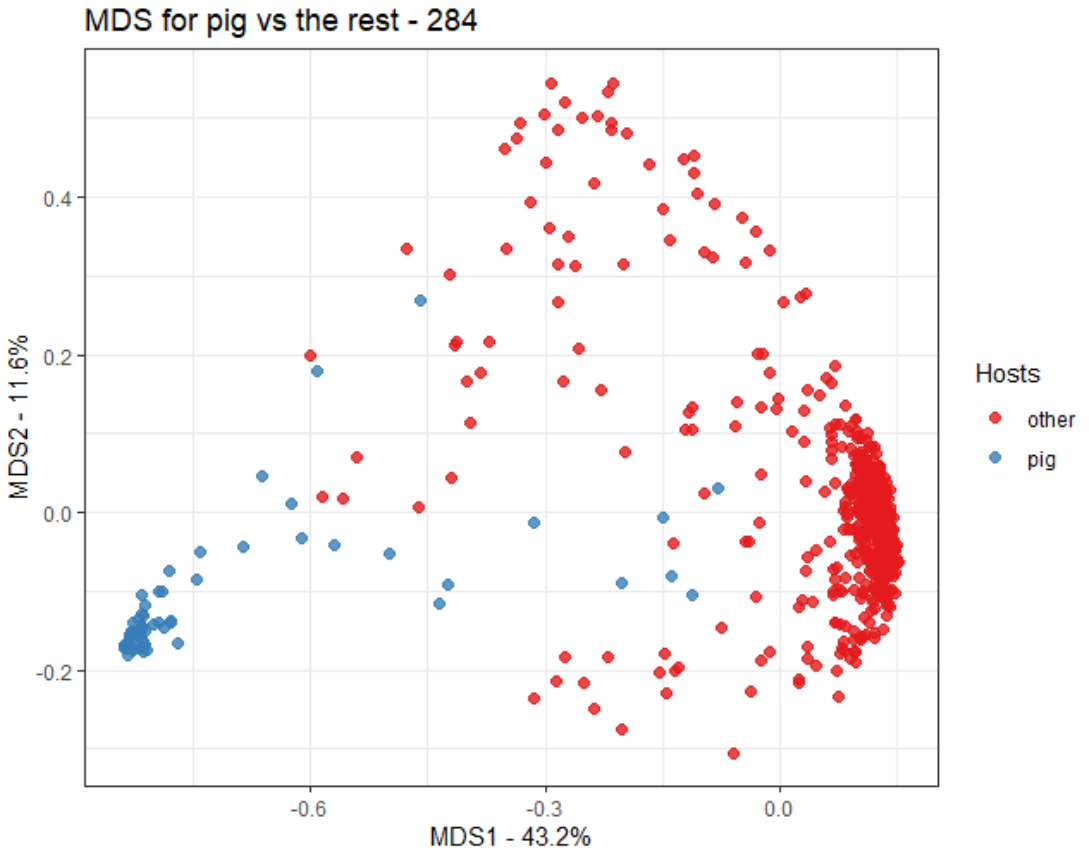
Εικόνα 82 MDS OvA Human - 284



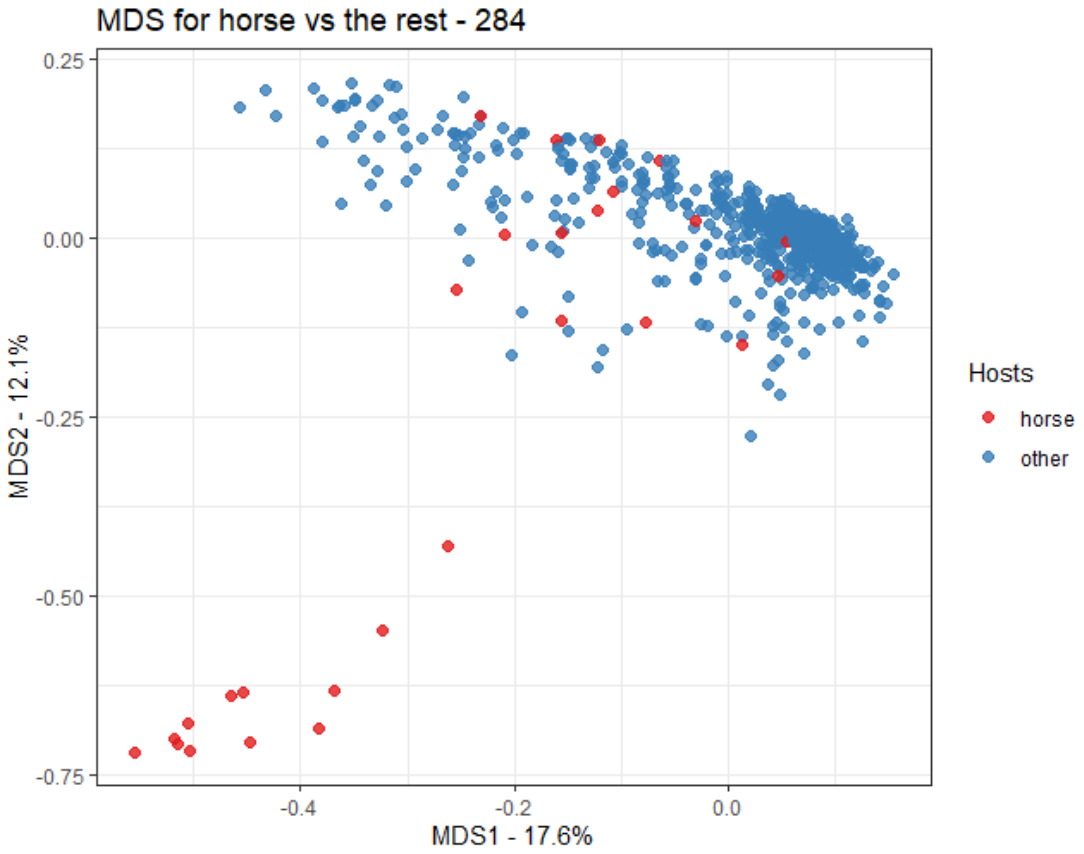
Εικόνα 83 MDS OvA Cattle - 284



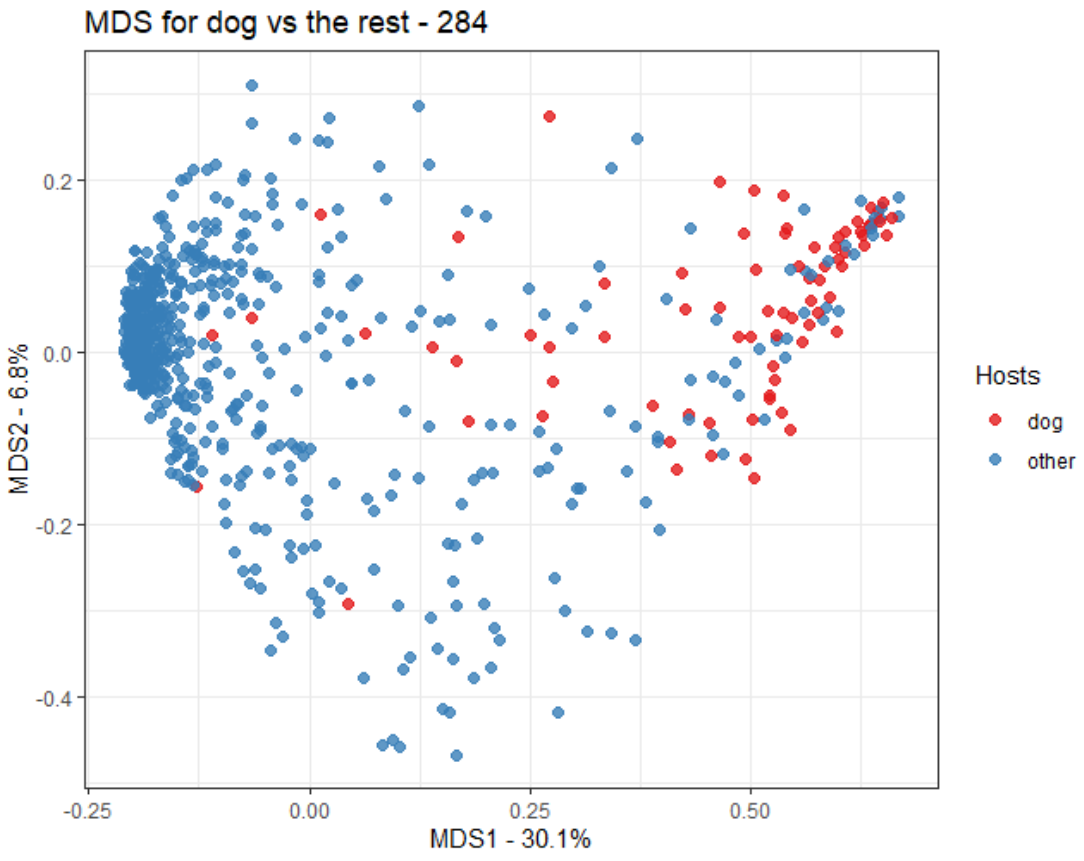
Εικόνα 84 MDS OvA Birds - 284



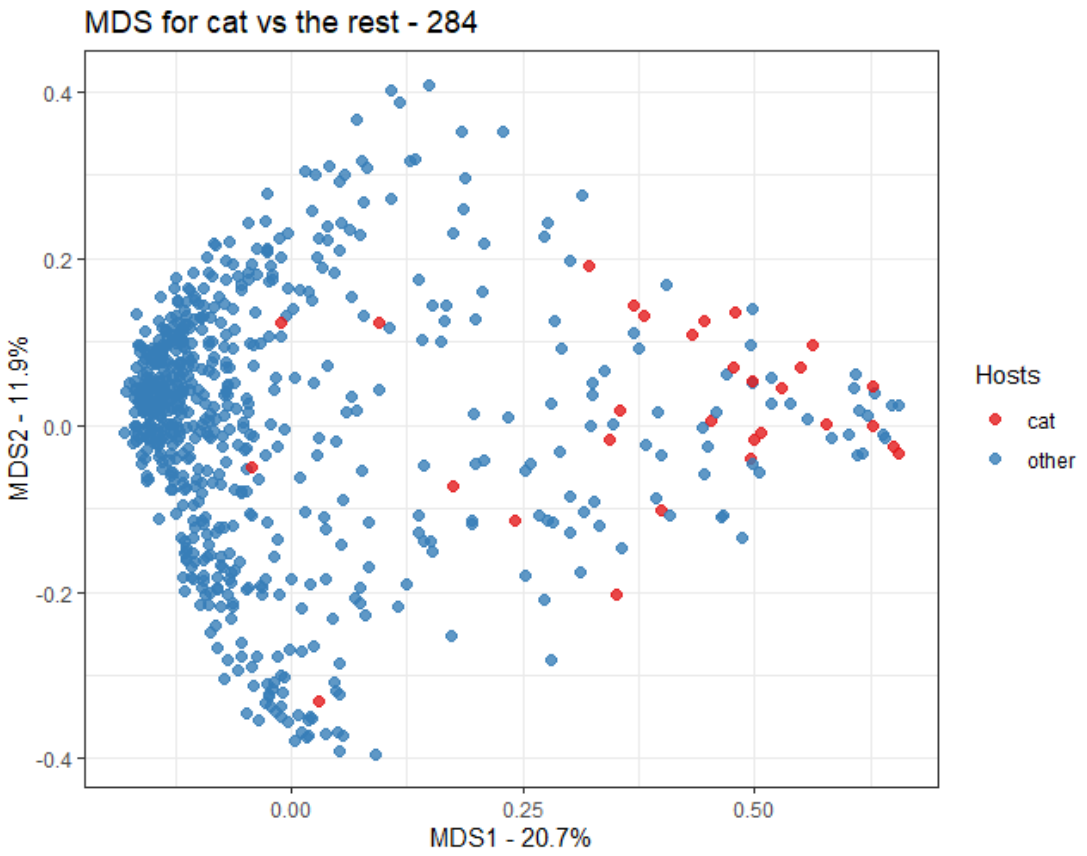
Εικόνα 85 MDS OvA Pig - 284



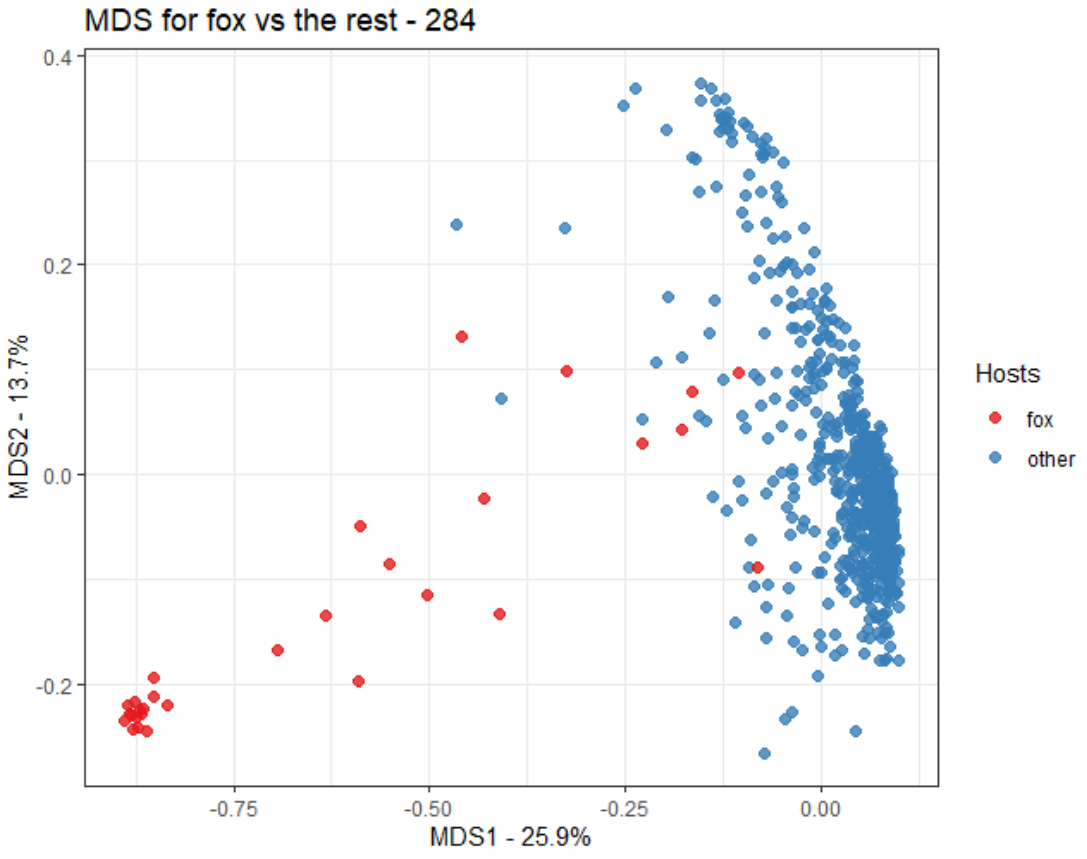
Εικόνα 86 MDS OvA Horse - 284



Εικόνα 87 MDS OvA Dog - 284



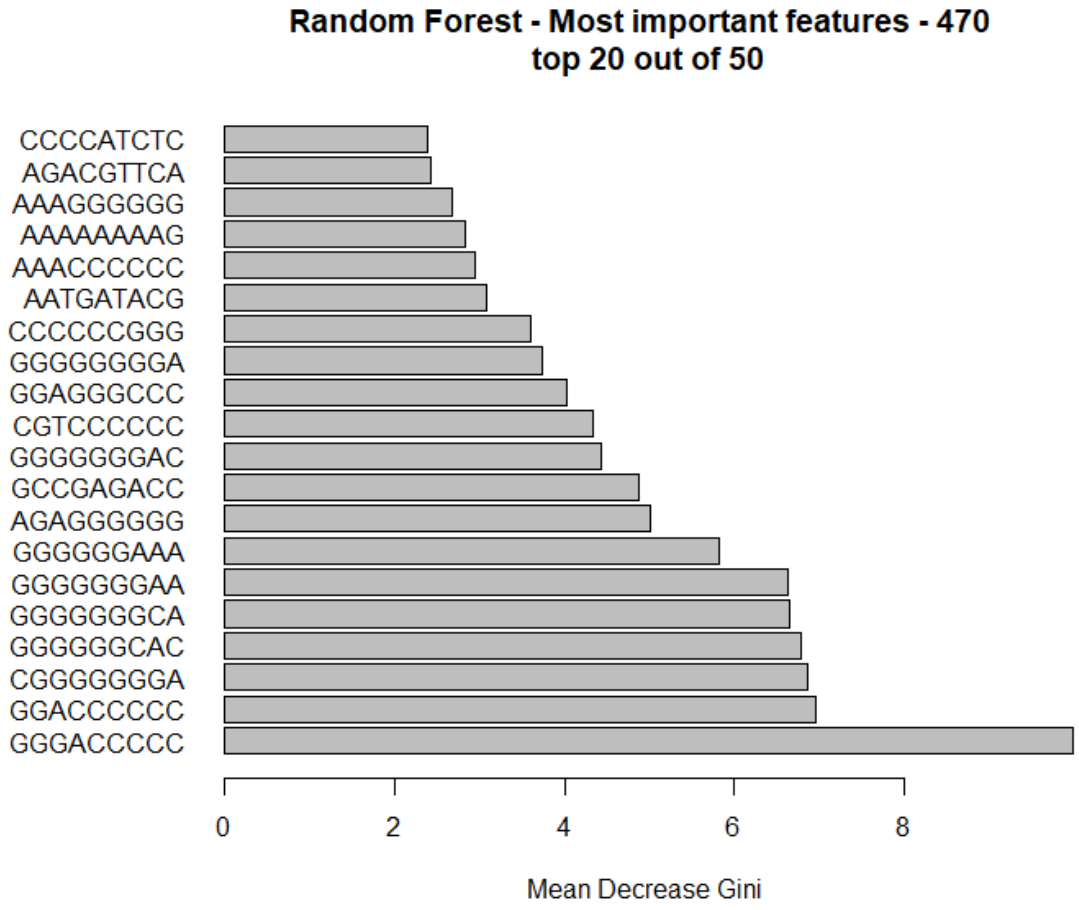
Εικόνα 88 MDS OvA Cat - 284



Εικόνα 89 MDS OvA Fox - 284

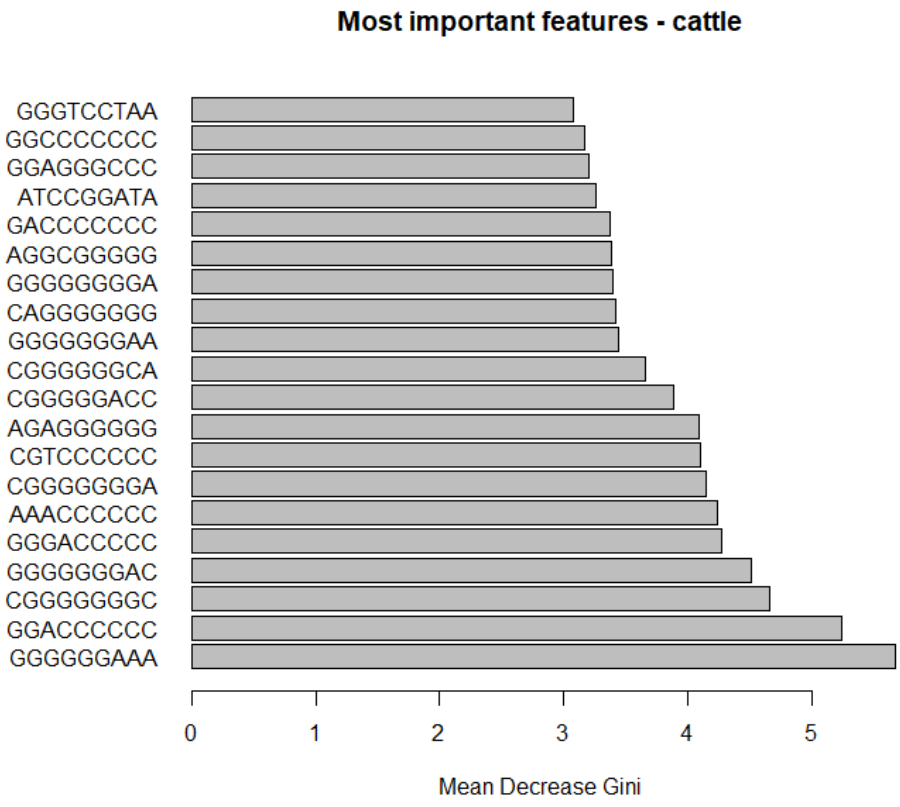
4.10 Variable importance - 470

Η αναπαράσταση της κατάταξης της σημαντικότητας των χαρακτηριστικών κάθε μεθόδου και κάθε οργανισμού γίνεται με bar-plots παρακάτω. Αρχικά είναι η αναπαράσταση των πιο σημαντικών κμερς της μεθόδου Random Forest, τα πρώτα 20 κμερς από τα 50. Στη συνέχεια βρίσκεται η συλλογή των 20 πιο σημαντικών kmers για κάθε οργανισμό βάση της μεθόδου One vs All.

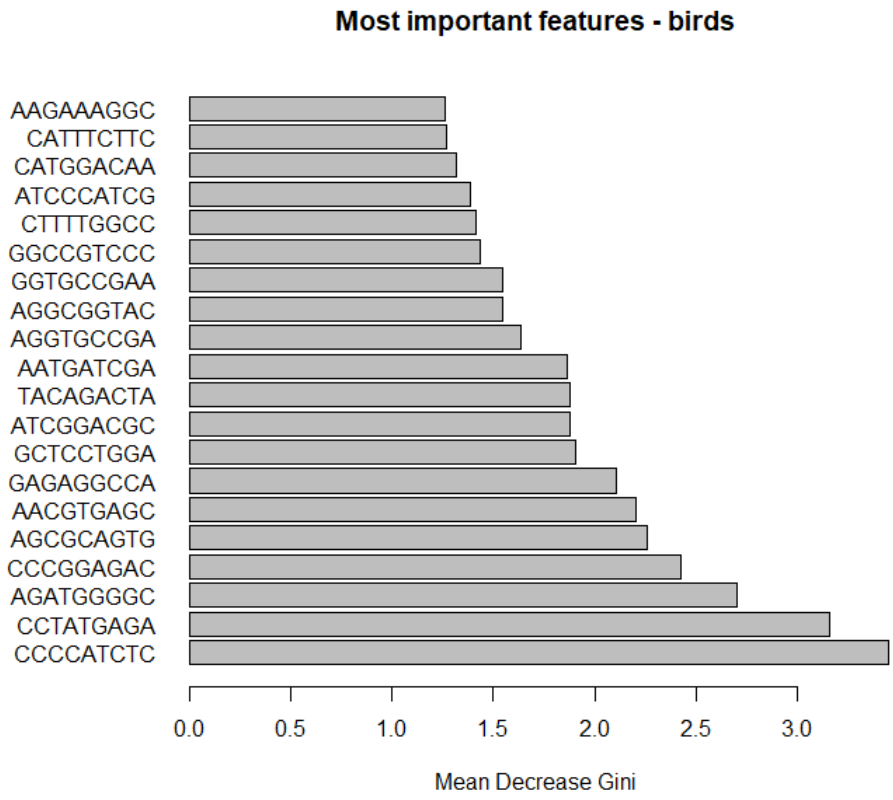


Εικόνα 90 Random Forest most important variables - 470

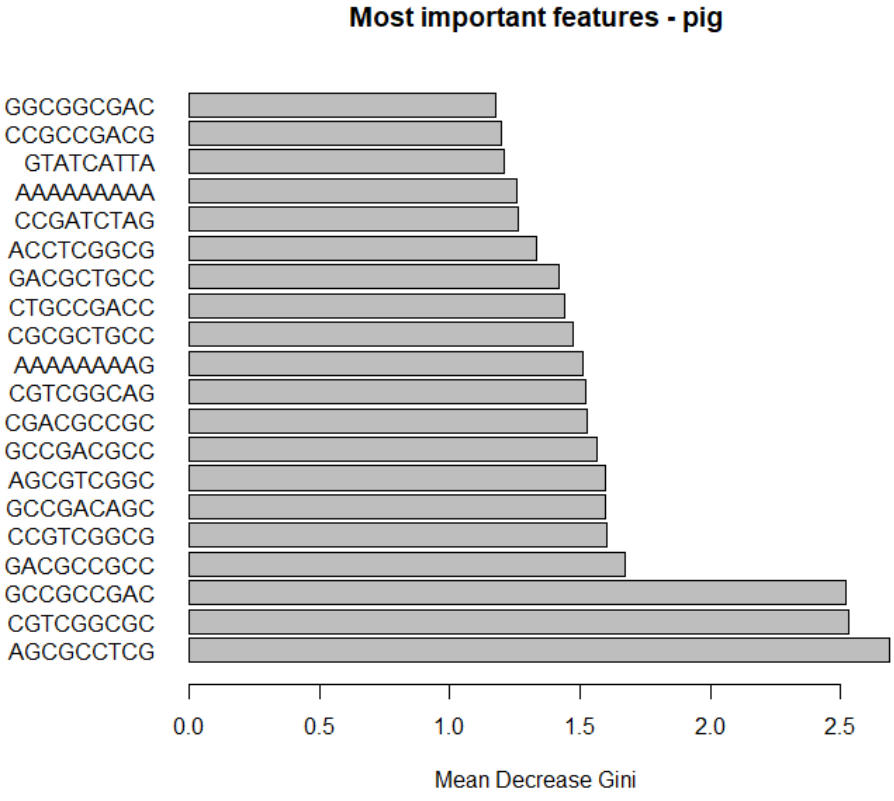
Εικόνα 91 OvA - Top 20 most important k-mers – Human – 470



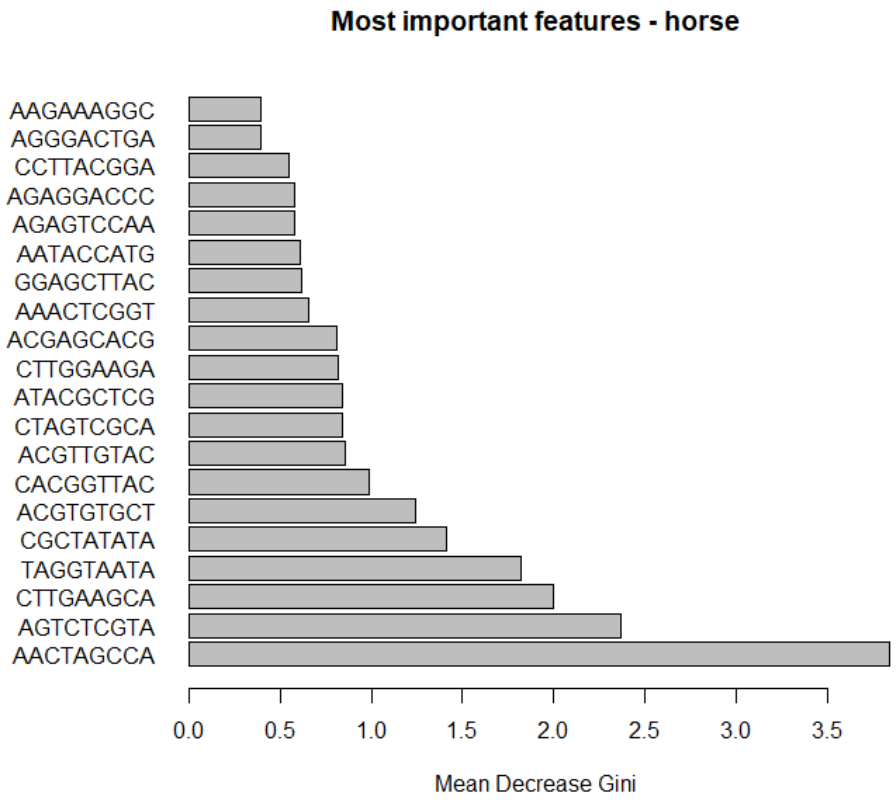
Εικόνα 92 Top 20 most important k-mers – Cattle – 470



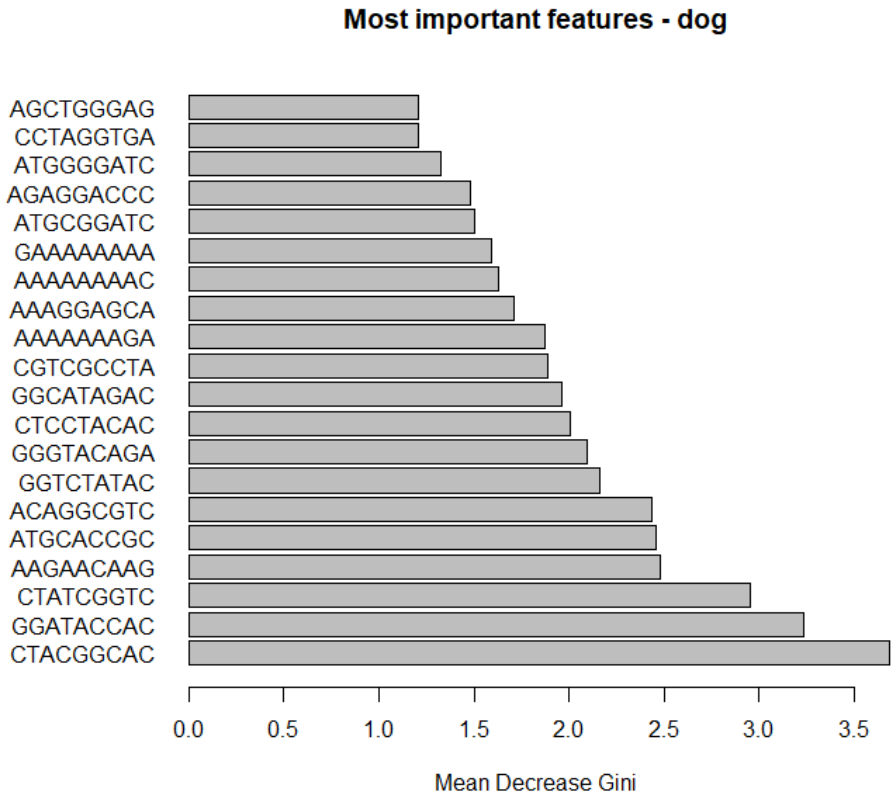
Εικόνα 93 Top 20 most important k-mers – Birds – 470



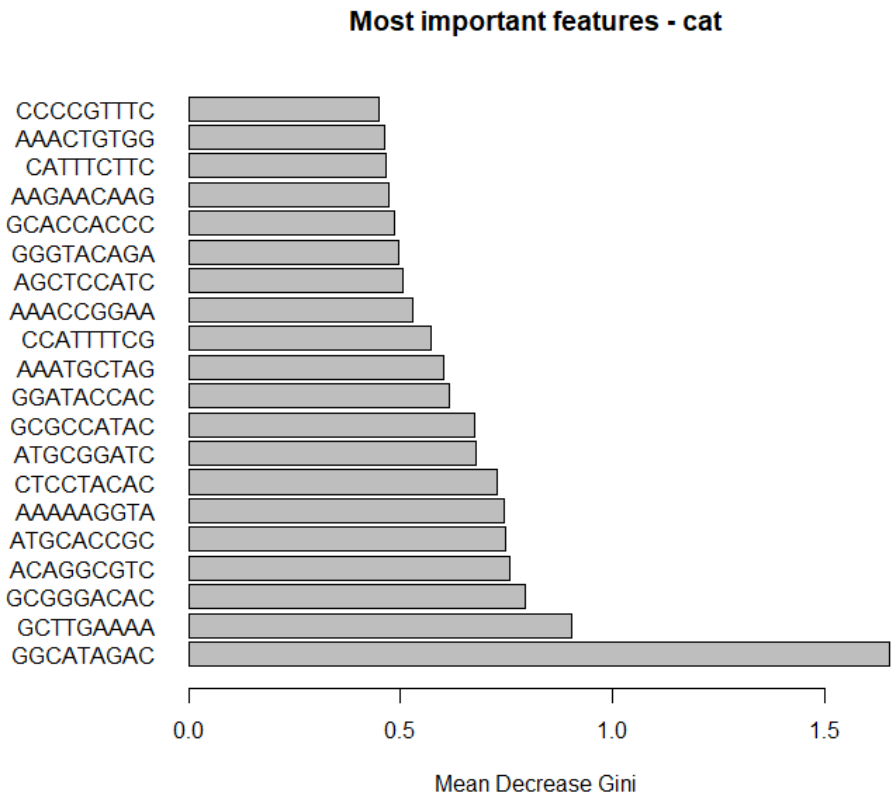
Εικόνα 94 Top 20 most important k-mers – Pig – 470



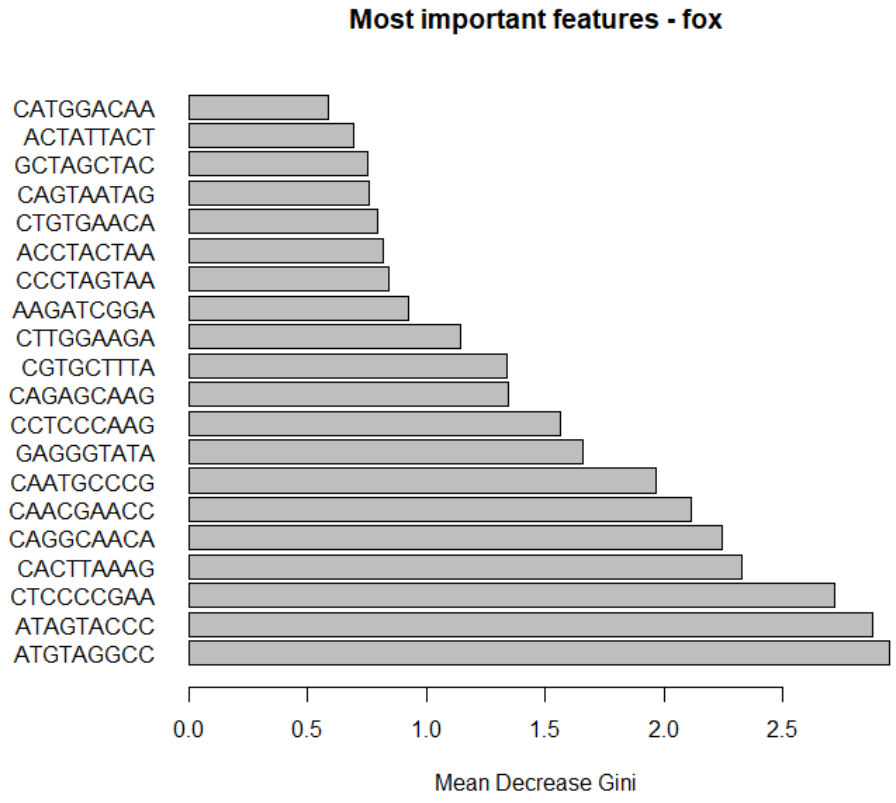
Εικόνα 95 Top 20 most important k-mers – Horse – 470



Εικόνα 96 Top 20 most important k-mers – Dog – 470



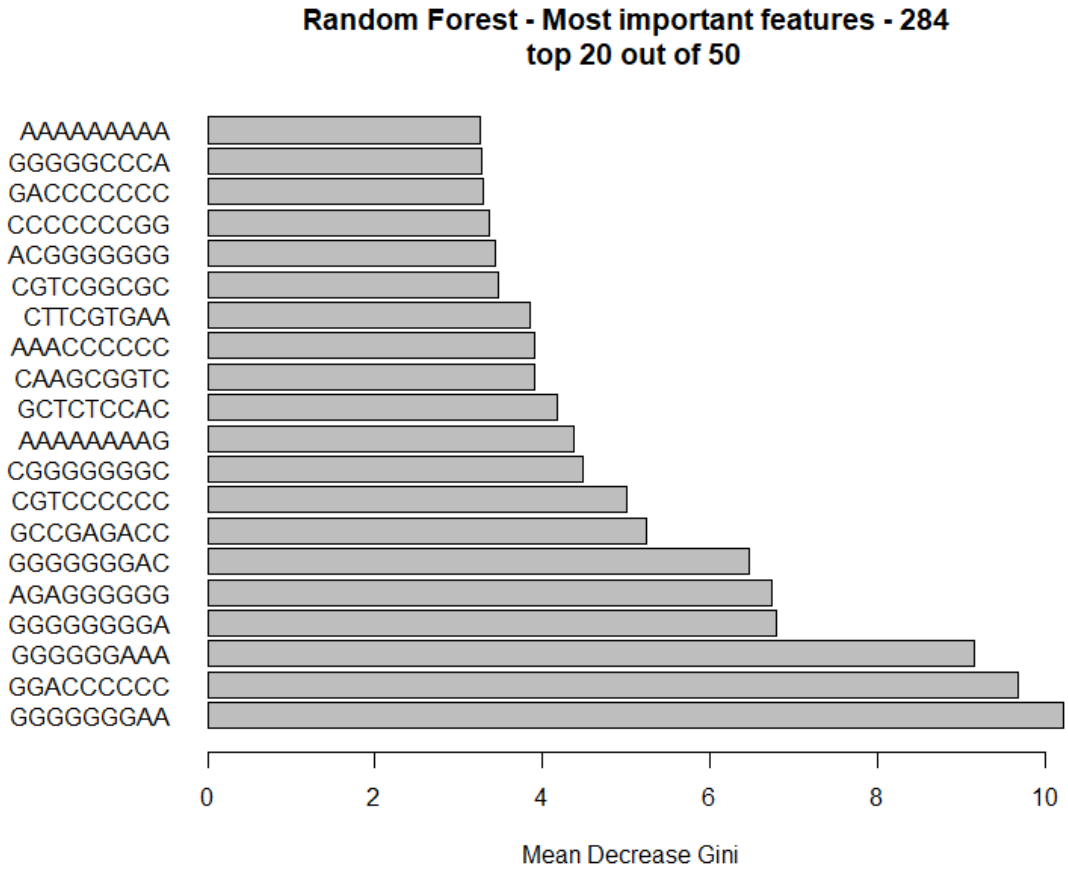
Εικόνα 97 Top 20 most important k-mers – Cat – 470



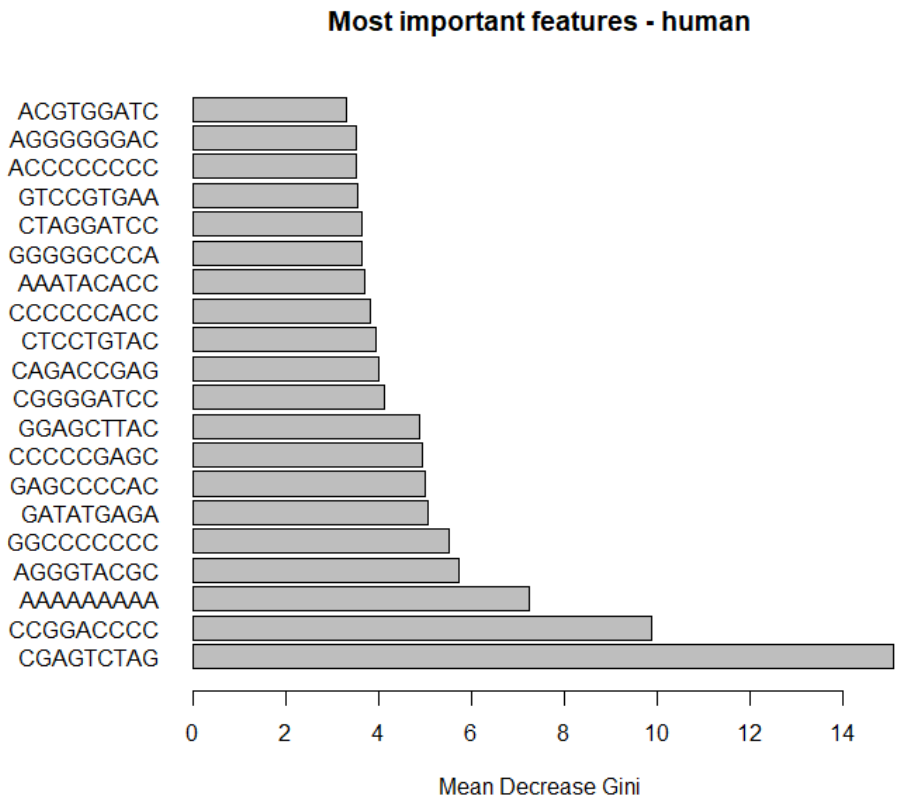
Εικόνα 98 Top 20 most important k-mers – Fox – 470

4.11 Variable importance - 284

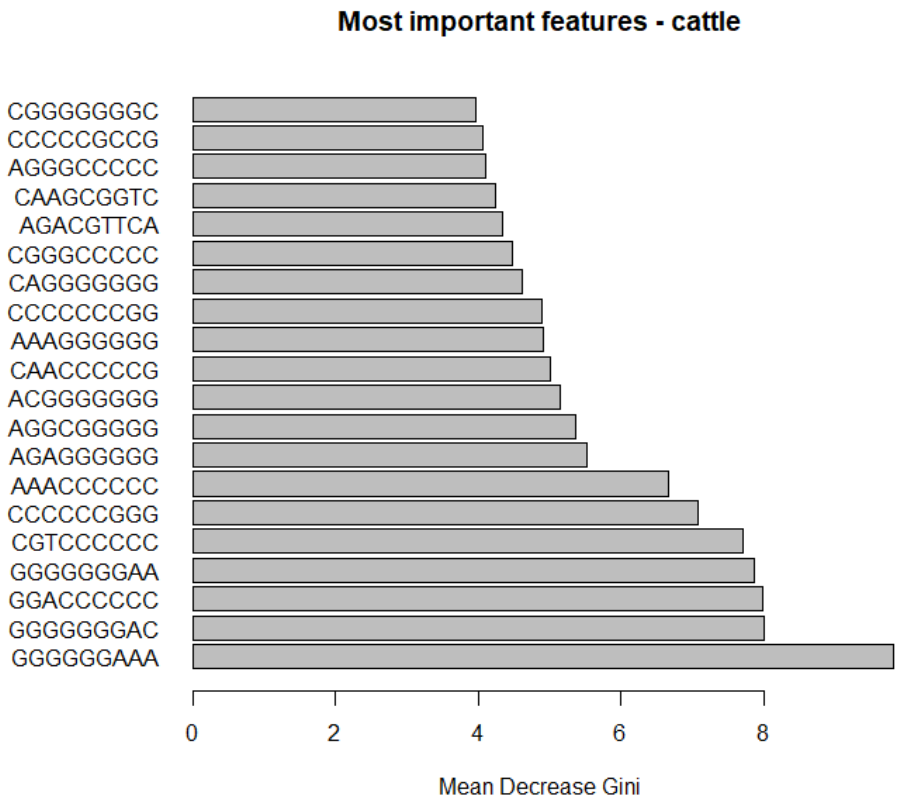
Όπως και στο σύνολο των 470, έτσι και εδώ αναπαρίστανται τα διαγράμματα, πρώτα του Random Forest και μετά οι οργανισμοί του OvA.



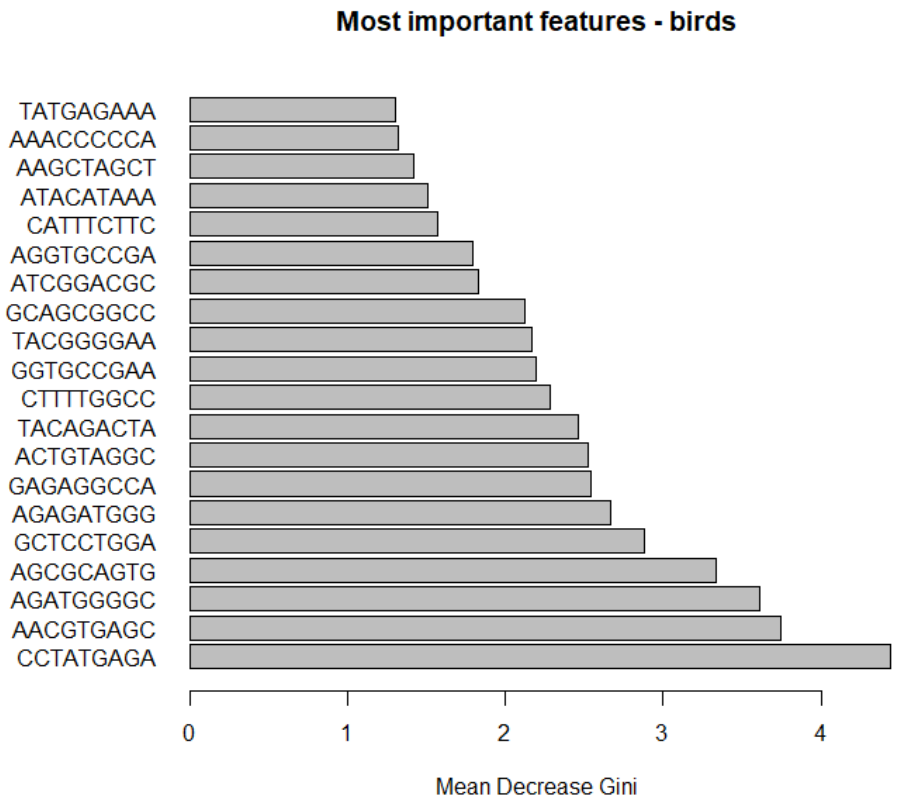
Εικόνα 99 Random Forest most important variables - 284



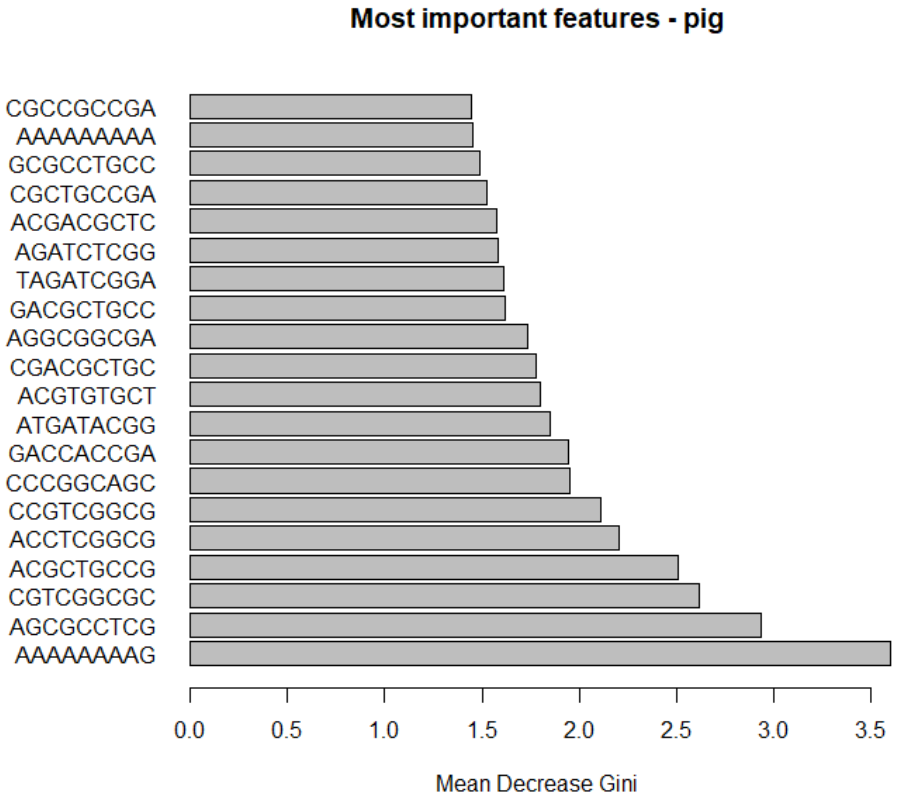
Εικόνα 100 Top 20 most important k-mers – Human – 284



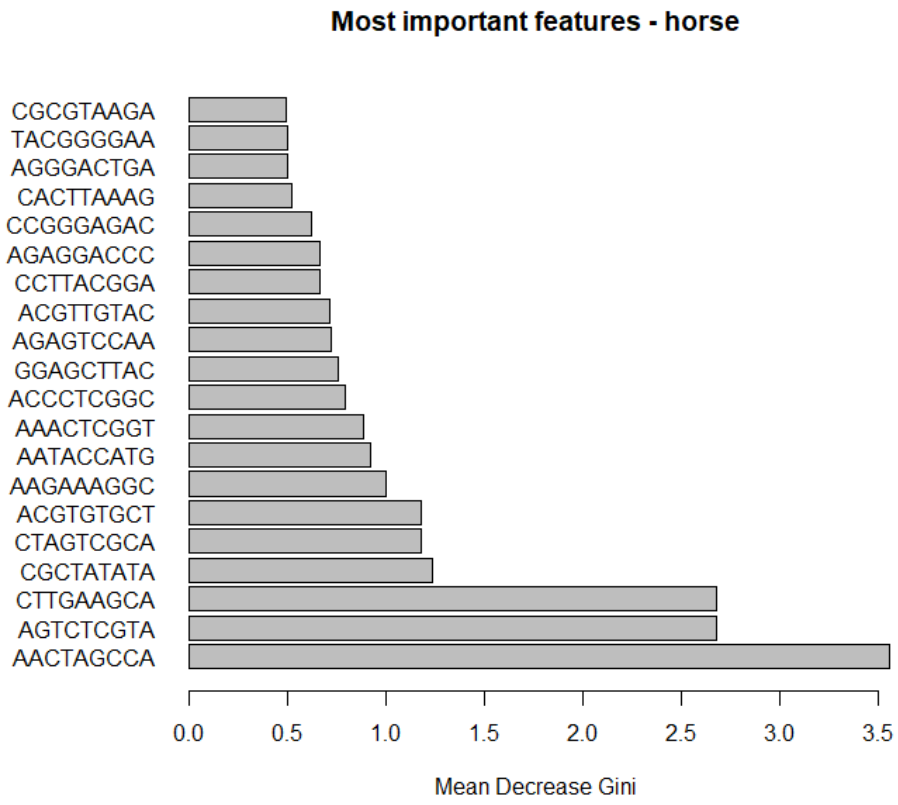
Εικόνα 101 Top 20 most important k-mers – Cattle – 284



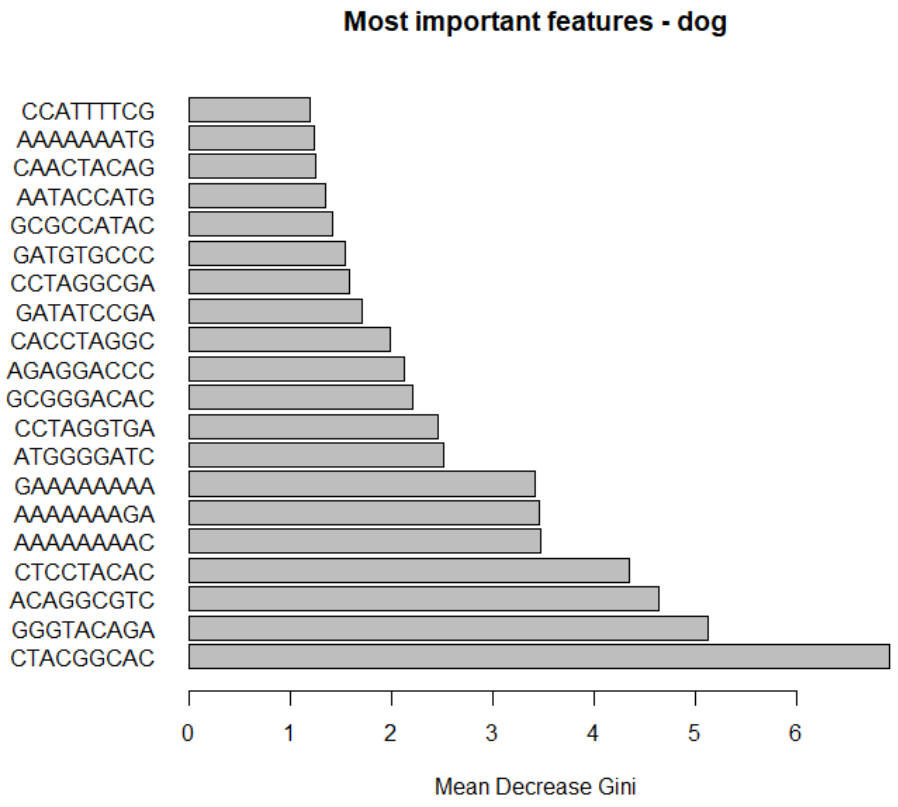
Εικόνα 102 Top 20 most important k-mers – Birds – 284



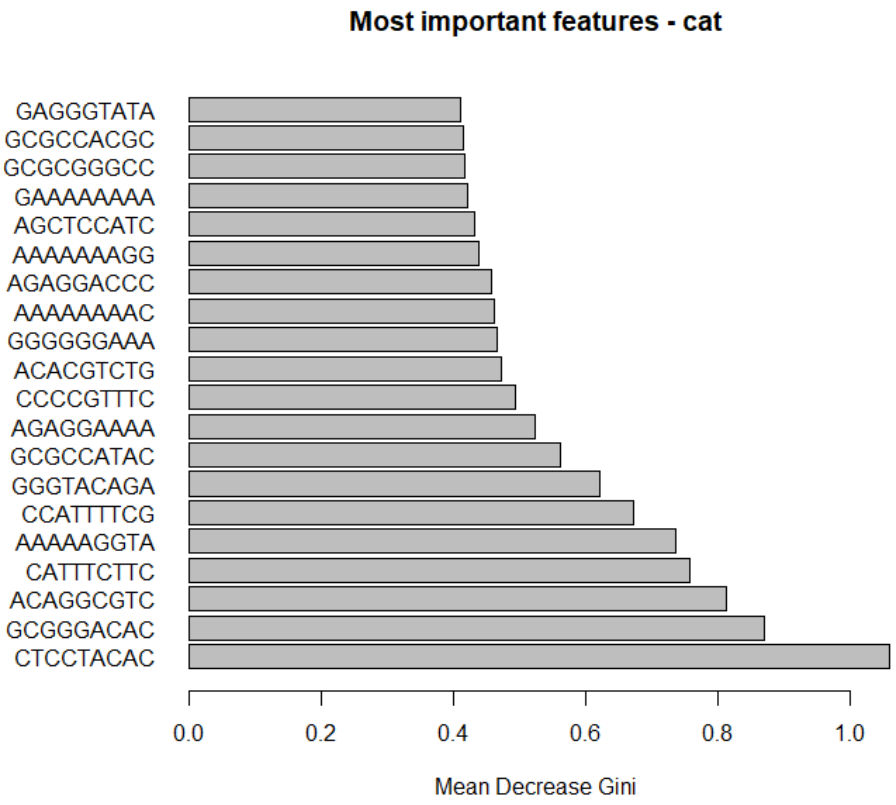
Εικόνα 103 Top 20 most important k-mers – Pig – 284



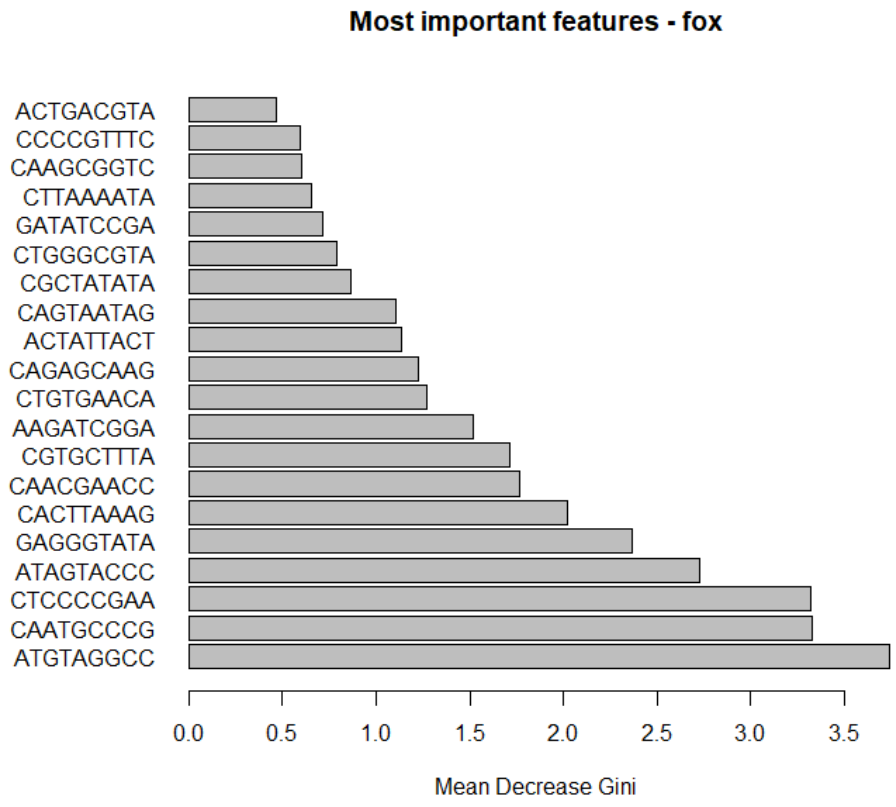
Εικόνα 104 Top 20 most important k-mers – Horse – 284



Εικόνα 105 Top 20 most important k-mers – Dog – 284



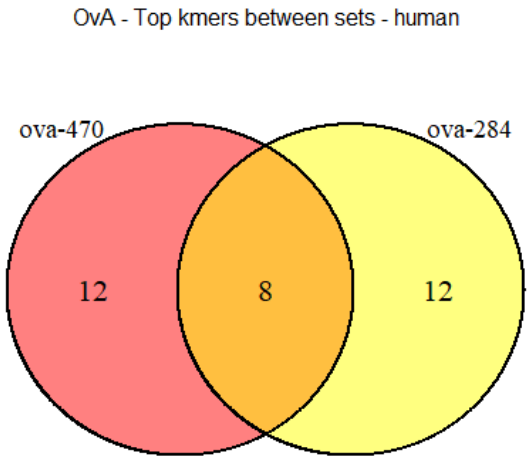
Εικόνα 106 Top 20 most important k-mers – Cat – 284



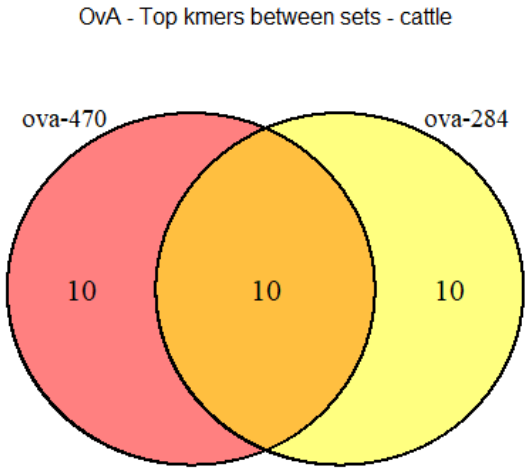
Εικόνα 107 Top 20 most important k-mers – Fox – 284

4.12 One vs All - τομή συνόλων 470-284

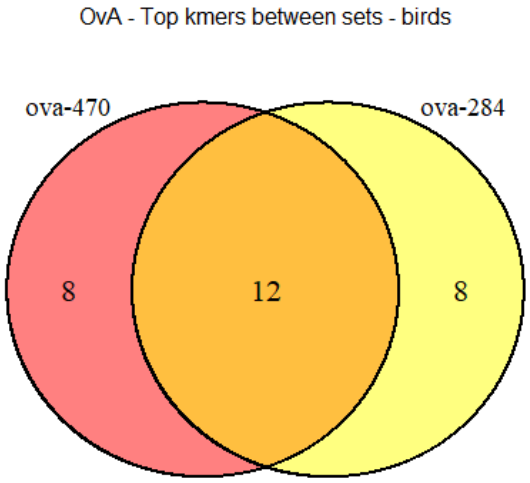
Η σύγκριση των πιο σημαντικών κμερς που έβγαλε το σύνολο δεδομένων των 470 με το σύνολο των 284 χρησιμοποιώντας τη μέθοδο One vs All έβγαλε κοινά k-mers σε κάθε οργανισμό. Οι παρακάτω εικόνες δείχνουν τα κμερς ως διαγράμματα Venn και ονομαστικά στο τέλος.



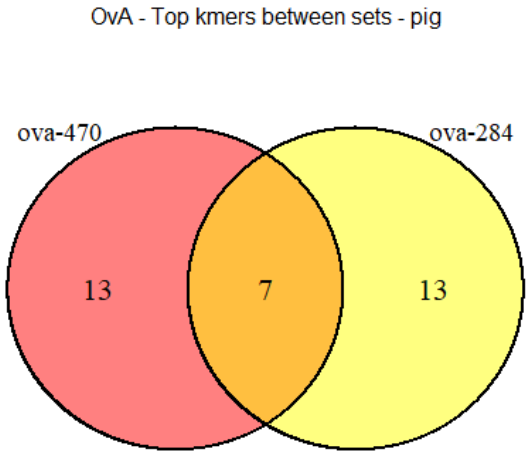
Εικόνα 108 Κοινά k-mers μεταξύ συνόλων - Human



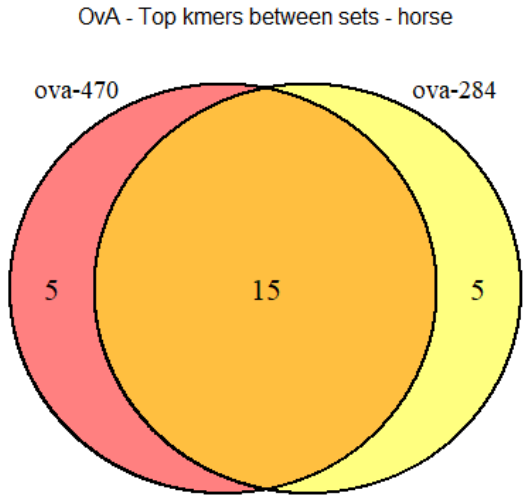
Εικόνα 109 Κοινά k-mers μεταξύ συνόλων - Cattle



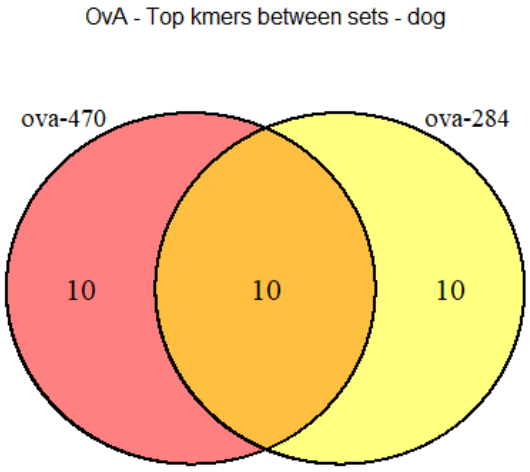
Εικόνα 110 Κοινά k-mers μεταξύ συνόλων - Birds



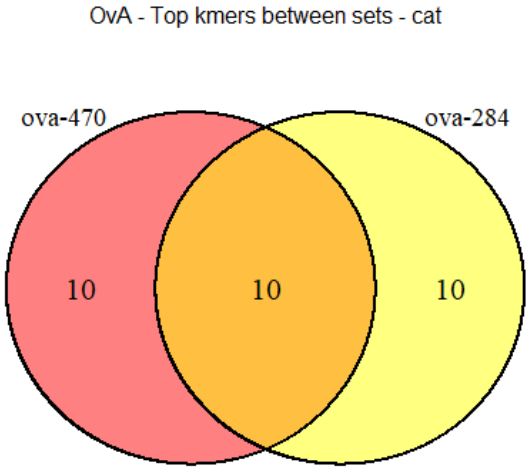
Εικόνα 111 Κοινά k-mers μεταξύ συνόλων - Pig



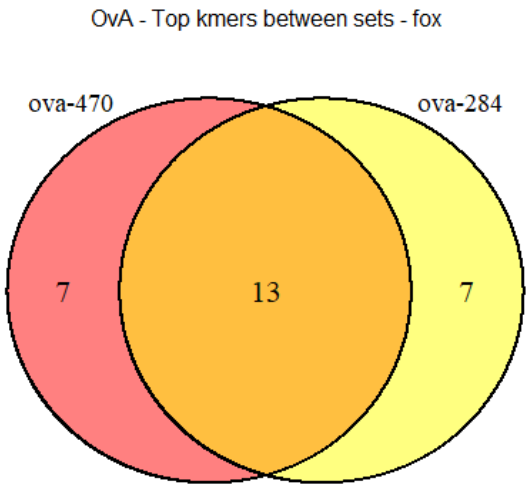
Εικόνα 112 Κοινά k-mers μεταξύ συνόλων - Horse



Εικόνα 113 Κοινά k-mers μεταξύ συνόλων - Dog

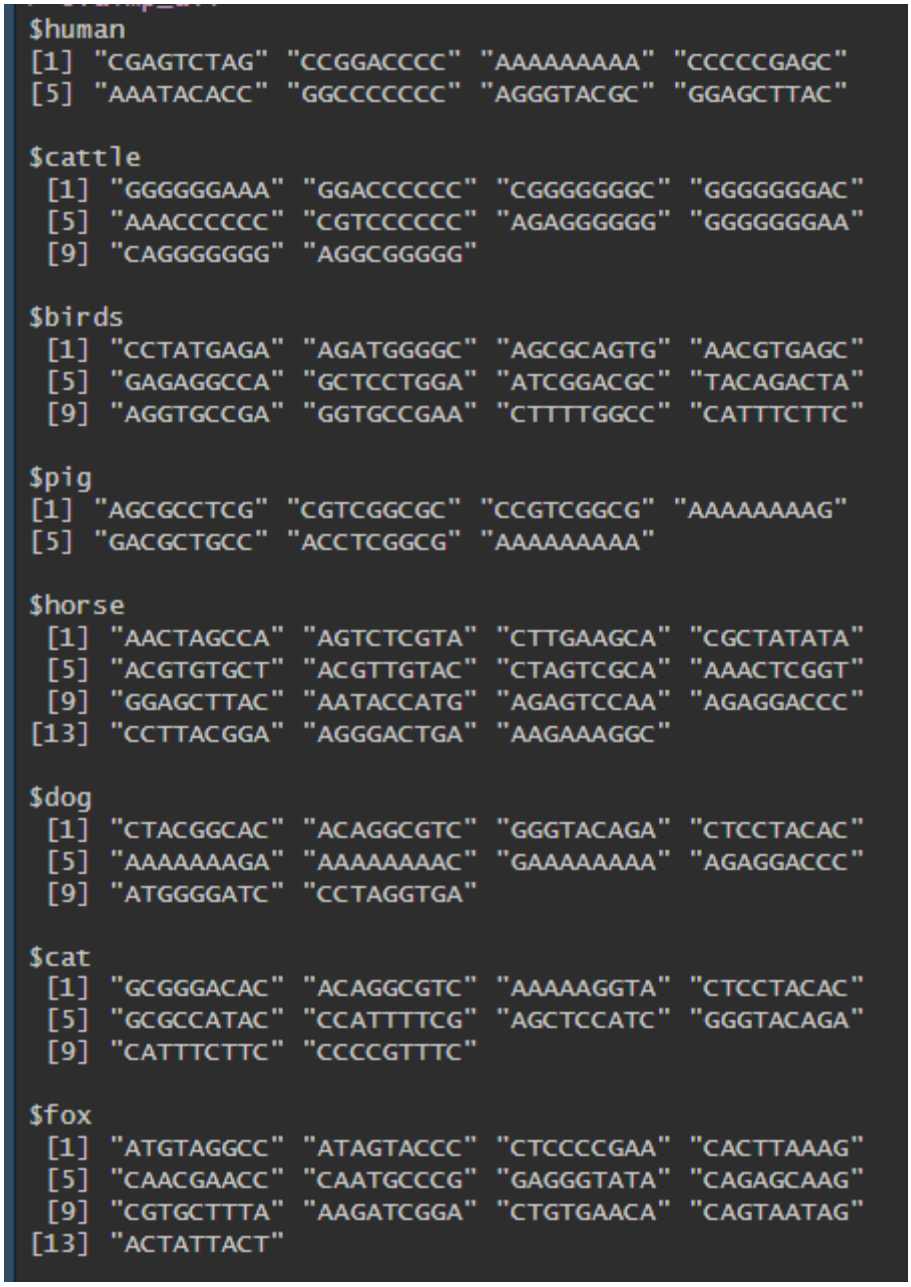


Εικόνα 114 Κοινά k-mers μεταξύ συνόλων - Cat



Εικόνα 115 Κοινά k-mers μεταξύ συνόλων - Fox

Κάθε σύνολο δεδομένων έχει 20 κμερς για κάθε οργανισμό, με το σύνολο των 470 να είναι με κόκκινο και των 284 με κίτρινο. Στο κέντρο αναγράφεται ο αριθμός των κοινών κμερς μεταξύ των συνόλων. Τα κοινά αυτά k-mers είναι τα παρακάτω.



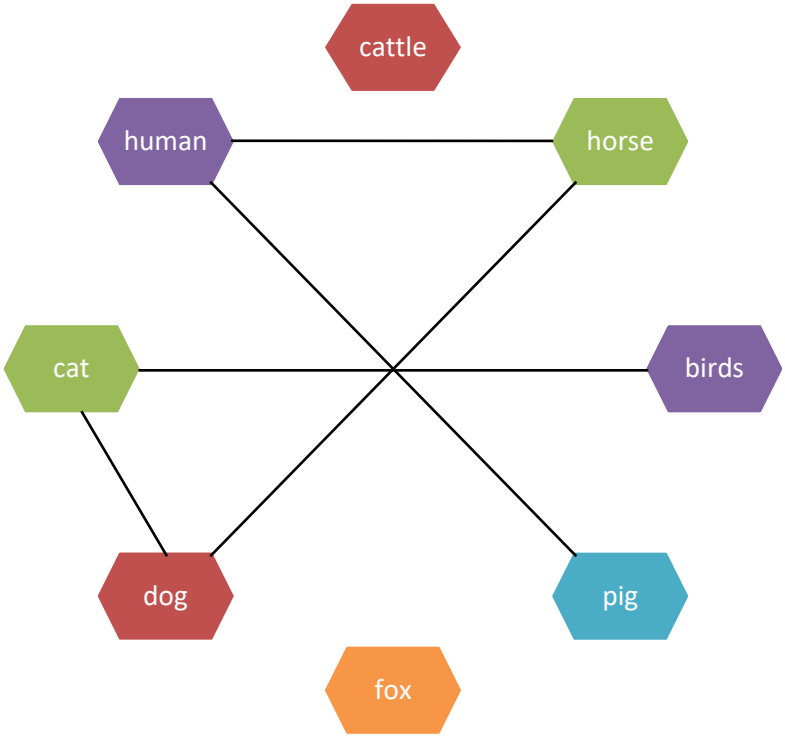
Εικόνα 116 Κοινά k-mers μεταξύ 470-284 για κάθε οργανισμό

Γίνεται ένας έλεγχος μεταξύ των οργανισμών για παρουσία κοινών k-mers. Τα κμερς αυτά θα αφαιρεθούν γιατί αναζητούνται κμερς μοναδικά για κάθε οργανισμό.

```
human pig AAAAAAAAA
human horse GGAGCTTAC
birds cat CATTTCTTC
horse dog AGAGGACCC
dog cat ACAGGCGTC GGGTACAGA CTCCTACAC
```

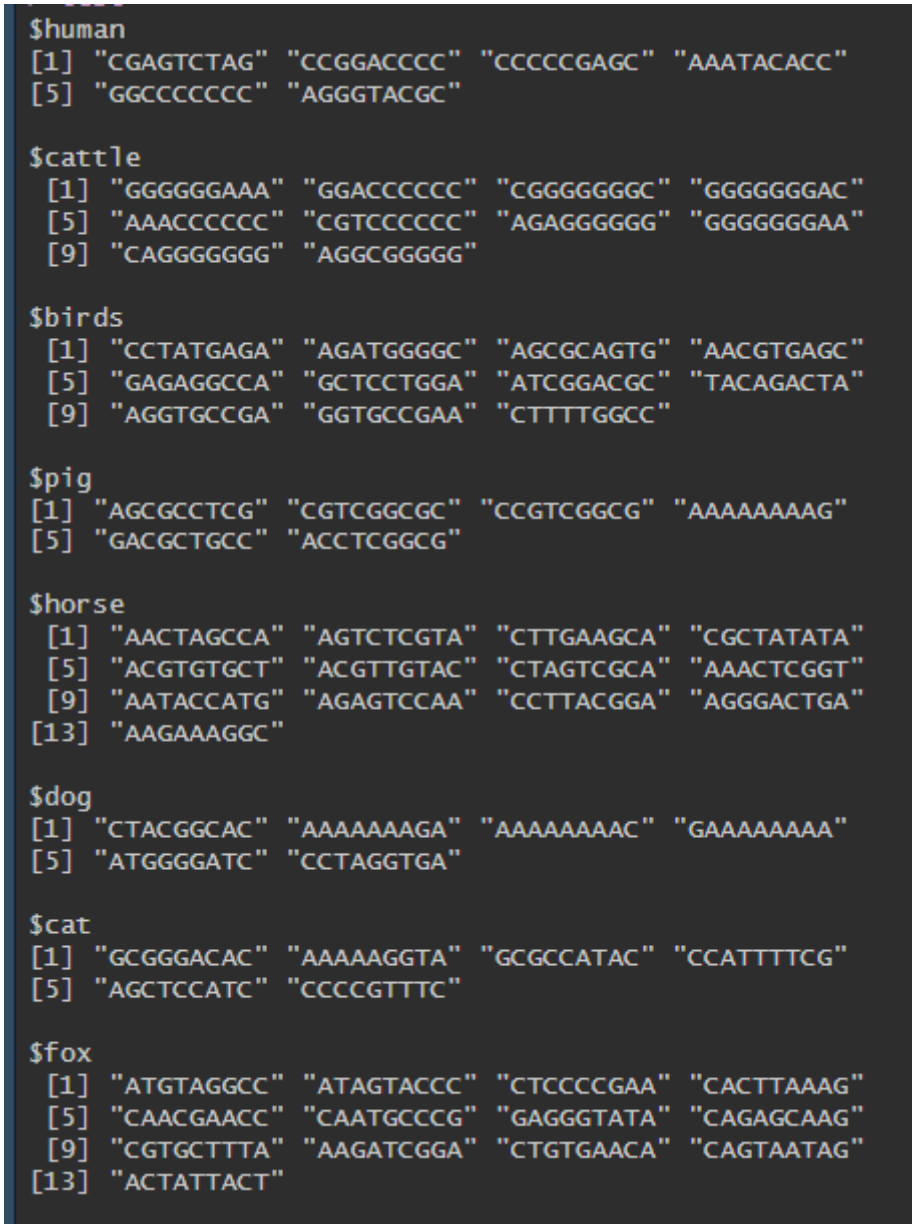
Εικόνα 117 Ίδια k-mers μεταξύ οργανισμών

Βρέθηκαν 5 αλληλεπιδράσεις μεταξύ ζευγών οργανισμών. Οι περισσότερες έχουν από ένα k-mer ενώ η αλληλεπίδραση του σκύλου και της γάτας έχει 3 κοινά k-mers. Οι αλληλεπιδράσεις αναπαριστώνται και ως διάγραμμα παρακάτω.



Εικόνα 118 Αναπαράσταση αλληλεπιδράσεων οργανισμών

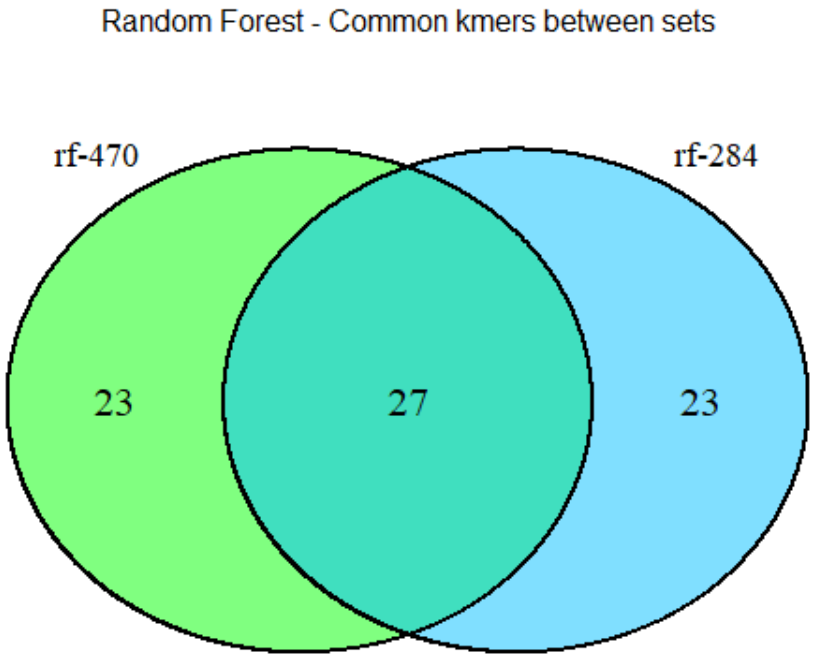
Έτσι πλέον ο άνθρωπος, το γουρούνι, ο σκύλος και η γάτα έχουν 6 k-mers, τα πουλιά 11 και το άλογο 13 και τα βοοειδή και η αλεπού παραμένουν στα 10 και 13 αντίστοιχα.



Εικόνα 119 Μοναδικά k-mers για κάθε οργανισμό

4.13 Random Forest - τομή συνόλων 470-284

Συνεχίζοντας γίνεται η ίδια ανάλυση για τα πιο σημαντικά κμερς που συλλέχθηκαν από τον Random Forest και το διάγραμμα είναι το παρακάτω.



Εικόνα 120 Κοινά k-mers μεταξύ συνόλων με Random Forest

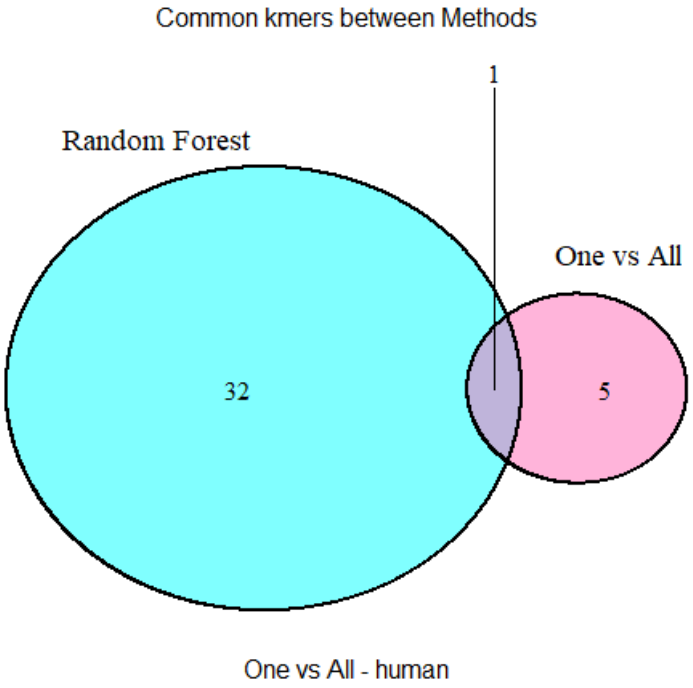
Τα σύνολα των 470 και των 284 έχουν 27 κοινά κμερς από τα 50 που είχαν συλλεχθεί με τη μέθοδο Random Forest. Ο Random Forest δεν ξεχωρίζει τους οργανισμούς γι' αυτό είναι ένα το διάγραμμα Venn. Παρακάτω αναγράφονται τα κοινά αυτά κμερς.

```
 [1] "GGACCCCCC" "GGGGGGGAA" "GGGGGGAAA" "AGAGGGGGG" "GCCGAGACC"
 [6] "GGGGGGGAC" "CGTCCCCCC" "GGGGGGGGA" "CCCCCCGGG" "AAACCCCCC"
[11] "AAAAAAAAG" "AAAGGGGGG" "AGACGTTCA" "GACCCCCCC" "GCTCTCCAC"
[16] "CTTCGTGAA" "CGGGCCCCC" "CGGGGGGGC" "AAAAAAAAA" "CAAGCGGTC"
[21] "ACCCCCCGG" "GGGGCCCCC" "GGGTACAGA" "CCCCCCCGG" "ACGCTGCCG"
[26] "AAGAGCACA" "GGGTCCTAA"
```

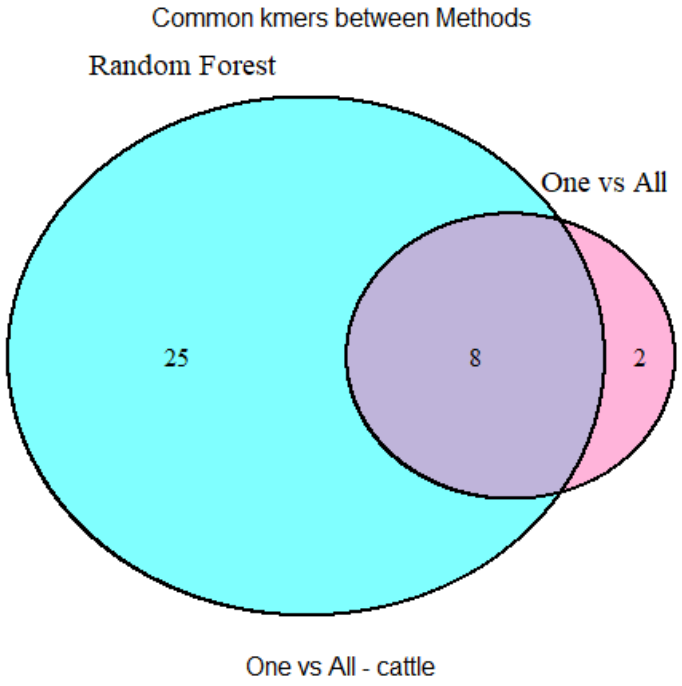
Εικόνα 121 Μοναδικά k-mers μεταξύ συνόλων με Random Forest

4.14 Τομή μεθόδων RF - OvA

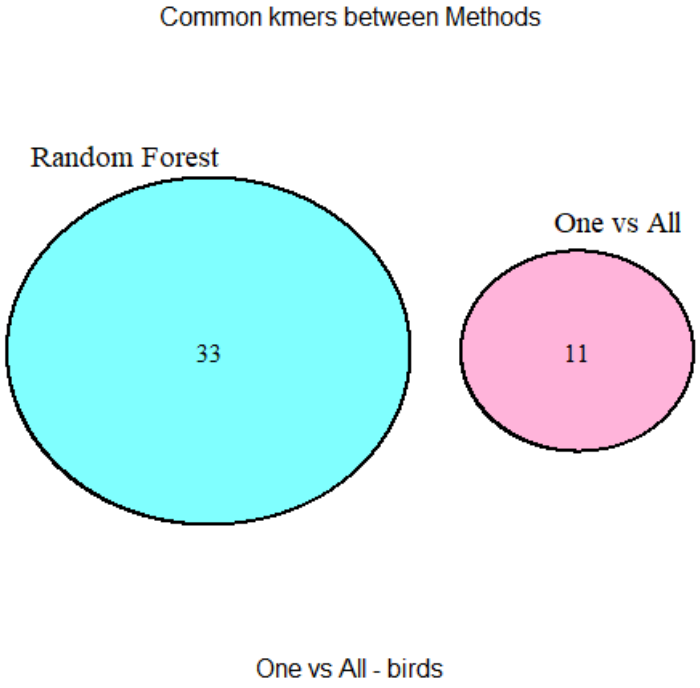
Ως τελική σύγκριση γίνεται μία μεταξύ των μεθόδων. Η τομή των κμερς που θεωρήθηκαν σημαντικά από τη μέθοδο OvA για κάθε οργανισμό και η τομή των κμερς που θεωρήθηκαν σημαντικά από τον Random Forest. Η σύγκριση γίνεται ανεξαρτήτου συνόλου δεδομένων, χρησιμοποιούνται δηλαδή τα αποτελέσματα των δύο προηγούμενων συγκρίσεων (Εικόνα 119 της μεθόδου OvA και Εικόνα 121 της μεθόδου Random Forest). Τα κοινά k-mers μεταξύ των μεθόδων για κάθε οργανισμό είναι τα εξής.



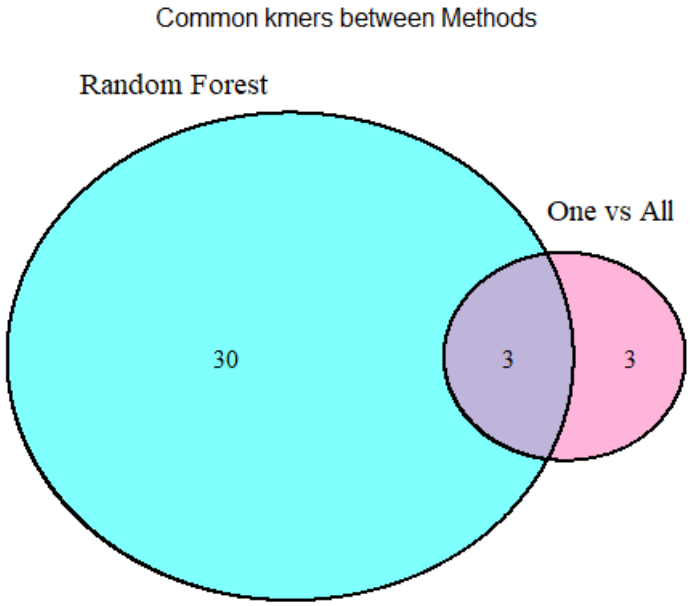
Εικόνα 122 Κοινά k-mers μεταξύ μεθόδων - Human



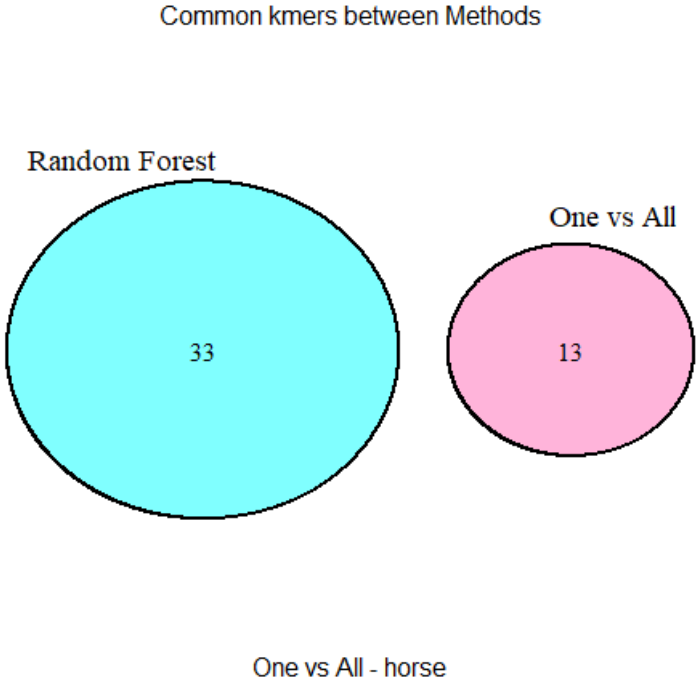
Εικόνα 123 Κοινά k-mers μεταξύ μεθόδων - Cattle



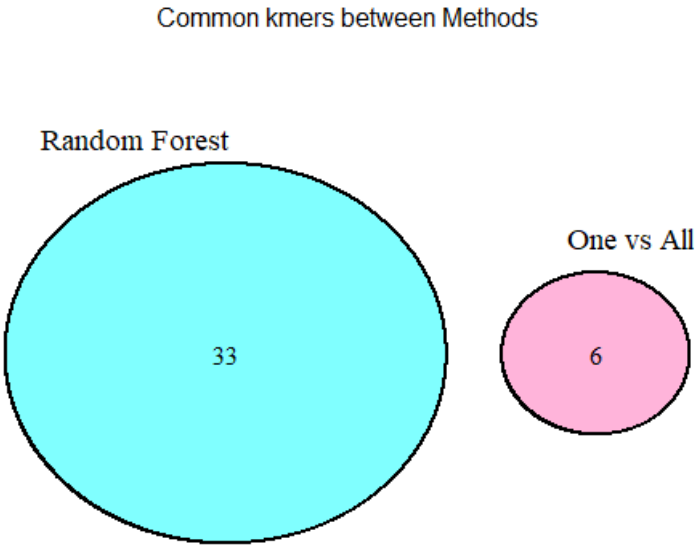
Εικόνα 124 Κοινά k-mers μεταξύ μεθόδων - Birds



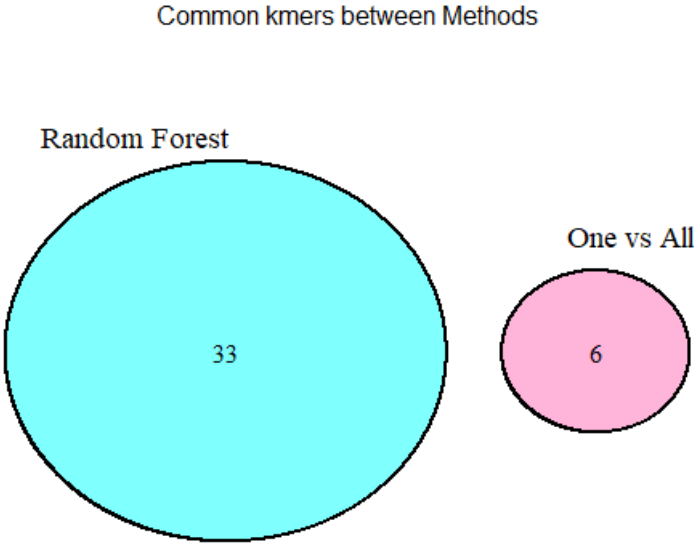
Εικόνα 125 Κοινά k-mers μεταξύ μεθόδων - Pig



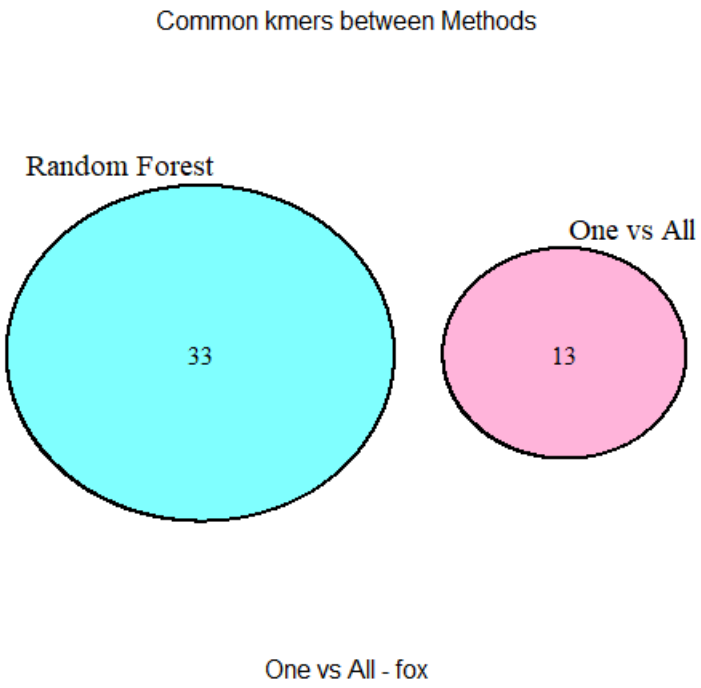
Εικόνα 126 Κοινά k-mers μεταξύ μεθόδων - Horse



Εικόνα 127 Κοινά k-mers μεταξύ μεθόδων - Dog



Εικόνα 128 Κοινά k-mers μεταξύ μεθόδων - Cat



Εικόνα 129 Κοινά k-mers μεταξύ μεθόδων - Fox

```
$human
character(0)

$cattle
[1] "GGGGGGAAA" "GGACCCCCC" "CGGGGGGGC" "GGGGGGGAC"
[5] "AAACCCCCC" "CGTCCCCCC" "AGAGGGGGG" "GGGGGGGAA"

$birds
character(0)

$pig
[1] "AAAAAAAAG"

$horse
character(0)

$dog
character(0)

$cat
character(0)

$fox
character(0)
```

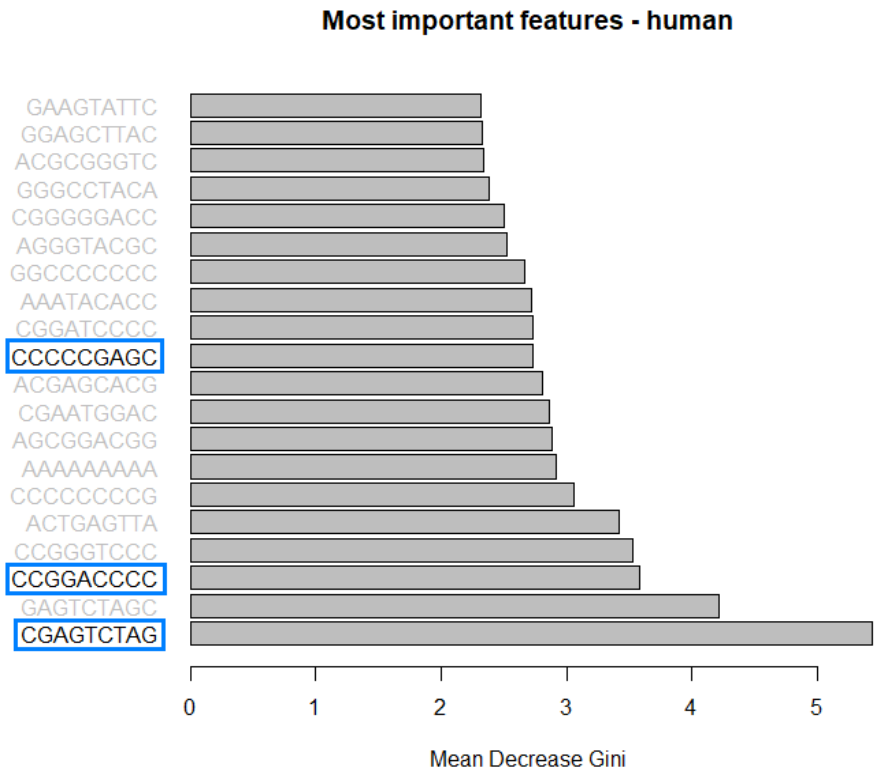
Εικόνα 130 Κοινά k-mers μεταξύ RF- OvA για κάθε οργανισμό

Τα κμερς που θεωρήθηκαν πως είναι σημαντικά και στα δύο σύνολα και στις δύο μεθόδους είναι τα παραπάνω. Συγκρίνοντας τα αποτελέσματα αυτά (Εικόνα 130) με τα αποτελέσματα της ΟνΑ μεθόδου για τα δύο σύνολα (Εικόνα 119), παρατηρείται πως τα 8 k-mers των βοοειδών εδώ είναι τα πρώτα 8 από τα 10 που βγήκαν πριν και το ένα που βγήκε στο γουρούνι είναι το 4^{ο} από τα 6. Παρατηρείται πως δεν υπάρχουν αποτελέσματα σε όλους τους οργανισμούς, επομένως τα k-mers αυτά δεν μπορούν να μελετηθούν μόνα τους. Θα σημειωθεί απλά πως είναι σημαντικά.

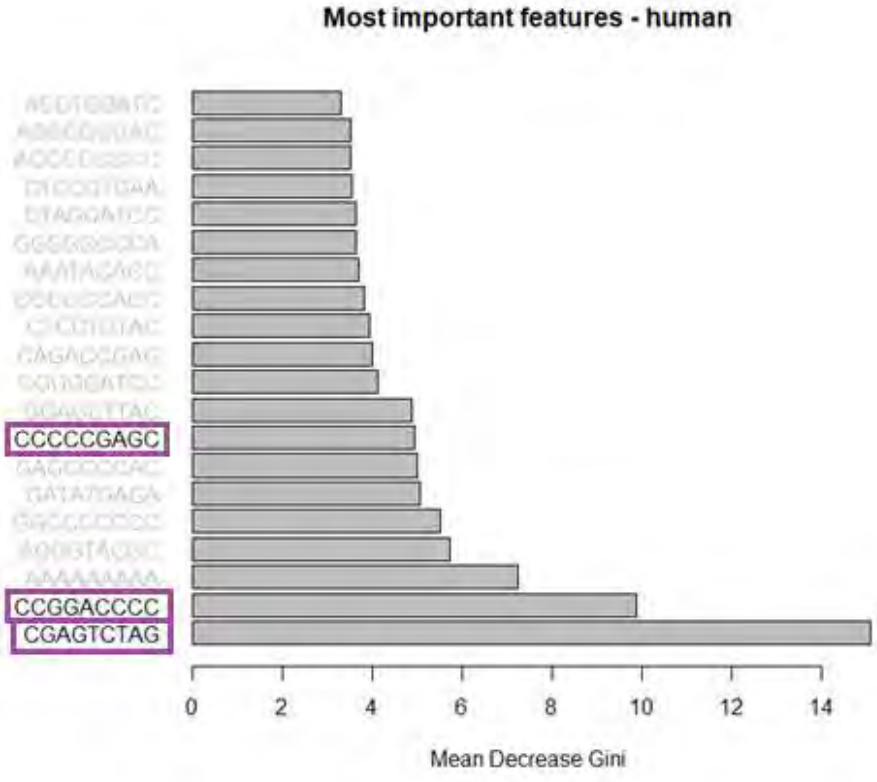
4.15 Επιλογή κμερς

Τέλος επιλέγονται τα 3 πιο σημαντικά κμερς για κάθε οργανισμό. Τα κμερς που συλλέχθηκαν (εικ. Εικόνα 119) συγκρίνονται με τα διαγράμματα κατάταξης των κμερς κάθε οργανισμού για κάθε σύνολο δεδομένων αλλά μόνο της μεθόδου OvA εφόσον σε αυτή βασίζονται τα αποτελέσματα [εικόνες: σύνολο 470: Εικόνα 91, Εικόνα 92, Εικόνα 93, Εικόνα 94, Εικόνα 95, Εικόνα 96, Εικόνα 97, Εικόνα 98, σύνολο 284: Εικόνα 100, Εικόνα 101, Εικόνα 102, Εικόνα 103, Εικόνα 104, Εικόνα 105, Εικόνα 106, Εικόνα 107]. Για κάθε οργανισμό δηλαδή ελέγχονται σε ποια θέση βρίσκονται σε καθένα από τα διαγράμματα στις εικόνες. Αυτά καταμετρώνται και κατατάσσονται ώστε να βρεθούν τα 3 πιο σημαντικά βάση και των δυο συνόλων. Η αρχική σειρά στην εικόνα Εικόνα 119 είναι βάση του συνόλου των 470, όπως επιβεβεώνεται και από τη σύγκριση. Τα 3 πρώτα κμερς τελικά φαίνεται να παραμένουν τα ίδια πρώτα 3 με της εικόνας εκτός των οργανισμών «Cattle» και «Birds» όπου τρίτο στη κατάταξη μπαίνει το τέταρτο κμερ της σειράς της εικόνας. Τα κμερς αυτά αναφέρονται στη συνέχεια μαζί με την αναπαράσταση της θέσης τους στα διαγράμματα των συνόλων.

- Human: "CGAGTCTAG", "CCGGACCCC", "CCCCCGAGC"

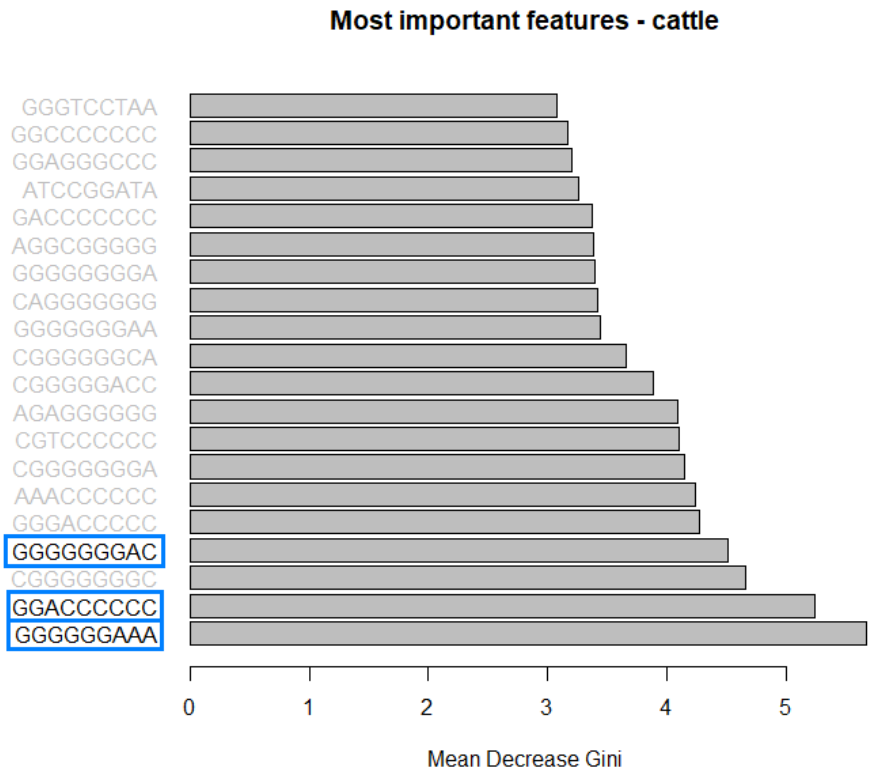


Εικόνα 131 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 470 - Human

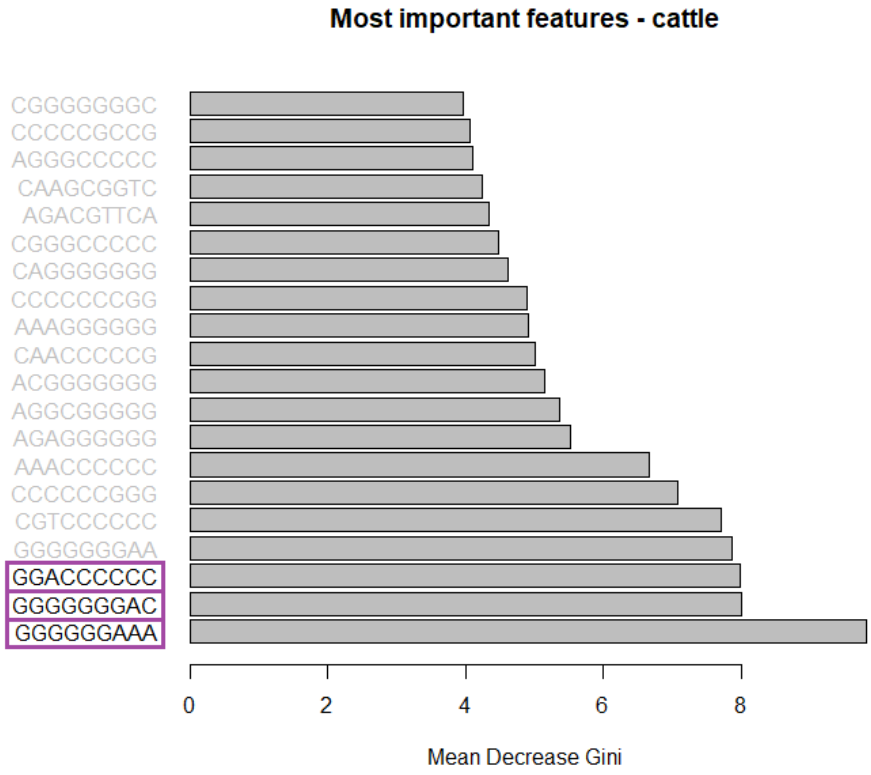


Εικόνα 132 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 284 - Human

➢ Cattle: "GGGGGGAAA", "GGACCCCCC", "GGGGGGGAC"

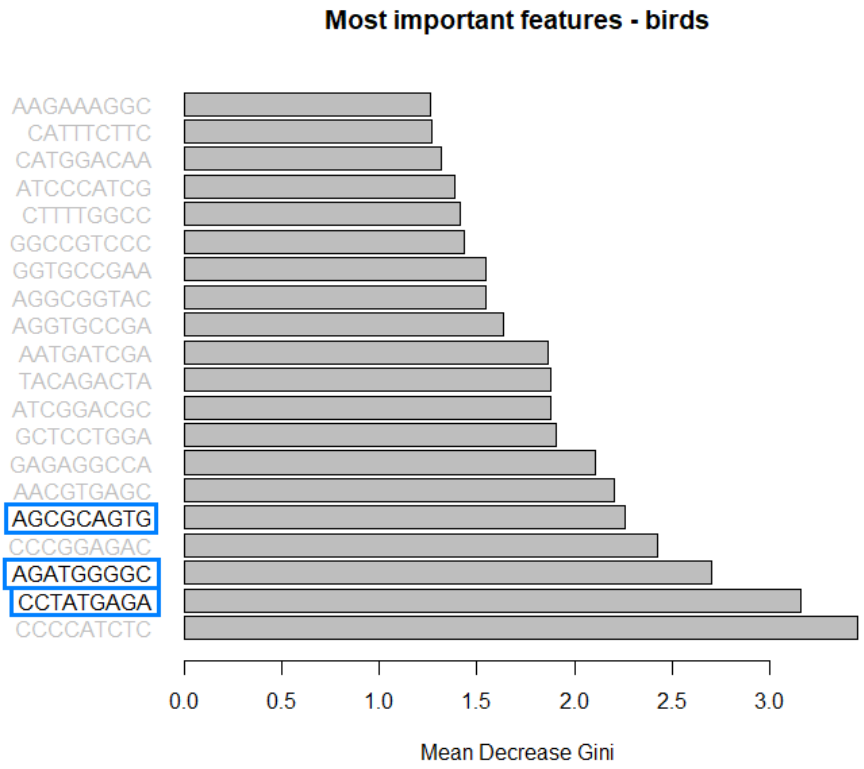


Εικόνα 133 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 470 - Cattle

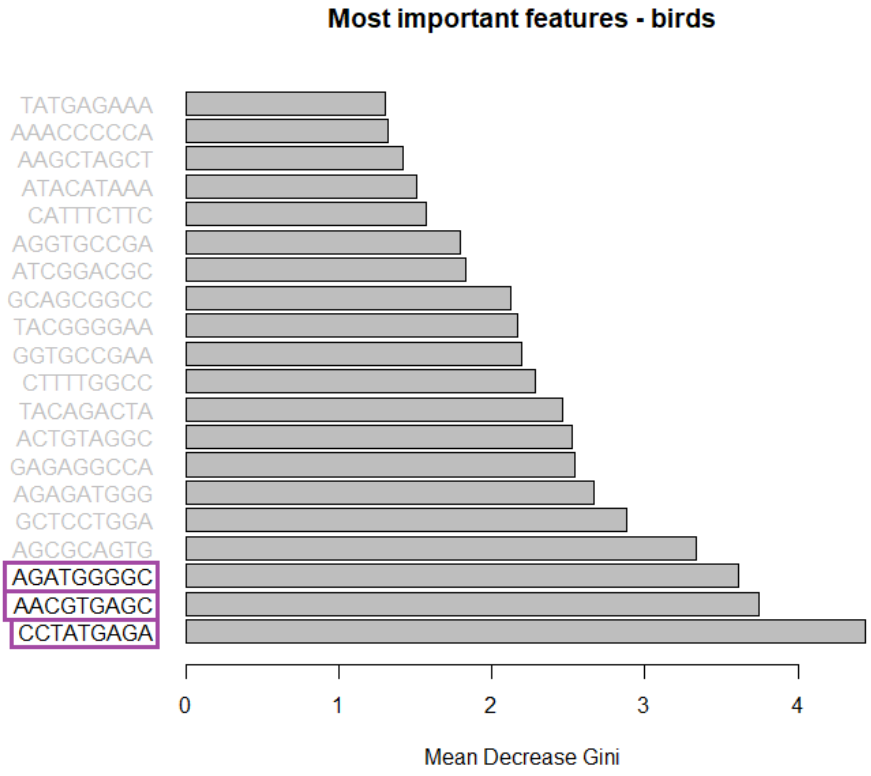


Εικόνα 134 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 284 - Cattle

- Birds: "CCTATGAGA", "AGATGGGGC", "AACGTGAGC"

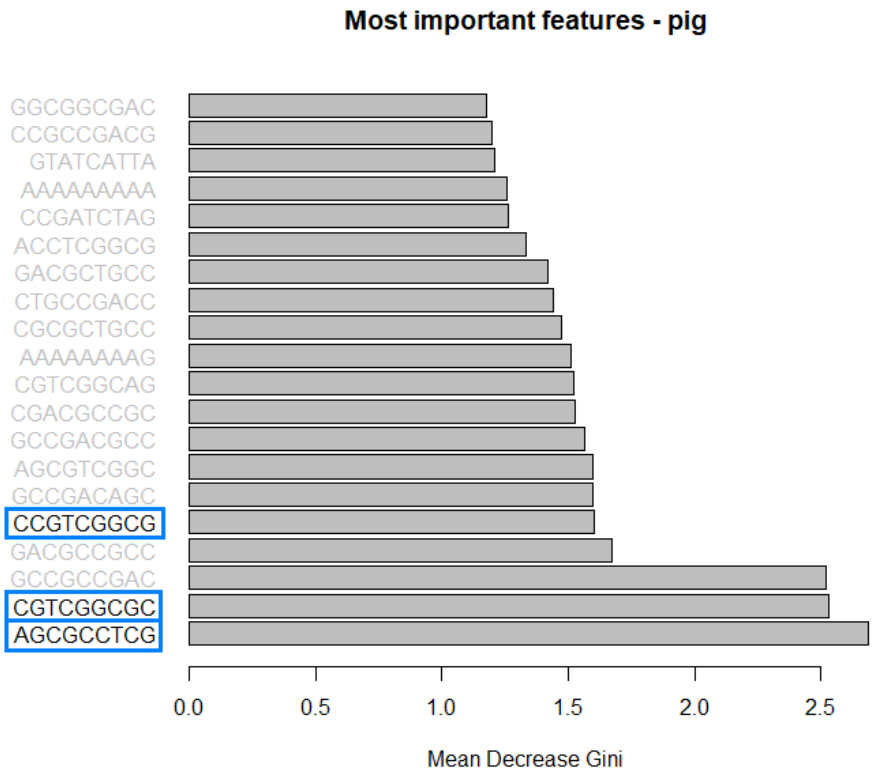


Εικόνα 135 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 470 - Birds

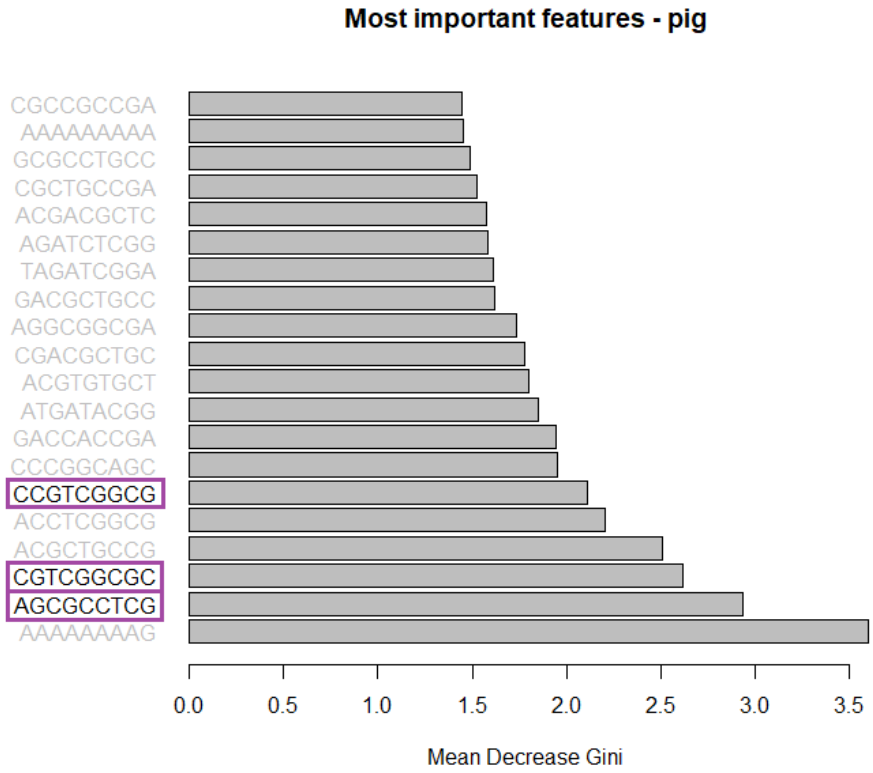


Εικόνα 136 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 284 – Birds

- Pig: "AGCGCCTCG", "CGTCGGCGC", "CCGTCGGCG"

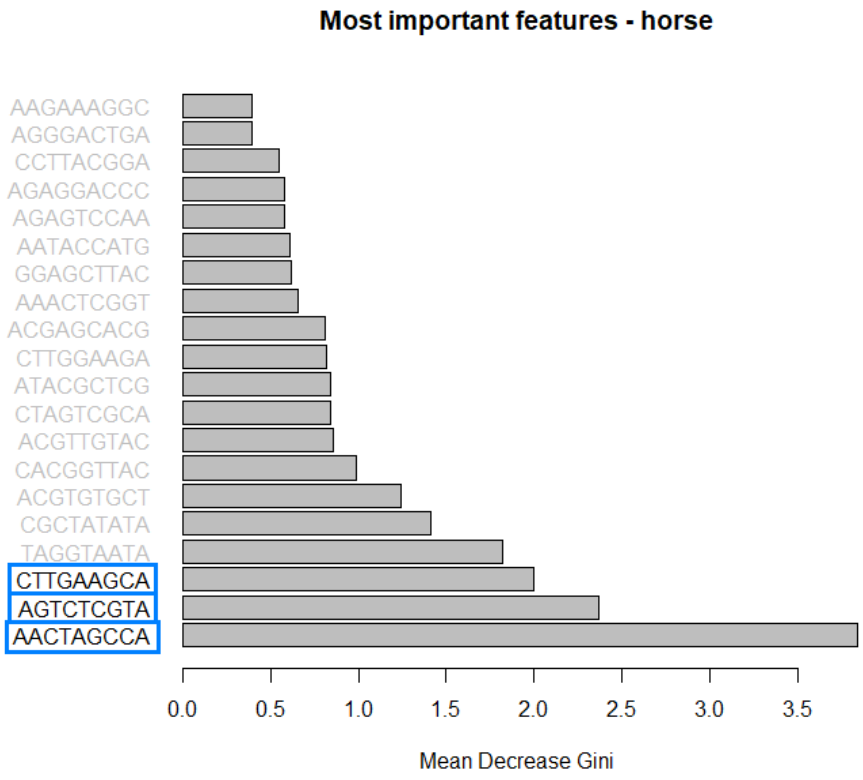


Εικόνα 137 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 470 - Pig

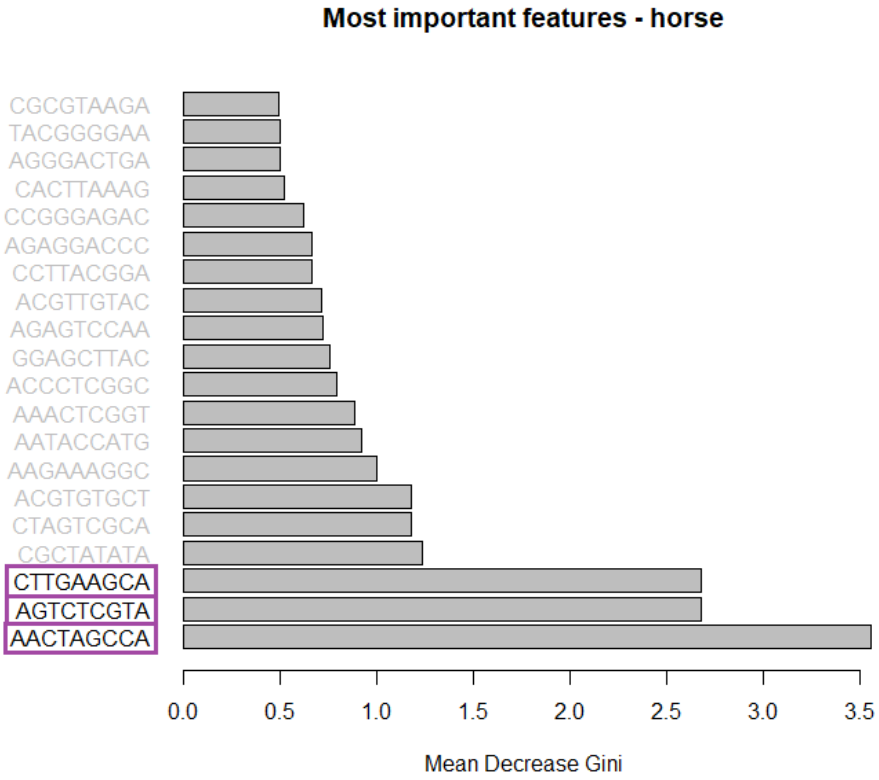


Εικόνα 138 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 284 - Pig

- Horse: "AACTAGCCA", "AGTCTCGTA", "CTTGAAGCA"

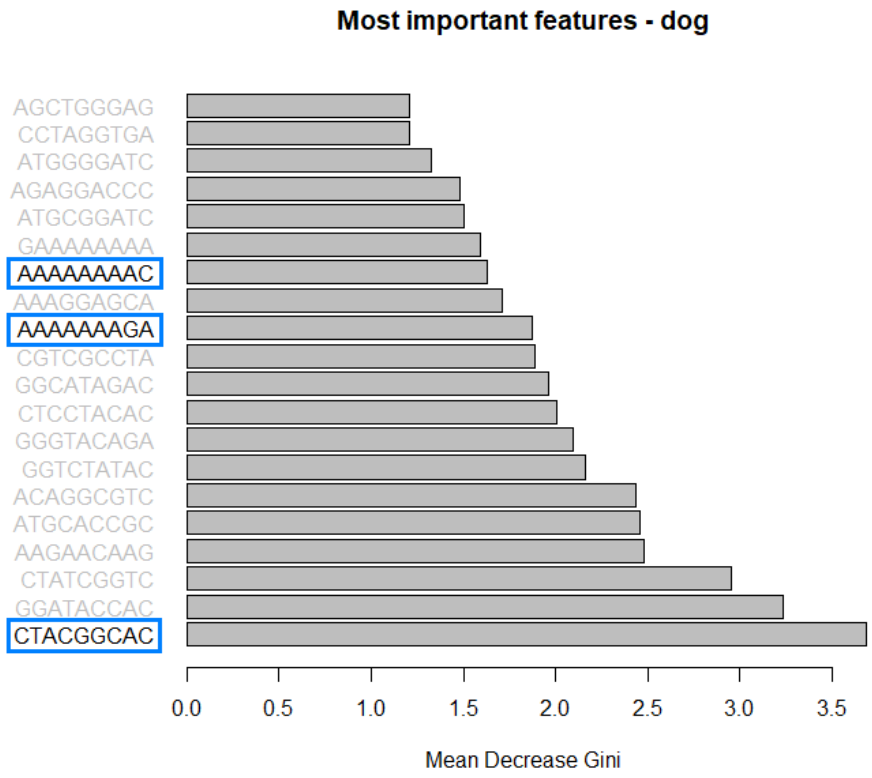


Εικόνα 139 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 470 - Horse

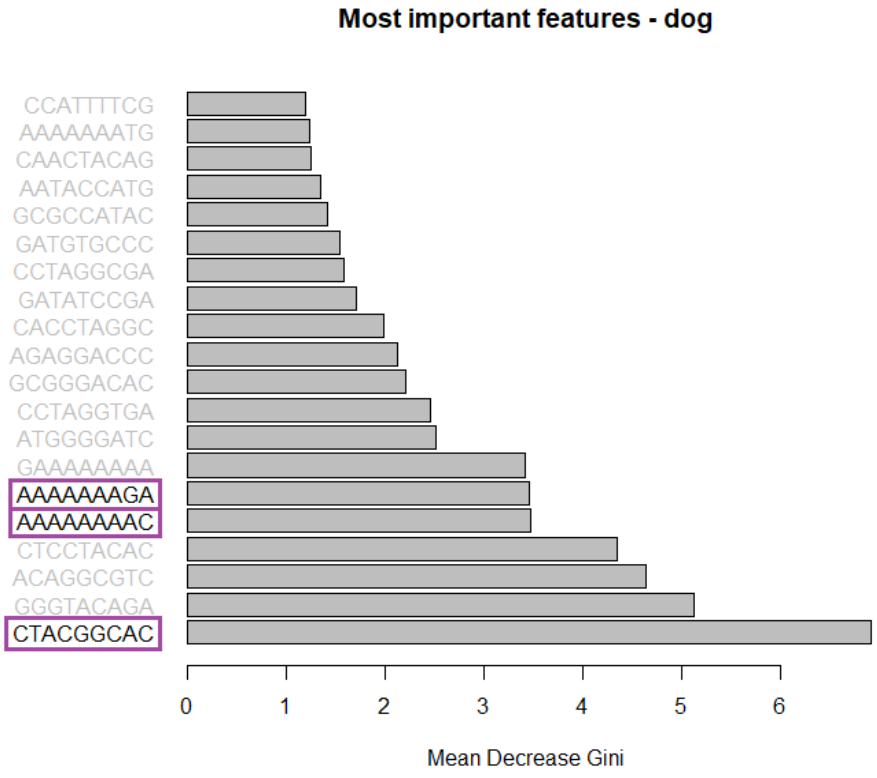


Εικόνα 140 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 284 - Horse

- Dog: "CTACGGCAC", "AAAAAAAGA", "AAAAAAAAC"

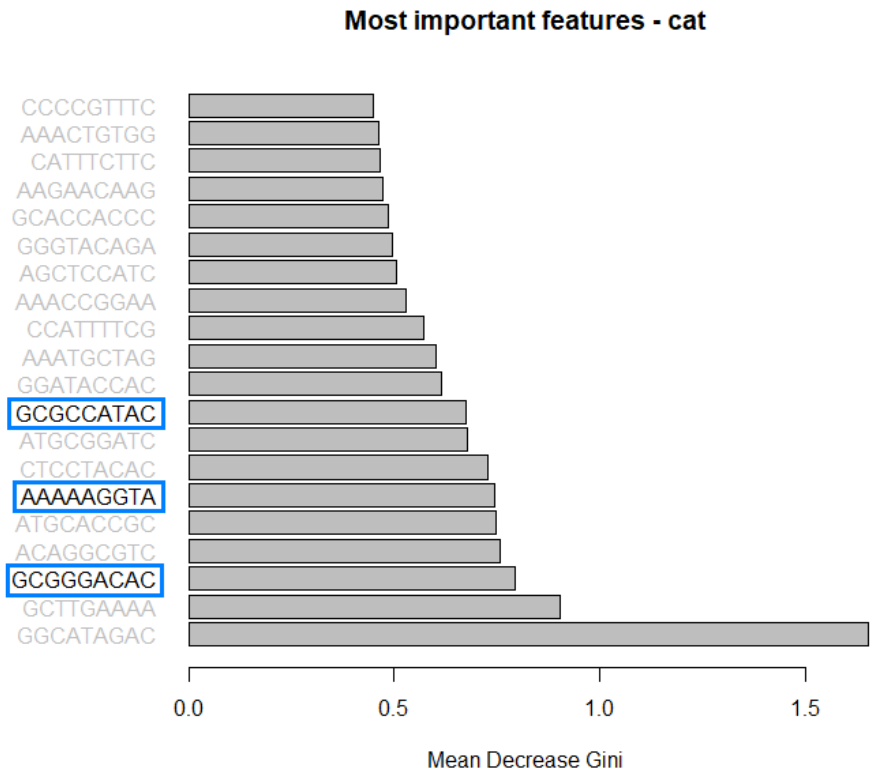


Εικόνα 141 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 470 - Dog

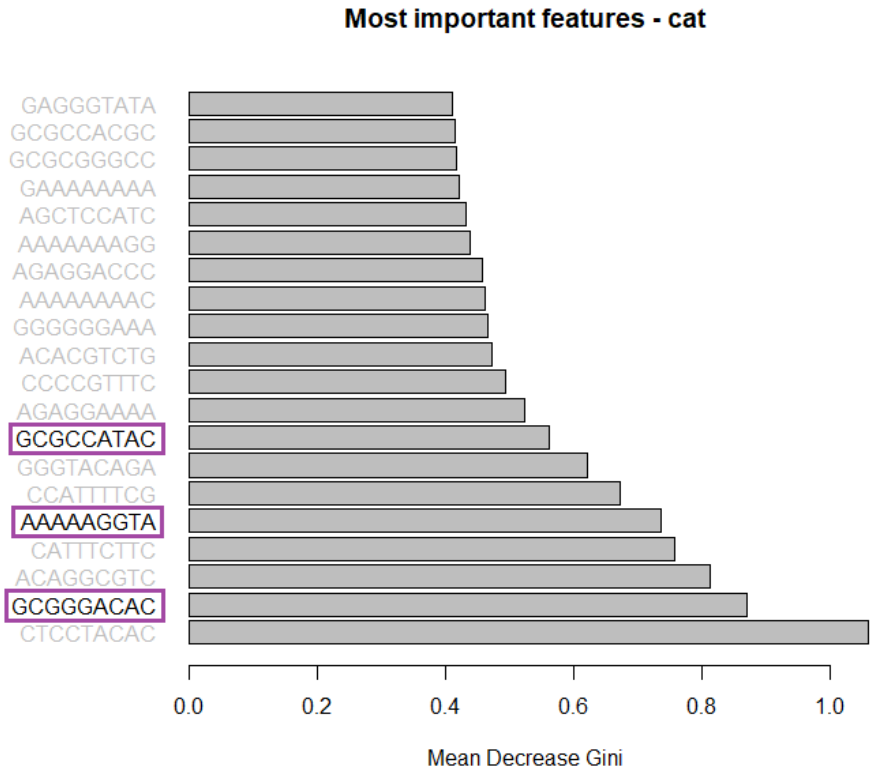


Εικόνα 142 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 284 - Dog

- Cat: "GCGGGACAC", "AAAAAGGTA", "GCGCCATAC"

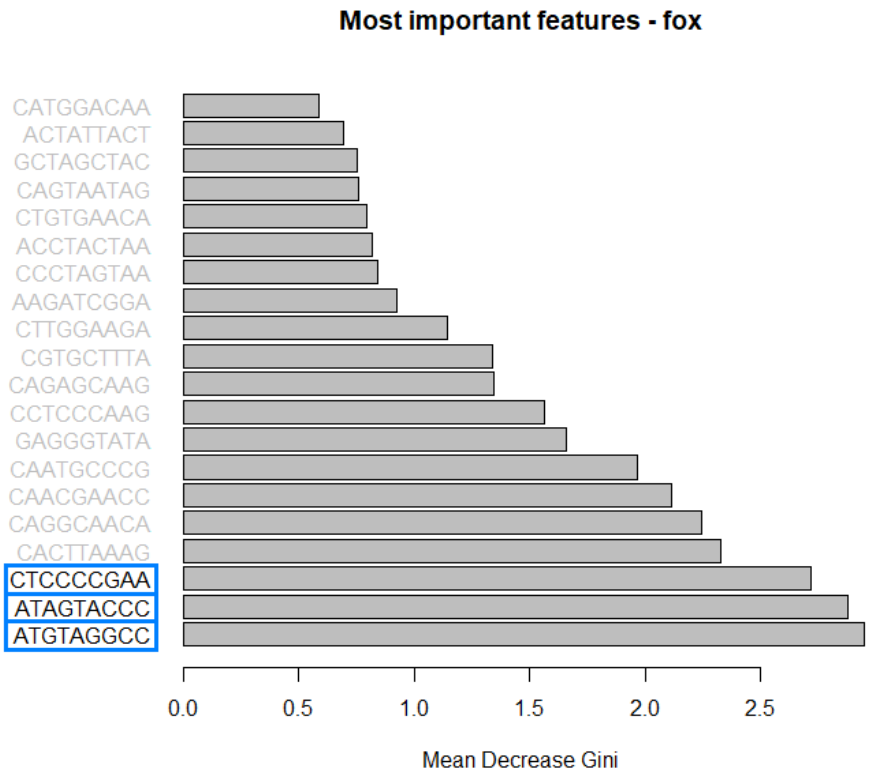


Εικόνα 143 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 470 - Cat

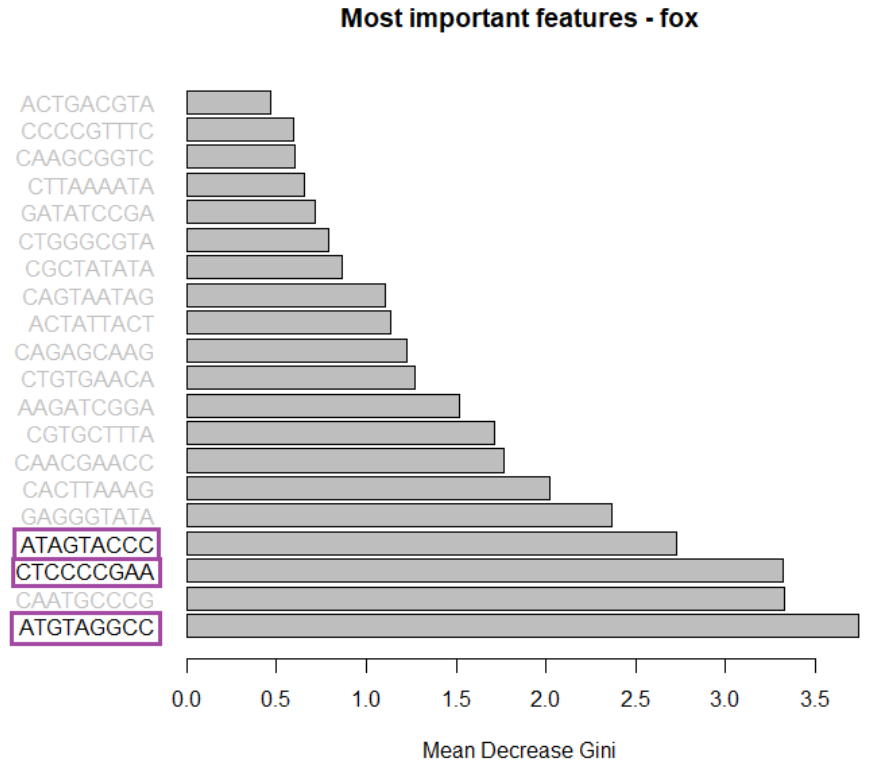


Εικόνα 144 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 284 - Cat

- Fox: "ATGTAGGCC", "ATAGTACCC", "CTCCCCGAA"



Εικόνα 145 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 470 - Fox



Εικόνα 146 Τα 3 πιο σημαντικά k-mers στην αρχική κατάταξη των 284 – Fox

4.16 BLAST

Τα k-mers λοιπόν που θα χρησιμοποιηθούν για την αναζήτηση βιοδεικτών που χαρακτηρίζουν κάθε οργανισμό από τους υπόλοιπους είναι τα εξής.

- Human:

  "CGAGTCTAG", "CCGGACCCC", "CCCCCGAGC"

- Cattle:

  "GGGGGGAAA", "GGACCCCCC", "CGGGGGGGC"

- Birds:

  "CCTATGAGA", "AGATGGGGC", "AGCGCAGTG"

- Pig:

  "AGCGCCTCG", "CGTCGGCGC", "CCGTCGGCG"

- Horse:

  "AACTAGCCA", "AGTCTCGTA", "CTTGAAGCA "

- Dog:

  "CTACGGCAC", "AAAAAAAGA", "AAAAAAAAC "

- Cat:

  "GCGGGACAC", "AAAAAGGTA", "GCGCCATAC "

- Fox:

  "ATGTAGGCC", "ATAGTACCC", "CTCCCCGAA "

Οι πρωτεΐνες που κωδικοποιούνται από γονίδια στα οποία βρίσκονται τα παραπάνω κμερς αναγράφονται παρακάτω. Τα 9μερή αυτά κωδικοποιούν ακριβώς αμινοξέα δηλαδή κάθε τριάδα βάσεων αντιστοιχεί ακριβώς στις τριπλέτες του DNA ή εν μέρει δηλαδή οι βάσεις αυτές συμπληρώνουν προηγούμενες και επόμενες τριπλέτες. Επίσης μπορεί να βρίσκονται είτε στη κωδική είτε στη συμπληρωματική αλυσίδα. Ορισμένες φορές μέρος των k-mers βρίσκεται σε αλληλουχίες έναρξης ή λήξης και σημειώνονται στις πρωτεΐνες με αγκύλες{} για τα έναρξης και ορθογώνια παρένθεση [] για της λήξης.

4.16.1 Human

Το πρώτο k-mer για τον άνθρωπο είναι το «CGAGTCTAG» και εμφανίζεται μία φορά στο πλασμίδιο pKPHS2 στο γονίδιο για τη πρωτεΐνη «**Class A Carbapenemase Kpc-2**». Εμφανίζεται επίσης 2 φορές στο χρωμόσωμα αλλά σε κανένα γονίδιο.

Το δεύτερο k-mer του ανθρώπου είναι το «CCGGACCCC» το οποίο εμφανίζεται στο χρωμόσωμα 7 φορές στα παρακάτω γονίδια:

- putative LysR-family transcriptional regulator ,
- hypothetical protein,
- LysR family transcriptional regulator,
- acetolactate decarboxylase,
- two-component regulatory system sensor protein,
- permease, cytosine/purine, uracil, thiamine, allantoin family,
- glycerol dehydratase

Το τρίτο k-mer του ανθρώπου είναι το «CCCCCGAGC» και εμφανίζεται 24 φορές στο χρωμόσωμα. Η εμφάνιση 7 δεν αντιστοιχεί σε γονίδιο.

- bifunctional isocitrate dehydrogenase kinase/phosphatase protein,
- TIM barrel protein,
- response regulator transcription factor,
- 4-hydroxythreonine-4-phosphate dehydrogenase,
- 3-methyl-2-oxobutanoate hydroxymethyltransferase,
- BCCT family transporter,
- YceH family protein,
- leucine export protein LeuE,
- transcriptional regulator CysB,
- putative regulatory protein TetR,
- putative ribulose-phosphate 3-epimerase,
- histidine transport,
- ABC superfamily ATP binding cassette transporter, ABC protein,
- shikimate 5-dehydrogenase,
- high-affinity branched-chain amino acid transport,
- transcriptional regulator HexR,
- 3-mercaptopyruvate sulfurtransferase,

- putative transmembrane protein,
- propanediol utilization: CoA-dependent propionaldehyde dehydrogenase,
- undecaprenyl pyrophosphate phosphatase,
- glutathione-regulated potassium-efflux system protein,
- peptidoglycan synthetase,
- putative oxidoreductase

4.16.2 Cattle

Το πρώτο k-mer στα βοοειδή είναι το «GGGGGGAAA» και συναντάται στο πλασμίδιο pKPHS2 σε γονίδιο που παράγει μια υποθετική πρωτεΐνη και στο pKPHS3 το οποίο δεν αντιστοιχεί σε γονίδιο. Στο χρωμόσωμα βρίσκεται 22 φορές στα παρακάτω γονίδια. Οι εμφανίσεις 2,3,9,11,13,17,18 από τις 22 είναι σε κενή περιοχή.

- putative hydrolase,
- putative creatininase,
- acridine efflux pump,
- patatin-like phospholipase RssA,
- dihydroneopterin aldolase,
- putative ABC transporter permease,
- glycogen debranching protein,
- periplasmic maltose-binding protein,
- hypothetical protein,
- hypothetical protein,
- DUF4186 domain-containing protein,
- hypothetical protein,
- hypothetical protein,
- 1,3-propanediol oxidoreductase,
- 7-cyano-7-deazaguanine/7-aminomethyl-7-deazaguanine transporter,

Το δεύτερο k-mer «GGACCCCCC» των βοοειδών εμφανίζεται μία μόνο φορά στο γονιδίωμα, στο χρωμόσωμα, για τη πρωτεΐνη «**fumarate reductase**».

Το τρίτο k-mer των βοοειδών, το «GGGGGGGAC» εμφανίζεται ολόκληρο μόνο στο χρωμόσωμα 13 φορές στα γονίδια που παράγουν τις παρακάτω πρωτεΐνες. Οι εμφανίσεις 1 με 4 και 9 δεν βρίσκονται σε γονίδιο.

- GntR family transcriptional regulator,
- putative helix-turn-helix AraC-type transcriptional regulator,
- carbohydrate porin,
- patatin-like phospholipase RssA,
- sensor domain-containing diguanylate cyclase,
- DUF418 family protein,
- putative acyl-CoA N-acyltransferase,
- hydrogenase maturation protein

4.16.3 Birds

Το πρώτο k-mer για τα πουλιά είναι το «CCTATGAGA» το οποίο εμφανίζεται 13 φορές στο χρωμόσωμα και μία στο πλασμίδιο pKPHS3. Σε αυτό βρίσκεται στο γονίδιο για την πιθανή πρωτεΐνη «putative chloramphenicol and florfenicol resistance protein (CmlA)». Στο χρωμόσωμα βρίσκεται στις παρακάτω πρωτεΐνες. Η εμφάνιση 5 δεν αντιστοιχεί σε γονίδιο.

- periplasmic maltose-binding protein,
- FtsH protease regulator HflC,
- 4-hydroxyphenylacetate catabolism,
- Na(+)-translocating NADH-quinone reductase subunit D,
- putative chitinase II ,
- putative protease ,
- electron transport complex protein RnfD,
- TonB-dependent receptor,
- putative oligogalacturonide transporter,
- [D-arabinitol transporter],
- bifunctional NADH:ubiquinone oxidoreductase subunit C/D,
- DNA polymerase III subunit beta

Το δεύτερο k-mer για τα πουλιά «AGATGGGGC» εμφανίζεται μία φορά στο πλασμίδιο pKPHS3 σε γονίδιο που κωδικοποιεί κάποια υποθετική πρωτεΐνη και 41 φορές στο χρωμόσωμα. Η 21 εμφάνιση δεν βρίσκεται σε γονίδιο.

- nitrogen regulation protein NR(II),
- IclR family transcriptional regulator,

- lysyl-tRNA synthetase,
- putative periplasmic binding protein,
- carbon starvation protein,
- glutathione-regulated potassium-efflux system protein KefC,
- peptidyl-prolyl cis-trans isomerase SurA,
- lac repressor,
- putative transport protein, PTS system,
- putative aminotransferase,
- putative carbohydrate kinase,
- putative negative regulator,
- L,D-transpeptidase,
- putative glutathione S-transferase,
- D-alanyl-D-alanine carboxypeptidase,
- asparaginyl-tRNA synthetase,
- putative DsbA oxidoreductase,
- integral membrane protein MviN,
- putative LysR-family transcriptional regulator,
- putative ABC transporter permease,
- hypothetical protein,
- IclR family transcriptional regulator,
- formate dehydrogenase-N gamma subunit,
- putative NAD(P)-binding dehydrogenase,
- acridine efflux pump,
- putative H+/gluconate symporter,
- oligopeptide ABC transporter ATP-binding protein,
- High-molecular-weight nonribosomal peptide/polyketide synthetase 1 (HMWP1),
- putative LysR-family transcriptional regulator,
- putative protease,
- oligopeptide ABC transporter substrate-binding protein,
- putative acyl-CoA N-acyltransferase,
- 6-phospho-beta-glucosidase,
- bifunctional chorismate mutase/prephenate dehydrogenase,
- GntR family transcriptional regulator,

- hydrogenase expression/formation protein HypE,
- acetolactate synthase III large subunit,
- putative dienelactone hydrolase,
- glycogen phosphorylase,
- aspartate-semialdehyde dehydrogenase

Για το τρίτο k-mer των πουλιών, το «AACGTGAGC» βρίσκεται στα γονίδια που κωδικοποιούν τις παρακάτω πρωτεΐνες του χρωμοσώματος. Οι εμφανίσεις 1, 6, 17, 18 των 19 δεν αντιστοιχούν σε γονίδιο.

- DUF413 domain-containing protein,
- 23S rRNA pseudouridine synthase F,
- sodium/proton antiporter NhaA,
- cell division protein FtsA,
- putative Zn-dependent carboxypeptidase,
- serine hydroxymethyltransferase,
- thioredoxin 2,
- propanediol utilization polyhedral bodies protein,
- conjugative transfer ATPase,
- putative keto-hydroxyglutarate-aldolase/keto- deoxy- phosphogluconate aldolase,
- bifunctional metallophosphatase/5'-nucleotidase,
- formate C-acetyltransferase,
- murein DD-endopeptidase MepM,
- High-molecular-weight nonribosomal peptide/polyketide synthetase 2 (HMWP2),
- ATP-dependent metallopeptidase HflB

4.16.4 Pig

Τα k-mers του γουρουνιού εμφανίζονται και σε πλασμίδια αλλά το σημαντικό είναι να αναφερθεί πως οι εμφανίσεις τους στο χρωμόσωμα είναι στο επίπεδο των εκατοντάδων και η αναζήτηση και καταγραφή τους θεωρήθηκε πλεονασμός. Για το k-mer «AGCGCCTCG», η εμφάνιση του στα πλασμίδια pKPHS2 και pKPHS3 αντιστοιχούν στα γονίδια για τις πρωτεΐνες «conjugal transfer nickase/helicase TraI»

και «aminoglycoside adenylyltransferase» αντίστοιχα. Στο χρωμόσωμα εμφανίζεται 248 φορές. Για το δεύτερο k-mer, το «CGTCGGCGC» εμφανίζεται δύο φορές στο πλασμίδιο pKPHS2, με τη πρώτη να μην αντιστοιχεί σε γονίδιο και τη δεύτερη να βρίσκεται σε κάποια υποθετική πρωτεΐνη. Στο πλασμίδιο pKPHS1 βρίσκεται επίσης δύο φορές οι οποίες αντιστοιχούν και οι δύο σε υποθετικές πρωτεΐνες. Στο χρωμόσωμα εμφανίζεται 273 φορές. Τέλος για το τρίτο k-mer «CCGTCGGCG» εμφανίζεται στο πλασμίδιο pKPHS1 τρεις φορές, η πρώτη στη πρωτεΐνη Gp21 και οι άλλες δύο σε υποθετικές πρωτεΐνες. Οι εμφανίσεις του στο χρωμόσωμα είναι 271.

4.16.5 Horse

Για το άλογο το πρώτο k-mer «AACTAGCCA» εμφανίζεται στο γονιδίωμα της Klebsiella pneumoniae στο πλασμίδιο pKPHS2 στο γονίδιο «conjugal transfer pilus assembly protein TraB» και στο πλασμίδιο pKPHS1 στο πιθανό γονίδιο «putative phage tail protein». Στο χρωμόσωμα εμφανίζεται 3 φορές με τη δεύτερη να μη βρίσκεται σε γονίδιο και τις άλλες δύο να είναι στα:

- microcin H47 secretion protein
- putative structural protein

Το δεύτερο k-mer του αλόγου «AGTCTCGTA» βρίσκεται στα πλασμίδια pKPHS3 και pKPHS1 σε υποθετικές πρωτεΐνες στο καθένα και στο χρωμόσωμα 4 φορές. Τα γονίδια είναι τα:

- hypothetical protein,
- outer membrane protein X,
- putative sensor kinase,
- ABC transporter ATP-binding protein

Το τρίτο k-mer «CTTGAAGCA» συναντάται στα πλασμίδια pKPHS2 και pKPHS3 σε υποθετικές πρωτείνες και στο χρωμόσωμα 15 φορές με την $7^{η}$ και $12^{η}$ να είναι σε κενή περιοχή. Η $10^{η}$ εμφάνιση του k-mer στο χρωμόσωμα ανήκει σε δύο γονίδια ταυτόχρονα.

- 6-N-hydroxylaminopurine resistance protein,
- putative amidohydrolase 2,
- DnaK suppressor protein,

- acridine efflux pump,
- putative inner membrane protein,
- 30S ribosomal protein S12 methylthiotransferase RimO,
- 30S ribosomal protein S12 methylthiotransferase RimO,
- dihydrodipicolinate synthase,
- putative endoribonuclease + [putative cytosine deaminase],
- putative transposase,
- antibiotic biosynthesis monooxygenase,
- lipoprotein NlpI,
- alpha-xylosidase

4.16.6 Dog

Στο σκύλο το k-mer «CTACGGCAC» συναντάται στο πλασμίδιο pKPHS3 τρεις φορές στα γονίδια των πρωτεϊνών «class I integron integrase», «beta-lactamase CTX-M-14» και «Ner-like protein». Στο πλασμίδιο pKPHS1 βρίσκεται στο γονίδιο για τη «beta-lactamase CTX-M-14» και στο χρωμόσωμα είναι 51 φορές στα παρακάτω γονίδια.

- trehalase 6-P hydrolase,
- "putative aminotransferase, class I and II",
- right origin-binding protein,
- MFS family transporter,
- "outer membrane protein assembly complex, YaeT protein",
- queuine tRNA-ribosyltransferase,
- protoheme IX farnesyltransferase,
- transcriptional regulator BolA,
- "Carbohydrate kinase, FGGY-like protein",
- chitoporin,
- molybdopterin biosynthesis protein MoeB,
- anaerobic dimethyl sulfoxide reductase subunit A,
- asparaginyl-tRNA synthetase,
- aldehyde dehydrogenase,
- putative LysR-family transcriptional regulator,
- putative chitinase II,

- putative arginine/ornithine antiporter,
- NAD(P) transhydrogenase subunit alpha,
- acetyl-CoA:acetoacetyl-CoA transferase alpha subunit,
- putative dehydrogenase,
- putative beta-ketoacyl synthase,
- putative epoxide hydrolase protein,
- putative ABC transport system permease,
- putative transmembrane protein,
- FAD-binding protein,
- TonB-dependent receptor,
- threonyl-tRNA synthetase,
- bifunctional acetaldehyde-CoA/alcohol dehydrogenase,
- nitrate reductase 1 subunit alpha,
- high-affinity branched-chain amino acid ABC transporter membrane protein,
- AMP nucleosidase,
- putative chaperone,
- glycine betaine/choline ABC transporter membrane protein,
- DUF4440 domain-containing protein,
- putative iron-regulated membrane protein,
- formate hydrogenlyase subunit 3,
- outer membrane porin for ferric enterobactin and colicins B and D,
- glycerate kinase,
- “cobyrinic Acid a,c-diamide synthase”,
- PduO protein,
- bifunctional glutathionylspermidine amidase/glutathionylspermidine synthetase,
- “3,4-dihydroxy-2-butanone 4-phosphate synthase”,
- glycerol dehydratase,
- “cob(I)yrinic acid a,c-diamide adenosyltransferase”,
- lipoprotein NlpI,
- glutamate synthase large subunit,
- 30S ribosomal protein S9,
- cAMP-regulatory protein,
- GntP family high-affinity gluconate permease,

- glycerol kinase,
- putative dehydrogenase

Το επόμενο k-mer «AAAAAAAGA» του σκύλου εμφανίζεται στο πλασμίδιο pKPHS5 σε υποθετική πρωτεΐνη, στο pKPHS2 δύο φορές αλλά σε κανένα γονίδιο και στο pKPHS1 τη πρώτη εμφάνιση σε κενό και τη δεύτερη σε υποθετική πρωτεΐνη. Τέλος, εμφανίζεται στο χρωμόσωμα, 81 φορές. Από αυτές η $2^{η}$, $3^{η}$, $4^{η}$, $7^{η}$, $10^{η}$, $11^{η}$, $12^{η}$, $16^{η}$, $18^{η}$, $20^{η}$, $24^{η}$, $29^{η}$, $30^{η}$, $31^{η}$, $32^{η}$, $33^{η}$, $35^{η}$, $38^{η}$, $42^{η}$, $50^{η}$, $51^{η}$, $52^{η}$, $54^{η}$, $57^{η}$, $58^{η}$, $59^{η}$, $63^{η}$, $66^{η}$, $69^{η}$, $74^{η}$, $75^{η}$, $76^{η}$ ανήκουν σε γονίδια πρωτεϊνών όπως αναγράφονται παρακάτω ενώ οι υπόλοιπες 49 από τις 81 δεν ανήκουν σε γονίδιο.

- hypothetical protein,
- putative chaperone,
- type IV toxin-antitoxin system AbiEi family antitoxin,
- amidohydrolase family protein,
- DNA polymerase III subunits gamma and tau,
- DUF2213 domain-containing protein,
- putative inner membrane protein,
- regulator for leucine (or lrp) regulon and high-affinity branched-chain amino acid transport system,
- hypothetical protein,
- hypothetical protein,
- hypothetical protein,
- putative Retron-type reverse transcriptase,
- glycosyltransferase family 61 protein,
- EpsG family protein,
- EpsG family protein,
- carbohydrate lyase,
- DUF418 family protein,
- hypothetical protein,
- transcription elongation factor GreB,
- hypothetical protein,
- major facilitator superfamily permease,
- hypothetical protein,
- putative carbohydrate kinase,

- hypothetical protein,
- putative arginine/ornithine antiporter,
- molybdate ABC transporter permease,
- hypothetical protein,
- hypothetical protein,
- ATP-binding protein,
- hypothetical protein,
- hypothetical protein,
- hypothetical protein,

Το k-mer «AAAAAAAAC» του σκύλου εμφανίζεται 4 φορές στο πλασμίδιο pKPHS2 στα γονίδια «hypothetical protein», «hypothetical protein» και «conjugal transfer pilus assembly protein TraB» με τη τρίτη εμφάνιση να μην ανήκει σε γονίδιο. Το πλασμίδιο pKPHS1 έχει το k-mer 2 φορές με τη πρώτη εμφάνιση να είναι σε υποθετικό γονίδιο και τη δεύτερη να βρίσκεται σε κενή περιοχή. Στο πλασμίδιο pKPHS3 υπάρχει 3 φορές αλλά σε κανένα γονίδιο. Στο χρωμόσωμα εμφανίζεται 76 φορές αλλά μόνο οι 21 φορές είναι σε γονίδια και αυτές είναι η $6^{η}$, $9^{η}$, $11^{η}$, $12^{η}$, $15^{η}$, $16^{η}$, $24^{η}$, $28^{η}$, $29^{η}$, $33^{η}$, $34^{η}$, $35^{η}$, $39^{η}$, $42^{η}$, $44^{η}$, $46^{η}$, $60^{η}$, $62^{η}$, $63^{η}$, $69^{η}$ και $73^{η}$.

- hypothetical protein ,
- sensor protein for basR, -,
- hypothetical protein, hypothetical protein, -, -,
- DMT family transporter,
- " phosphate/phosphonate ABC superfamily ATP binding cassette transporter, binding protein",
- EpsG family protein,
- tRNA1(Val) (adenine(37)-N6)-methyltransferase,
- [ATP-dependent RNA helicase SrmB],
- hypothetical protein,
- hypothetical protein,
- adenosine diphosphate sugar pyrophosphatase,
- hypothetical protein,
- [DUF2950 family protein],
- hypothetical protein,
- FAD-dependent oxidoreductase,

- putative ARAC-type regulatory protein,
- N-acylhomoserine lactone degredation protein AhlK,
- phosphoglycerate mutase,
- protein kinase-like protein,
- tRNA pseudouridine synthase D

4.16.7 Cat

Στη γάτα το πρώτο k-mer «GCGGGACAC» βρίσκεται στα πλασμίδια pKPHS2, pKPHS3 και pKPHS1 για το ίδιο γονίδιο, το «transposase for transposon Tn1721» ενώ στο πλασμίδιο pKPHS3 βρίσκεται και σε μία υποθετική πρωτεΐνη. Στο χρωμόσωμα εμφανίζεται 31 φορές από τις οποίες η $5^{η}$, $6^{η}$, $14^{η}$, και $29^{η}$ δεν βρίσκονται σε γονίδιο.

- rRNA-23S ribosomal RNA,
- rRNA-23S ribosomal RNA,
- rRNA-23S ribosomal RNA,
- rRNA-23S ribosomal RNA,
- putative histidine protein kinase,
- rRNA-23S ribosomal RNA,
- DNA polymerase II,
- rRNA-23S ribosomal RNA,
- FAD-dependent oxidoreductase,
- putative ABC transport system permease,
- potassium-transporting ATPase subunit B,
- 30S ribosomal protein S12 methylthiotransferase accessory protein YcaO,
- transcriptional regulator,
- putative LysR-family transcriptional regulator,
- LysR family transcriptional regulator,
- putative monooxygenase,
- winged helix-turn-helix domain-containing protein,
- phosphoenolpyruvate carboxykinase,
- putative type IV secretory pathway VirB4 component,
- hypothetical protein,

- D-alanyl-D-alanine carboxypeptidase,
- putative regulatory protein,
- transcriptional regulation of gcv operon,
- rRNA-23S ribosomal RNA,
- acyl carrier protein phosphodiesterase,
- rRNA-23S ribosomal RNA,
- DNA ligase

Το δεύτερο k-mer της γάτας «AAAAAGGTA» εμφανίζεται στα πλασμίδια pKPHS3, pKPHS2 και pKPHS1, με το πρώτο να έχει 3 εμφανίσεις του k-mer στα γονίδια για υποθετικές πρωτεΐνες, το δεύτερο στο γονίδιο για τη «DNA-binding protein» και το τρίτο στη «DNA polymerase III». Στο χρωμόσωμα βρίσκεται 37 φορές με τη $1^{η}$, $2^{η}$, $5^{η}$, $6^{η}$, $9^{η}$, $10^{η}$, $14^{η}$, $28^{η}$, $31^{η}$, και $32^{η}$ να βρίσκονται σε κενή περιοχή.

- catabolite repression sensor kinase,
- DNA polymerase II,
- cytosine permease,
- pyruvate formate lyase-activating enzyme 1,
- major facilitator superfamily permease,
- ATP phosphoribosyltransferase,
- elongation factor P-like protein YeiP,
- histidine/lysine/arginine/ornithine transporter subunit,
- gp36,
- di-/tripeptide transporter,
- ASCH domain-containing protein,
- integral membrane sensor signal transduction histidine kinase,
- "D-xylose ABC superfamily ATP binding cassette transporter, binding protein",
- heptosyl III transferase,
- heptosyl III transferase,
- putative acetyltransferase,
- EAL domain-containing protein,
- inner membrane transporter YjeM,
- "putative transport protein, PTS system",

- DUF3440 domain-containing protein,
- bifunctional putative acetyl-CoA:acetoacetyl-CoA transferase: alpha subunit/beta subunit,
- [oxidoreductase alpha],
- YeiH family protein,
- putative phopshatase/sulfatase,
- 3-dehydroquinate synthase,
- putative ABC-type multidrug transport system ATPase component,
- recombination protein F,

Το τρίτο k-mer της γάτας «GCGCCATAC» βρίσκεται στο πλασμίδιο pKPHS2 σε μια υποθετική πρωτεΐνη, στο πλασμίδιο pKPHS3 στο γονίδιο για τη πρωτεΐνη «streptomycin 3'-kinase» και στο pKPHS1 σε κενή περιοχή. Στο χρωμόσωμα εμφανίζεται 94 φορές, με τις εμφανίσεις 12, 13, 17, 22, 58, 59, 73, 75, 76 να μη βρίσκονται σε γονίδια.

- putative endonuclease,
- nitrogen regulation protein NR(I),
- putative inner membrane protein,
- rhaRS operon positive regulator,
- primosome assembly protein PriA,
- 4-alpha-L-fucosyltransferase,
- outer membrane ferric enterobactin receptor,
- putative polysaccharide deacetylase,
- fumarate reductase,
- putative mannitol/fructose-specific PTS family enzyme IIA component,
- 5-deoxy-glucuronate isomerase,
- 4-hydroxy-3-methylbut-2-enyl diphosphate reductase,
- ATP-dependent helicase,
- hydroxamate-dependent iron transport system permease component,
- “ethanolamine ammonia-lyase, heavy chain”,
- homocysteine methyltransferase,
- DUF1615 domain-containing protein,
- DNA-binding ATP-dependent protease La,
- acridine efflux pump,

- putative cation:proton antiport protein,
- putative phenylalanine-specific permease,
- putative oxidoreductase,
- putative sugar-binding domain protein,
- ABC transporter membrane protein,
- flavin reductase family protein,
- integrase family protein,
- putative nucleoside-diphosphate-sugar epimerase,
- amino acid-binding protein,
- sugar efflux transporter,
- putative cytochrome oxidase,
- succinylarginine dihydrolase,
- metal-dependent hydrolase,
- putative transmembrane protein,
- putative dehydrogenase,
- aconitate hydratase,
- putative helix-turn-helix AraC-type transcriptional regulator,
- hypothetical protein,
- putative ARAC-type regulatory protein,
- putative oxidoreductase,
- ATP-dependent helicase,
- succinylglutamate desuccinylase,
- mechanosensitive ion channel,
- putative aldehyde dehydrogenase,
- putative 2-nitropropane dioxygenase,
- major facilitator family transporter,
- fucose permease,
- GrpB family protein,
- phosphoglycerate transporter,
- oxalurate catabolism protein HpxZ,
- putative AraC-type regulatory protein,
- putative regulator,
- putative oxidoreductase,
- acridine efflux pump,

- 5-methyltetrahydropteroyltriglutamate/homocysteine S-methyltransferase,
- YchE family NAAT transporter,
- two-component nitrate/nitrite sensory histidine protein kinase,
- threonine/serine transporter TdcC,
- alanine racemase,
- putative resistance protein,
- phosphogluconate dehydratase,
- arginyl-tRNA synthetase,
- High-molecular-weight nonribosomal peptide/polyketide synthetase 1 (HMWP1),
- hypothetical protein,
- DNA gyrase subunit A,
- menaquinone-specific isochorismate synthase,
- NADH dehydrogenase subunit N,
- uracil phosphoribosyltransferase,
- putative high-affinity nickel-transporter,
- putative LysR-family transcriptional regulator,
- hemolysin F,
- 23S rRNA (uracil-5-)-methyltransferase RumA,
- exonuclease V (RecBCD complex) subunit gamma,
- EamA family transporter,
- DUF1349 domain-containing protein,
- altronate hydrolase,
- tryptophan permease,
- hypothetical protein,
- putative competence protein,
- osmolarity sensor protein,
- low-affinity gluconate transporter,
- aromatic amino acid transporter,
- putative transmembrane protein,
- 2-hydroxyacid dehydrogenase,
- O-antigen polymerase,
- F0F1 ATP synthase subunit alpha,

4.16.8 Fox

Για την αλεπού, το πρώτο k-mer «ATGTAGGCC» εμφανίζεται στο πλασμίδιο pKPHS2 στο γονίδιο για τη πρωτεΐνη «antirestriction protein Klca» και στο χρωμόσωμα βρίσκεται 30 φορές με τις εμφανίσεις 8, 9, 10, 13, 15, 20, 24, 25 και 29 να μην αντιστοιχούν σε γονίδιο

- hypothetical protein,
- MFS transporter,
- major facilitator family transporter,
- sigma-54 dependent transcriptional regulator,
- putative Na+ dependent nucleoside transporter,
- [sodium ion-translocating decarboxylase subunit beta],
- penicillin-binding protein 1b,
- proline permease,
- putative citrate synthase,
- carbon starvation protein,
- manganese catalase family protein,
- DNA-binding transcriptional activator PspC,
- putative dimethyl sulfoxide reductase,
- major subunit,
- OmpR family transcriptional regulatory protein,
- PqiB family protein,
- High-molecular-weight nonribosomal peptide/polyketide synthetase 2 (HMWP2),
- manganese transport protein MntH, exopolyphosphatase,
- iron transporter: fur regulated,
- nucleoside permease NupG,
- cytoplasmic trehalase

Το δεύτερο k-mer της αλεπούς είναι το «ATAGTACCC» και βρίσκεται 13 φορές στο γονιδίωμα στο χρωμόσωμα. Οι εμφανίζεις 2, 8, 9 και 12 δεν αντιστοιχούν σε γονίδιο.

- excinuclease ABC subunit A,

- {sucrose-specific enzyme II},
- TcfC E-set like domain-containing protein,
- inner-membrane translocator,
- major facilitator superfamily protein,
- phenylacetaldehyde dehydrogenase,
- cytochrome oxidase bd-II subunit I,
- protein phosphatase 1,
- methionyl-tRNA formyltransferase,

Το τρίτο k-mer για την αλεπού, το «CTCCCCGAA» βρίσκεται στο πλασμίδιο pKPHS1 σε μια υποθετική πρωτεΐνη και στο χρωμόσωμα 19 φορές που αναγράφονται παρακάτω. Η 13^{η} εμφάνιση βρίσκεται σε κενή περιοχή.

- rRNA-23S ribosomal RNA,
- formate dehydrogenase accessory protein,
- rRNA-23S ribosomal RNA,
- rRNA-23S ribosomal RNA,
- rRNA-23S ribosomal RNA,
- rRNA-23S ribosomal RNA,
- hypothetical protein,
- rRNA-23S ribosomal RNA,
- cytosine permease,
- coenzyme PQQ synthesis protein F,
- putative carboxylesterase,
- rRNA-23S ribosomal RNA,
- formate hydrogen-lyase transcriptional activator,
- LOG family protein,
- hypothetical protein,
- 50S ribosomal protein L11 methyltransferase,
- Roi protein,
- rRNA-23S ribosomal RNA

5. Συζήτηση

Για αρχή κάποια σχόλια σχετικά με κάποιες από τις προηγούμενες διαδικασίες. Πρώτα, στο κεφάλαιο 4.3 Recursive Feature Elimination γίνεται η σύγκριση των συνόλων που έβγαλε το Boruta μεταξύ τους. Ένα σχόλιο είναι πως τα σύνολα 470 και 730 προέρχονται από το ίδιο dataset και ειδικότερα το 470 προέρχεται από το 730. Επίσης θεωρήθηκαν σημαντικά από το Boruta το οποίο βασίζεται σε Random Forest όπως και το RFE. Παράλληλα στο μοντέλο εφαρμόζεται μεταβλητή που παράγει τυχαιότητα στα αποτελέσματα. Επομένως μπορεί τα αποτελέσματα να μην σημαίνουν απαραίτητα ότι το dataset 470 έχει καλύτερα features αλλά επειδή είναι πιο μικρό, είναι πιο πιθανό να παίρνει πιο συχνά τα σημαντικά features και να βγάζει έτσι καλύτερα αποτελέσματα.

Ένα άλλο σχόλιο είναι σχετικά με την μέθοδο Random Forest στο κεφάλαιο 4.6 και 4.7. Κατά τη διαδικασία αναζήτησης των βέλτιστων παραμέτρων γίνεται μια επανάληψη χρησιμοποιώντας διάφορα mtry και διάφορα σύνολα δέντρων. Αποθηκεύεται το μικρότερο error, σε ποιο mtry βρέθηκε και σε ποιο ntree ανήκει. Όμως ορίζοντας τις παραμέτρους αυτές δεν θα δώσει αυτό ακριβώς το error rate που βρήκε. Αυτό συμβαίνει διότι υπάρχει ο παράγοντας της τυχαιότητας αλλά και επειδή οι παράμετροι βγάζουν πολύ κοντινά error rates ανά σύνολο δέντρων (Εικόνα 28 , Εικόνα 34). Χρήσει αυτών των παραμέτρων, εκτελείται το μοντέλο πάνω από μία φορά μέχρι την εμφάνιση των αποτελεσμάτων που έχουν error rate όσο πιο κοντά γίνεται σε αυτό που έβγαλε κατά την αναζήτηση των βέλτιστων παραμέτρων. Επομένως, πρακτικά δεν υπάρχουν σταθερά συγκεκριμένοι βέλτιστοι παράμετροι αλλά κατά προσέγγιση χρησιμοποιούνται αυτοί που μπορεί να δίνουν τα καλύτερα αποτελέσματα. Αυτό ισχύει και για τη μοντελοποίηση του OvA Random Forest στα κεφάλαια 4.8, 4.9, όπου η προσέγγιση της μεθοδολογίας είναι η ίδια απλά εκτελείται για κάθε οργανισμό ξεχωριστά.

Τελευταίο θέμα είναι η συζήτηση των αποτελεσμάτων της αναζήτησης των πιο σημαντικών k-mers κάθε οργανισμού μέσω της διαδικασίας BLAST που περιγράφεται στο κεφάλαιο 4.16 BLAST. Μια πρώτη ματιά δείχνει πως καθένα από τα περισσότερα κμερς των οργανισμών συναντάται σε πολλά μέρη του γονιδιώματος του βακτηρίου, είτε αυτό είναι στα πλασμίδια είτε στο χρωμόσωμα. Η ανάλυση των εμφανίσεων αυτών με τα παρόντα δεδομένα που υπάρχουν μπορεί να ξεκαθαρίσουν τη σημασία των k-mers αυτών.

Σε κάθε οργανισμό τα κμερς βρίσκονται στο χρωμόσωμα ή και σε πλασμίδια. Στις περισσότερες εμφανίσεις τους βρίσκονται σε περισσότερα από ένα γονίδια με εξαίρεση των k-mers «CGAGTCTAG» του ανθρώπου και «GGACCCCCC» των βοοειδών. Οι πρωτεΐνες που κωδικοποιούνται είναι η «Class A Carbapenemase Kpc-2» και η «fumarate reductase». Αυτό σημαίνει πως η παρουσία των πρωτεϊνών αυτών είναι πιθανό να σχετίζεται με την αλληλεπίδραση του βακτηρίου με τους οργανισμούς αυτούς. Αυτό όμοια ισχύει και για τη παρουσία των k-mers στο γονιδίωμα του βακτηρίου. Επομένως για μεταγενέστερες μελέτες, πιθανώς να μπορεί να χρησιμοποιηθεί οποιοσδήποτε από τους δύο δείκτες, k-mer ή πρωτεΐνη.

Κάτι άλλο που παρατηρήθηκε είναι στο γουρούνι, όπου η παρουσία των k-mers ήταν επαναλαμβανόμενη στο γονιδίωμα σε πολύ μεγάλο βαθμό. Αυτό από μόνο του ίσως να μπορεί να θεωρηθεί δείκτης αλληλεπίδρασης ξενιστή – βακτηρίου. Θα μπορούσε φυσικά, όπως στα υπόλοιπα να είναι σημαντική κάποια συγκεκριμένη εμφάνιση στο γονιδίωμα όμως είναι σημαντικό να υπολογιστεί ως πιθανότητα να είναι παράγοντας η συχνότητα εμφάνισης.

Επίσης, κάτι που μπορεί να υποδεικνύει γιατί ένα k-mers είναι σημαντικό για το κάθε οργανισμό, είναι η παρουσία του σε γονίδια κατ' επανάληψη ή γονίδια που παράγουν την ίδια πρωτεΐνη. Αυτό μπορεί να παρατηρηθεί στους οργανισμούς άλογο, σκύλος, γάτα και αλεπού. Δεν είναι όμως καθοριστικός δείκτης διότι μπορεί το ίδιο γονίδιο ή πρωτεΐνη να εμφανίζονται σε άλλους οργανισμούς. Ίσως η παρουσία γονιδίου ή πρωτεΐνης κατ' επανάληψη με συνδυασμό τη μοναδικότητα του έναντι των άλλων οργανισμών να είναι καλύτερος κριτής.

Τέλος, καλό είναι να ληφθεί υπόψιν κάτι τελευταίο. Βάση των αναλύσεων που έγιναν με τις μεθόδους, με το Random Forest και το One vs All Random Forest (κεφ. 4.6, 4.7, 4.8, 4.9) βγήκαν κάποια ποσοστά αποτυχημένης πρόβλεψης των οργανισμών, δηλαδή η ακρίβεια. Κανένας από τους οργανισμούς δεν έβγαλε τέλειο αποτέλεσμα και κάποιοι οργανισμοί είχαν χειρότερα αποτελέσματα από άλλους. Επίσης τα κμερς που χρησιμοποιήθηκαν στην ανάλυση με το BLAST δεν ήταν πάντα η πρώτη επιλογή κάθε μεθόδου όπως φάνηκε στις εικόνες στο κεφάλαιο 4.15. Γι' αυτό χρειάζεται να σημειωθεί πως τα κμερς αυτά δεν είναι οι απόλυτοι δείκτες αλλά θα μπορούσαν να χαρακτηριστούν έτσι σε διαφορετικά επίπεδα εμπιστοσύνης το καθένα.

6. Συμπεράσματα

Κλείνοντας την εργασία η γενική εικόνα που μένει είναι πως χρησιμοποιώντας μεθόδους μηχανικής μάθησης και βιολογικών δεδομένων στη μορφή k-mer, μπορούν να βρεθούν πιθανοί βιοδείκτες που συσχετίζονται με τις αλληλεπιδράσεις ξενιστή-παθογόνου μικροοργανισμού. Είναι μια καλή προσέγγιση που βγάζει αποτελέσματα και είναι πιθανό να μπορεί να χρησιμοποιηθεί και για περεταίρω μελέτες.

Κάποια επιπλέον πράγματα που θα μπορούσαν να ελεγχθούν για την επιβεβαίωση και ενίσχυση της μεθόδου είναι τα εξής. Για αρχή το μέγεθος των k-mers. Για αυτή την εργασία χρησιμοποιήθηκαν κμερς μεγέθους 9 νουκλεοτιδικών βάσεων. Αυτό οδηγεί σε μεγάλο όγκο δεδομένων όπως φάνηκε στο κομμάτι της μείωσης των δεδομένων αλλά και στα αποτελέσματα, που ο όγκος εμφάνισης των 9μερών στο γονιδίωμα ήταν μεγάλος. Αυτό οδηγεί και σε μερικώς ασαφή αποτελέσματα δεδομένου πως δεν είναι πολύ εύκολη η αναγνώριση του παράγοντα που ορίζει ένα k-mer ως σημαντικό στην αλληλεπίδραση με τον οργανισμό. Παρόλα τα κμερς όντως αναγνωρίστηκαν μεταξύ άλλων οπότε το μήκος των k-mers δεν είναι κάτι που παρουσιάζει πρόβλημα. Ίσως δοκιμάζοντας κμερς μεγαλύτερου μήκους να βγουν σαφέστερα αποτελέσματα.

Η διαδικασία επιλογής των χαρακτηριστικών είναι από τα πιο χρονοβόρα βήματα της μεθοδολογίας όχι μόνο για την μελέτη και επιλογή των μεθόδων όσο και για την εκτέλεση των αλγορίθμων και την αναμονή των αποτελεσμάτων. Είναι όμως αντίστοιχα ένα πολύ σημαντικό βήμα για τη πορεία της ανάλυσης. Στην εργασία αυτή παρουσιάστηκαν οι μέθοδοι που χρησιμοποιήθηκαν για τη μείωση των χαρακτηριστικών και τη ταξινόμηση. Οι μέθοδοι αυτοί περιστρέφονται γύρω από την ίδια τεχνική, το Random Forest/ Decision Trees. Κάτι που θα πρότεινα για μελλοντική έρευνα θα ήταν να δοκιμαστούν άλλες μέθοδοι επιλογής χαρακτηριστικών σε συνδυασμό με άλλες μεθόδους ταξινόμησης. Οι μέθοδοι μείωσης των χαρακτηριστικών μπορεί να είναι όμοιες με τις μεθόδους ταξινόμησης όπως της εργασίας αυτής ή γενικότερες για να πιάνουν μεγαλύτερο εύρος μεθόδων ταξινόμησης. Οι μέθοδοι ταξινόμησης επίσης θα μπορούσαν να είναι διαφόρων κατηγοριών αλλά και να μελετηθούν συνδυαστικά ως ensemble για καλύτερα ενδεχομένως αποτελέσματα.

Κάτι που θα μπορούσε να προστεθεί εδώ είναι το μέτρο σύγκρισης των αποτελεσμάτων. Στη μελέτη αυτή χρησιμοποιήθηκε το ποσοστό αποτυχημένης κατηγοριοποίησης των δειγμάτων σε οργανισμούς του Random Forest. Για την

παρούσα εργασία που έχει σκοπό να ξεχωρίσει σημαντικά χαρακτηριστικά για τον κάθε οργανισμό ίσως να είναι ένα εξίσου καλό μέτρο σύγκρισης των μεθόδων / αποτελεσμάτων η ευαισθησία (sensitivity). Αυτό μετράει το ποσοστό των πραγματικών θετικών, δηλαδή το πόσο καλή είναι η μέθοδος να προβλέπει σωστά τους οργανισμούς.

Τέλος για τα δεδομένα, θα μπορούσε να δοκιμαστεί η μέθοδος σε δεδομένα με περισσότερα δείγματα. Αυτό ίσως βγάλει καλύτερα αποτελέσματα με την ιδέα πως θα είναι πιθανώς πιο ισορροπημένα και θα χαθεί έτσι το μικρό θέμα με το bias και skewness των αποτελεσμάτων.

Συνοψίζοντας, η χρήση 9μερών σε μεθόδους μηχανικής μάθησης με την επεξεργασία τους και την μελέτη τους και τη σύγκριση των δεδομένων και των αποτελεσμάτων έβγαλε ως αποτέλεσμα 24 9-μερή που μπορεί να χρησιμοποιηθούν ως βιοδείκτες για την αναγνώριση δειγμάτων του βακτηρίου Klebsiella pneumoniae για ξενιστές προέλευσης μεταξύ των οργανισμών άνθρωπος, βοοειδή, πουλιά, άλογο, σκύλος, γάτα και αλεπού. Οι μέθοδοι ταξινόμησης που χρησιμοποιήθηκαν είναι ο Random Forest και οι μέθοδοι μείωσης των χαρακτηριστικών είναι το Boruta, το Recursive Feature elimination και εν μέρει συσχέτιση και διακύμανση. Ως αναπαράσταση παρουσιάζονται τα δεδομένα με μεθόδους μείωσης των διαστάσεων, κυρίως MDS και διαγράμματα line plots, bar plots, box plots, confusion matrices, Venn diagrams και άλλα. Η υλοποίηση έγινε μέσω της γλώσσας R στο Rstudio και ο κώδικας βρίσκεται στο GitHub στον παρακάτω σύνδεσμο.

https://github.com/XrysaM/KlebsiellaML/blob/master/code.R

7. Βιβλιογραφία

feature selection:

[1] https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00327-4

[2] https://neptune.ai/blog/feature-selection-methods

[3] https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

[4] https://towardsdatascience.com/intro-to-feature-selection-methods-for-data-science-4cae2178a00a#:~:text=There%20are%20three%20types%20of,%2C%20Ridge%2C%20Decision%20Tree).

[5] http://www.feat.engineering/classes-of-feature-selection-methodologies.html

Filter:

[6] https://www.sciencedirect.com/science/article/pii/S016794731930194X

Boruta:

[7] https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/

Rfe

[8] https://towardsdatascience.com/effective-feature-selection-recursive-feature-elimination-using-r-148ff998e4f7#:~:text=First%2C%20it%20builds%20a%20model,%2C%20accuracy%2C%20and%20Kappa).

Ml (classification theory)

[9] PatternRecognitionandMachineLearning_BishopC_Springer2006 – intro

[10] https://machinelearningmastery.com/types-of-classification-in-machine-learning/

[11] https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/

[12] https://machinelearningmastery.com/what-is-imbalanced-classification/

[13] https://www.datacamp.com/blog/classification-machine-learning

Decision trees & Random Forest

[14] https://www.youtube.com/watch?v=_L39rN6gz7Y

[15] https://www.youtube.com/watch?v=J4Wdy0Wc_xQ

[16] https://github.com/StatQuest/random_forest_demo/blob/master/random_forest_demo.R

Klebsiella

[17] *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3302456/*

[18] *https://www.genome.jp/kegg-bin/show_organism?org=kpm*

[19] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4981674/