INTER-FACULTY PROGRAMME OF MASTER STUDIES in
COMPLEX SYSTEMS and NETWORKS
DEPARMENT OF MATHEMATICS
DEPARTMENT OF BIOLOGY
DEPARTMENT OF GEOLOGY
DEPARTMENT OF ECONOMIC STUDIES
ARISTOTLE UNIVERSITY OF THESSALONIKI

MASTER THESIS

Statistical analysis of epigenetic data for CLL patients

Chalkidis Nestoras

Supervisor**:** Antoniou Ioannis
Professor A.U.TH

Thessaloniki, November 2016

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ στα
ΠΟΛΥΠΛΟΚΑ ΣΥΣΤΗΜΑΤΑ και ΔΙΚΤΥΑ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΤΜΗΜΑ ΒΙΟΛΟΓΙΑΣ
ΤΜΗΜΑ ΓΕΩΛΟΓΙΑΣ
ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Στατιστική ανάλυση επιγενετικών δεδομένων ασθενών με Χρόνια Λεμφοκυτταρική Λευχαιμία

Χαλκίδης Νέστωρας

ΕΠΙΒΛΕΠΩΝ: Αντωνίου Ιωάννης
Καθηγητής Α.Π.Θ.

Θεσσαλονίκη, Νοέμβριος 2016

INTER-FACULTY PROGRAMME OF MASTER STUDIES in
COMPLEX SYSTEMS and NETWORKS
DEPARMENT OF MATHEMATICS
DEPARTMENT OF BIOLOGY
DEPARTMENT OF GEOLOGY
DEPARTMENT OF ECONOMIC STUDIES
ARISTOTLE UNIVERSITY OF THESSALONIKI

MASTER THESIS

Statistical analysis of epigenetic data for CLL patients

Chalkidis Nestoras

Supervisor**:** Antoniou Ioannis
Professor A.U.TH

Approved by the Three-member Assessment Committee.

……	……	……
Moysiadis Theodoros Post-doctoral Researcher, INAB/CERTH	Antoniou Ioannis Professor A.U.TH	Sgardelis Stefanos Professor A.U.TH

Thessaloniki, November 2016

……..
Nestoras Chalkidis
Graduate of the Department of Mathematics of A.U.TH

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ στα

ΠΟΛΥΠΛΟΚΑ ΣΥΣΤΗΜΑΤΑ και ΔΙΚΤΥΑ

ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΤΜΗΜΑ ΒΙΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΓΕΩΛΟΓΙΑΣ

ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Στατιστική ανάλυση επιγενετικών δεδομένων ασθενών με Χρόνια Λεμφοκυτταρική Λευχαιμία

Χαλκίδης Νέστωρας

ΕΠΙΒΛΕΠΩΝ: Αντωνίου Ιωάννης
Καθηγητής Α.Π.Θ.

Εγκρίθηκε από την Τριμελή Εξεταστική Επιτροπή την 8η Νοεμβρίου 2016.

……	……	……
Θ. Μωυσιάδης	Ι. Αντωνίου	Σ. Σγαρδέλης
Μεταδιδακτορικός	Καθηγητής Α.Π.Θ.	Καθηγητής Α.Π.Θ.
Ερευνητής, ΙΝΑΒ/ΕΚΕΤΑ		

Θεσσαλονίκη , Νοέμβριος 2016

…….

Νέστωρας Χαλκίδης

Πτυχιούχος Μαθηματικός Α.Π.Θ.

ABSTRACT

Chronic lymphocytic leukemia (CLL) is the most common type of leukemia in the western world. Many patients who suffer from CLL, will be in need at some point to receive treatment. Despite the existence of several effective therapies for CLL, like the FCR regimen, many patients initially responding to the treatment eventually relapse, underscoring a characteristic resistance of the disease to the existing therapeutic options. In this thesis, we studied the temporal patterns of DNA methylation of 40 patients with CLL. Sampling was performed before and after the relapse of patients. Due to the remarkable clinical heterogeneity of CLL, the patients were initially divided into two groups with two different ways: a) based on the time to relapse, which is calculated from the start of treatment until the relapse time, and b) based on the diversification of their epigenetic patterns in recurrence compared with treatment initiation. The aim of this study is to find the most important CpG sites of DNA methylation that could be used to efficiently classify the patients. Due to the high dimensionality of our real data (40x463442), we chose to work with machine learning and variable selection algorithms. The analysis of this study is mainly based on the random forest algorithm. The latter is suitable for microarray data because it shows good predictive accuracy even when most of the predictive variables are noise, and can be used in problems where the number of variables is much larger than the number of samples/observations. Furthermore, the variable selection algorithm was applied, to detect the most informative DNA methylation sites that achieve good predictive accuracy as well. Our experimental analysis has shown that the derived DNA methylation sites can efficiently classify the patients with high success rates. Moreover, these DNA methylation sites were used to evaluate standard methods, such as hierarchical clustering (HC) and principal component analysis (PCA). It turned out that when the derived sites were used as inputs in HC and PCA, the patients were clustered satisfactorily according to their original classes.

Key Words

Analysis of epigenetic data, classification, hierarchical clustering, prediction, random forest.

ΠΕΡΙΛΗΨΗ

Η Χρόνια Λεμφοκυτταρική Λευχαιμία (ΧΛΛ) είναι η πιο συχνή μορφή λευχαιμίας στο δυτικό κόσμο. Σε πολλές από τις περιπτώσεις ασθενών με ΧΛΛ θα χρειαστεί να χορηγηθεί θεραπεία. Μολονότι υπάρχουν διάφορες αποτελεσματικές θεραπείες για τη ΧΛΛ, όπως η αγωγή FCR, πολλοί ασθενείς, αν και αρχικά ανταποκρίνονται στη θεραπεία τελικά υποτροπιάζουν, κάτι που τονίζει τη χαρακτηριστική αντίσταση της νόσου στις υπάρχουσες θεραπείες. Στην παρούσα εργασία μελετήθηκαν τα διαχρονικά πρότυπα μεθυλίωσης του DNA σε 40 περιπτώσεις ασθενών με ΧΛΛ. Η δειγματοληψία πραγματοποιήθηκε πριν την υποτροπή και κατά την υποτροπή των ασθενών. Εξαιτίας της αξιοσημείωτης κλινικής ετερογένειας της ΧΛΛ, οι ασθενείς χωρίστηκαν εξαρχής σε δύο ομάδες με δύο διαφορετικούς τρόπους: α) με βάση το χρόνο που μεσολάβησε από την έναρξη της θεραπείας μέχρι την υποτροπή, και β) με βάση τη διαφοροποίηση των επιγενετικών προτύπων τους κατά την υποτροπή σε σχέση με την έναρξη της θεραπείας. Στόχος της παρούσας εργασίας είναι η ανεύρεση των πιο σημαντικών θέσεων μεθυλίωσης του DNA που θα μπορούσαν να χρησιμοποιηθούν για την πρόβλεψη της ομάδας στην οποία ανήκει ο ασθενής. Λόγω του μεγάλου όγκου των δεδομένων (40x463442), επιλέξαμε να δουλέψουμε με αλγόριθμους επιλογής και μάθησης. Η κύρια ανάλυση της παρούσας εργασίας βασίζεται στον αλγόριθμο των τυχαίων δασών. Ο αλγόριθμος των τυχαίων δασών είναι κατάλληλος για δεδομένα μικροσυστοιχιών επειδή δείχνει καλή προγνωστική ακρίβεια ακόμη και όταν οι περισσότερες μεταβλητές παρουσιάζουν θόρυβο. Μπορεί επίσης να χρησιμοποιηθεί σε προβλήματα όπου ο αριθμός των μεταβλητών είναι πολύ μεγαλύτερος από τον αριθμό των δειγμάτων/παρατηρήσεων. Ειδικότερα, εφαρμόστηκε ο αλγόριθμος επιλογής με σκοπό την εύρεση των πιο σημαντικών θέσεων μεθυλίωσης του DNA, οι οποίες έχουν καλή προβλεπτική σημασία. Τα αποτελέσματα της εφαρμογής του αλγορίθμου επιλογής οδήγησαν σε σημαντικές θέσεις μεθυλίωσης του DNA, με βάση τις οποίες είναι δυνατή η πρόβλεψη της ομάδας στην οποία ανήκει ο ασθενής με πολύ μεγάλα ποσοστά επιτυχίας. Επιπλέον, οι θέσεις αυτές χρησιμοποιήθηκαν για την εφαρμογή μεθόδων κατηγοριοποίησης, όπως η ιεραρχική κατηγοριοποίηση (hierarchical clustering) και η ανάλυση κυρίων συνιστωσών (principal component analysis) και παρατηρήθηκε πολύ ικανοποιητική κατηγοριοποίηση των ασθενών στις κλάσεις τους.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Ανάλυση επιγενετικών δεδομένων, ιεραρχική κατηγοριοποίηση, πρόβλεψη, ταξινόμηση, τυχαίο δάσος.

Contents

ΣΥΝΟΨΗ

Η ΧΛΛ είναι η πιο γνωστή μορφή λευχαιμίας στο δυτικό κόσμο. Εμφανίζεται με συχνότητα 1:7500, και περίπου 5000 άνθρωποι πεθαίνουν λόγω της συγκεκριμένης ασθένειας κάθε χρόνο. Η ΧΛΛ παρουσιάζεται κυρίως στους ενήλικες γιατί οι περισσότεροι που έχουν διαγνωστεί πρόσφατα με τη συγκεκριμένη ασθένεια είναι πάνω από την ηλικία των 50 ετών και κυρίως άνδρες. Μερικοί ασθενείς, που είχαν διαγνωστεί με ΧΛΛ, επιβιώνουν πολλά χρόνια χωρίς θεραπεία, και πεθαίνουν από άλλες αιτίες. Ωστόσο, άλλοι ασθενείς έχουν επιθετική νόσο και πεθαίνουν σύντομα. Σε σύγκριση με τα φυσιολογικά κύτταρα, στη ΧΛΛ έχουν βρεθεί αλλοιωμένα πρότυπα μεθυλίωσης του DNA. Η μεθυλίωση του DNA περιλαμβάνεται στα επιγενετικά φαινόμενα και είναι μία χημική τροποποίηση στο δινουκλεοτίδιο CG (CpG) που έχει σαν αποτέλεσμα την αλλαγή της διαμόρφωσης του DNA. Η υπομεθυλίωση του DNA σχετίζεται με ενεργά μεταγραφικό γονίδιο, ενώ η υπερμεθυλίωση σχετίζεται με μεταγραφική αποσιώπηση.

Ο όρος επιγενετική δόθηκε από τον Conrad Hal Waddington το 1942 και περιγράφει επιγενετικές τροποποιήσεις, οι οποίες μπορούν να επηρεάσουν την έκφραση του γονιδίου χωρίς να παρουσιάζεται παράλληλα κάποια αλλαγή στην αλληλουχία των νουκλεοτιδίων του DNA. Ο ορισμός για την επιγενετική παραμένει ακόμη διφορούμενος. Παρόλα αυτά, χρησιμοποιείται για την περιγραφή γεγονότων τα οποία ρυθμίζουν διεργασίες που επηρεάζουν το DNA. Η καλύτερη επιγενετική μελέτη στην ΧΛΛ είναι η μεθυλίωση του DNA. Η διαχρονική ανάλυση της μεθυλίωσης του DNA στην ΧΛΛ είναι αντικείμενο μελέτης τα τελευταία χρόνια, συγκρίνοντας πρότυπα μεθυλίωσης του DNA κατά την διάγνωση και εξέλιξη της νόσου όσο και μετά την χορήγηση θεραπείας (Cahill, Bergh et al. 2013, Landau, Clement et al. 2014, Oakes, Claus et al. 2014).

Τα δεδομένα πάρθηκαν από την πλατφόρμα Infinium HumanMethylation450 BeadChip array, η οποία περιλαμβάνει 463442 θέσεις CpG. Στην παρούσα μελέτη, τα δεδομένα αποτελούνται 40 ασθενείς με ΧΛΛ, με δύο στιγμιότυπα ανά ασθενή (πριν την χορήγηση της θεραπείας και μετά την υποτροπή). Κάθε ασθενής έχει 463442 θέσεις CpG, με διακύμανση των τιμών μεθυλίωσης (βήτα τιμών=$\frac{U}{U+M+100}$) του από 0-1. Παλαιότερα η λήψη τόσο μεγάλου αριθμού θέσεων ήταν αδύνατη. Πλέον, με την ραγδαία εξέλιξη της τεχνολογίας οι βιολόγοι είναι σε θέση να έχουν στην κατοχή τους πολλές θέσεις CpG μέσω της συγκεκριμένης πλατφόρμας. Αυτό έχει ως αποτέλεσμα την προώθηση της σχετικής ερευνητικής δραστηριότητας,

δημιουργώντας παράλληλα νέες δυσκολίες λόγω του μεγάλου όγκου των δεδομένων που καλούνται να διαχειριστούν οι ερευνητές.

Σκοπός της παρούσας εργασίας, είναι η ανεύρεση των πιο σημαντικών θέσεων μεθυλίωσης του DNA που θα μπορούσαν να χρησιμοποιηθούν για την πρόβλεψη της ομάδας στην οποία ανήκει ο ασθενής. Όμως, ο μεγάλος όγκος των δεδομένων καθιστά δύσκολη την εύρεση των πιο σημαντικών θέσεων CpG και τον διαχωρισμό των ασθενών σε ομάδες με βάση τα κλινικό-βιολογικά χαρακτηριστικά τους.

Αρχικά χρησιμοποιήσαμε διερευνητικά διάφορες μεθόδους. Μια από τις μεθόδους που χρησιμοποιήθηκε επειδή είχαμε δύο διαφορετικά στιγμιότυπα ανά ασθενή και τα δεδομένα μας δεν ακολουθούσαν κανονική κατανομή (ανά CpG) ήταν το μη παραμετρικό τεστ Wilcoxon για κάθε θέση CpG. Η μηδενική υπόθεση που εξετάστηκε ήταν H_0: τα δύο δείγματα προέρχονται από τον ίδιο πληθυσμό και εναλλακτική υπόθεση ότι ένας πληθυσμός τείνει να έχει μεγαλύτερες τιμές από τον άλλο. Στη συνέχεια κρατήσαμε τις θέσεις CpG που είχαν p.value<0.05 στις οποίες εφαρμόστηκε η μέθοδος του «ποσοστού εσφαλμένης αποδοχής» (false discovery rate-FDR), συγκεκριμένα των Benjamini-Hochberg (1995) με κατώφλι 0.1. Η επιλογή αυτή αφενός οδηγούσε σε μεγάλο αριθμό θέσεων CpG και αφετέρου οι θέσεις αυτές δεν ομαδοποιούσαν ικανοποιητικά τους ασθενείς στις κλάσεις τους. Εφαρμόσαμε την μέθοδο του FDR εξαιτίας του μεγάλου αριθμού των τεστ που έγιναν (463442) το οποίο προκαλεί και την αύξηση των false positives. Επιπλέον,εφαρμόστηκε η μέθοδος των Benjamini-Hochberg (Benjamini and Hochberg 1995) αντί της μεθόδου των Benjamini-Yekutieli (Benjamini and Yekutieli 2001) διότι η δεύτερη μέθοδος είναι πιο συντηρητική και μετά από την εφαρμογή της, με το ίδιο κατώφλι καταλήγαμε σε μηδενικό αριθμό θέσεων CpG.

Οι παραπάνω μέθοδοι που χρησιμοποιήθηκαν δεν οδήγησαν σε επιθυμητά αποτελέσματα. Για αυτό τον λόγο επιλέξαμε να εφαρμόσουμε αλγορίθμους μάθησης και επιλογής. Ο αλγόριθμος μάθησης που εφαρμόστηκε είναι το τυχαίο δάσος διότι είναι κατάλληλος για δεδομένα μικροσυστοιχιών επειδή δείχνει καλή προγνωστική ακρίβεια ακόμη και όταν οι περισσότερες μεταβλητές παρουσιάζουν θόρυβο, και μπορεί να χρησιμοποιηθεί σε προβλήματα όπου ο αριθμός των μεταβλητών είναι πολύ μεγαλύτερος από τον αριθμό των δειγμάτων (ασθενών/παρατηρήσεων). Σύμφωνα με τον αλγόριθμο αυτό, κατασκευάζονται πολλά δέντρα αποφάσεων και κάθε δέντρο ψηφίζει την επικρατέστερη κλάση. Στη συνέχεια το τυχαίο δάσος αποφασίζει/ψηφίζει με βάση όλα τα δέντρα ποια είναι η επικρατέστερη κλάση. Ο

αλγόριθμος αυτός παρέχει πολλά χρήσιμα μέτρα, μερικά από τα οποία είναι το εκτός δείγματος σφάλμα (out of bag error rate) και ένα μέτρο της σημαντικότητας της κάθε μεταβλητής (variable importance). Με την βοήθεια αυτών των μέτρων ο χρήστης μπορεί να παρατηρήσει πόσο είναι το σφάλμα πρόβλεψης του αλγορίθμου, να ανακαλύψει ποιες είναι οι μεταβλητές που παίζουν σημαντικό ρόλο για την κατασκευή του δάσους και πως αυτές καθορίζουν την κλάση στην οποία θα ταξινομηθεί ο ασθενής.

Ο αλγόριθμος του τυχαίου δάσους διακρίνεται σε δύο κατηγορίες ανάλογα με τα δεδομένα τα οποία θέλει να επεξεργαστεί ο χρήστης. Η πρώτη κατηγορία αναφέρεται σε προβλήματα ταξινόμησης (classification) και η δεύτερη σε παλινδρόμηση (regression). Παράλληλα, ο αλγόριθμος του τυχαίου δάσους μπορεί να γίνει με επιβλεπόμενη ή μη επιβλεπόμενη μάθηση. Στην επιβλεπόμενη μάθηση, ο χρήστης παρέχει σαν δεδομένα τις κλάσεις με τις οποίες θα δουλέψει ο αλγόριθμος. Αντιθέτως, στη μη επιβλεπόμενη μάθηση ο χρήστης δεν δίνει καμία πληροφορία για τις κλάσεις ως προς τις οποίες θα πρέπει να ομαδοποιηθούν τα δεδομένα του και περιμένει από τον αλγόριθμο να ταξινομήσει τις παρατηρήσεις βάσει κάποιων κοινών χαρακτηριστικών τους.

Αρχικά, εφαρμόσαμε το τυχαίο δάσος (random forest) σε όλα τα δεδομένα πριν το φιλτράρισμα. Παρόλη την υπολογιστική δύναμη που είχαμε στη διάθεση μας (server ΙΝΕΒ/ΕΚΕΤΑ) ήταν αδύνατο να εφαρμόσουμε τον αλγόριθμο του τυχαίου δάσους σε όλα τα δεδομένα, λόγω υπερβολικών απαιτήσεων μνήμης που χρειαζόταν ο αλγόριθμος εξαιτίας του όγκου των δεδομένων. Για τον λόγο αυτό, έγινε ένα φιλτράρισμα στα δεδομένα και εφαρμόστηκε στη συνέχεια μία μέθοδος επιλογής που βασίζεται στον αλγόριθμο του τυχαίου δάσους.

Η μέθοδος επιλογής που εφαρμόστηκε βασίζεται στο μέτρο της σημαντικότητας της κάθε θέσης CpG, το οποίο προκύπτει από την εφαρμογή της μεθόδου του τυχαίου δάσους (random forest). Το μέτρο της σημαντικότητας δείχνει πόσο σημαντικός είναι ο ρόλος της κάθε θέσης CpG κατά την διαδικασία της ομαδοποίησης (classification). Όλες οι παραπάνω μέθοδοι που αναφέρθηκαν δεν οδήγησαν σε ένα ρεαλιστικά μικρό αριθμό θέσεων CpG, οι οποίες ταυτόχρονα να έχουν την δυνατότητα να ομαδοποιήσουν τους ασθενείς στην κατάλληλη κλάση τους βάσει των κλινικό-βιολογικών χαρακτηριστικών τους.

Ο στόχος ήταν να καταλήξουμε σε λίγες και σημαντικές θέσεις CpG, διότι αυτό θα επέτρεπε τη μετέπειτα πειραματική ανάλυση σε βιολογικά εργαστήρια. Η ανάλυση αυτή έχει υψηλό κόστος και στοχεύει στη διερεύνηση των γονιδίων που

συνδέονται με τις τελικές θέσεις CpG . Ο αλγόριθμος επιλογής είναι ο κατάλληλος γιατί συνδυάζει παράλληλα μέσω του τυχαίου δάσους (random forest) την εύρεση ενός μικρού αριθμού θέσεων CpG τα οποία όμως ομαδοποιούν ικανοποιητικά τους ασθενείς μας. Αυτό είναι κάτι πολύ σημαντικό που οι άλλες μέθοδοι δεν μπόρεσαν να μας δώσουν ως αποτέλεσμα. Ένα άλλο σημαντικό πλεονέκτημα αυτής της μεθόδου που εφαρμόστηκε είναι ότι μείωνε τον αριθμό των σημαντικών θέσεων CpG και κρατούσε αυτά τα οποία ήταν πιο σημαντικά κατά την διαδικασία εφαρμογής του τυχαίου δάσους.

Αξίζει να σημειωθεί, πως βάσει της μεθόδου οι ασθενείς ομαδοποιήθηκαν ικανοποιητικότερα από οποιαδήποτε άλλη εφαρμογή μεθόδου. Βάσει βιβλιογραφίας ο αριθμός των θέσεων CpG στα οποία κατέληξε ο αλγόριθμος επιλογής είναι ικανοποιητικός συγκριτικά με άλλες δοκιμές που πραγματοποιήθηκαν σε παρόμοια δεδομένα (datasets), μέσα σε ένα εύρος τιμών από 2 έως 230 θέσεις CpG. Επιπλέον, οι πιο σημαντικές θέσεις CpG που βρέθηκαν μετά την εφαρμογή των αλγορίθμων επιλογής και μάθησης χρησιμοποιήθηκαν για την εφαρμογή γνωστών μεθόδων κατηγοριοποίησης, όπως η ιεραρχική κατηγοριοποίηση (hierarchical clustering) και η ανάλυση κυρίων συνιστωσών (principal component analysis). Εφαρμόσαμε αυτές τις μεθόδους για λόγους σύγκρισης και για να διερευνήσουμε αν όντως ο αριθμός των θέσεων CpG που βρήκαμε ομαδοποιούσε ικανοποιητικά τους ασθενείς και με αυτές τις μεθόδους. Οι μέθοδοι αυτοί χρησιμοποιήθηκαν και σε περισσότερες θέσεις CpG και έγινε σύγκριση μεταξύ αυτών και των επιλεγμένων θέσεων και παρατηρείται πως η μέθοδος μας κατέληξε σε πολύ καλά αποτελέσματα με καλή προβλεπτική σημασία. Με την εφαρμογή του συγκεκριμένου μοντέλου ασθενείς με μικρό χρόνο μέχρι την υποτροπή ή με λίγες αλλαγές στα προφίλ μεθυλίωσης του DNA θα έπρεπε ενδεχομένως να αποφύγουν τη χορήγηση της συγκεκριμένης θεραπείας.

INTRODUCTION

Chronic lymphocytic leukemia (CLL) is the type of leukemia which most affects adults in the western world. It appears with frequency 1:7500, and 5000 people approximate die every year. CLL is a disease of adults because most people newly diagnosed are over the age of 50 and the majority is men. Some patients, diagnosed with CLL, survive many years without a cure, and die from other reasons. However, other patients have aggressive disease and die shortly. The ontogeny of CLL and the origination cell remain undiscovered. CLL results in the spleen, liver and eventually anemia. Early CLL is not treated and late is treated with chemotherapy.

The term epigenetics defined by Conrad Hal Waddington in 1942 describes changes in genome function that occur without a change in nucleotide sequence within the DNA. The term epigenetics remains controversial. However, it's used to describe events, which adjust processes and these processes affect DNA. The best epigenetic study in CLL is methylation of DNA, which is a chemical modification in dinucleotide cytosine and guanine (CpG) which modifies the DNA. Longitudinal analysis of DNA methylation in CLL has been performed, only recently, in many cases, comparing the DNA methylation patterns in the diagnosis and progression of disease after the administration of treatment (Cahill, Bergh et al. 2013, Landau, Clement et al. 2014, Oakes, Claus et al. 2014). Hypermethylation, and hypomethylation are both aberrant DNA methylation patterns, and they associated with a large number of human malignancies. Hypermethylation is associated with gene inactivation and transcriptional silence. Hypomethylation is associated with active gene transcription.

Recently these problems caught the attention of the worldwide life science community. The technical challenges and the high dimensionality problems are resolved with statistical analysis, data visualization, interpretation, and storage. The problems of high dimensionality are solved with data mining and machine learning algorithms because they aim to excavate knowledge and find patterns from big datasets.

Our real data retrieved from the Infinium HumanMethylation450 BeadChip array and include 463442 CpG sites. In the past few years, biologists were able to take so many CpG sites, due to the rapid technology boom. The evolution of technology assisted the researchers and promoted research in this field. In our study, we have 40 patients with CLL. Each patient has two states, one state before the

relapse and one after the relapse. Moreover, each patient has 463442 CpG sites with range of their values (beta values= $\frac{M}{U+M+100}$) from 0 to 1.

The aim of our study is to find the most important CpG sites which have the ability to separate patients to their pre-defined classes based on their clinic-biological data before the relapse. Howbeit, the big heterogeneity and the high dimensionality of our real data (40x463442) cause memory computational/processing problems to the server. In order to overcome these problems, we have chosen to work with machine learning and variable selection algorithms.

Before the evaluation of the machine learning and variable selection algorithms, we applied several statistical methods to our real data. However, none of these methods could achieved to detect a small number of CpG sites who achieved good predictive accuracy. We applied a Wilcoxon rank sum test to our data. Wilcoxon rank sum test was selected because our data didn't follow the normal distribution (per Cpg). The null hypothesis of Wilcoxon rank sum test is H_0: two samples come from the same population against the alternative hypothesis that a particular population tends to have larger values than the other. After the evaluation of the Wilcoxon rank sum test we kept only the CpG sites with p-value<0.05.

In addition, we applied an FDR method to account for the multiple comparisons. Specifically we used the method proposed by Benjamini-Hocheberg (Benjamini and Hochberg 1995) with threshold 0.1 because it is less conservative and more appropriate than the method of Benjamini-Yekutieli (Benjamini and Yekutieli 2001). In addition, the evaluation of Benjamini-Yekutieli method arrived at zero CpG sites with the same threshold and seems to be more restrictive and conservative.

The results of the above methods weren't satisfying. In our study we evaluated the random forest algorithm because it can handle noisy variables and it is used when the number of variables is larger than the number of the samples (observations/patients) in the dataset. Random Forests are an ensemble of tree predictors such that each tree votes the best class, and the forest takes as class the majority of trees classes. The random forest algorithm provides useful measures such as the out-of-bag error (OOB) rate and the variable importance. The OOB error rate, represents the classification error of the entire random forest procedure. The variable importance shows the importance of a variable during the classification procedure on all trees.

The random forest algorithm can be applied both for regression and classification problems. In addition, the random forest algorithm can be evaluated both for supervised and unsupervised learning problems. In supervised learning problems the user provides the actual classes of observations, and the algorithm tries to learn a general rule that matches inputs to outputs via a function. On the other hand, in unsupervised learning problems the user doesn't provide any class of observations, and the algorithm tries to find structure in his input and discover patterns from data on its own.

We applied the random forest algorithm to the entire dataset, but the algorithm crashes due to large memory requirements and lack of memory capacity. For this reason, we reduced the data, in order to apply the variable selection algorithm, which is based on the random forest algorithm.

Furthermore, we applied the variable selection algorithm, which reduces the CpG sites based on their variable importance provided by random forest algorithm. At each step the algorithm keeps the most important CpG sites according to the random forest classification procedure. After the algorithm evaluation we found a small number of CpG sites who achieves good predictive accuracy. The small number of CpG sites is very important, because biologists can easier check them and find their impact. This is very important because from 463442 we conclude only to a few CpG sites with good predictive accuracy. We managed to find the relevant CpG sites who classify the patients correctly to their pre-defined classes and remove the irrelevant CpG sites.

Moreover, these CpG sites are used to compare the results with well-known methods, such as hierarchical clustering and principal component analysis. We evaluated these methods based on bibliography. The CpG sites which are selected from the variable selection algorithm tend to group the patients better to their pre-defined classes than more CpG sites. To conclude, these CpG sites except from the good predictive accuracy, tend to group the patients better than other CpG sites in more than one method. In addition these CpG sites are connected with some genes and this might be show the importance of our results. The application of this model could prevent the administration of this treatment (FCR), in patients with little time to relapse or with few changes in their DNA methylation profiles.

ACKNOWLEDGEMENTS

I would like to gratefully thank my beloved family and grandparents for their unconditionally psychological support and for encouraging me all these years. I would also like to thank my supervisor Prof. Ioannis Antoniou for all his efforts, and both the knowledge and the opportunities he provided to me during my studies. Moreover, I would like to thank Dr. Theodoros Moysiadis for his guidance during my thesis. Last but not least, I would like to thank Mrs. Maria Tsagiopoulou (PhD candidate in Biology) for the fruitful collaboration, the Institute of Applied Biosciences and its director, Kostas Stamatopoulos, for the kind hospitality and support.

1. Knowledge Discovery Techniques

1.1 Data Mining

Data Mining is a subfield of computer science. With data mining, we can discover patterns in large datasets, and involve methods of artificial intelligence, machine learning, and statistics. Data mining aims to extract knowledge and information from a large data set and transform it for further use. It tries to discover unknown and interesting patterns like groups (cluster analysis) from big data sets and use these patterns for further analysis in machine learning and predictive analytics.

As data mining can only uncover patterns presented in the data, the data must be large enough for these patterns to be uncovered. Moreover, the procedure must give results in finite time. Data cleaning removes the noisy observations and those with missing data.

According to (Fayyad, Piatetsky-Shapiro et al. 1996), data mining involves the following common tasks:

- Association rule learning – Searches for relationships and correlations between variables. For example in supermarkets, they keep track of consumer's habits and give purposeful offers in various consumer goods.
- Clustering – Is the procedure of discovering groups and structures in the data, without knowing the classes of data.
- Classification – In classification, the model knows the class of the samples it wants to predict.
- Regression – Attempts to find a function which models the data with the least error.

1.2 Machine Learning

Machine learning is part of computer science, in which computers have the ability to learn without being programmed, and is related to computational statistics, pattern recognition, and artificial intelligence. Machine learning uses algorithms that can learn and make predictions of a dataset. In the field of data analytics, machine learning is a method which is used to analyze complex models and algorithms to make predictions. These analytical models allow researchers, data scientists and analysts to take reliable results and uncover patterns and relationships in high dimensional data. Machine learning use is very important these days, e.g. the application of machine learning is typical in problems such as fraud detection, web search results, predictions and pattern/image recognition

Machine learning tasks are classified into four different categories, depending on the available information that the user gives to the learning system. These four categories are:

1. Supervised learning: The user provides the actual classes of observations, and the algorithm tries to learn a general rule that matches inputs to outputs via a function.
2. Semi-supervised learning: The user provides incomplete information about the class of observations.
3. Unsupervised learning: The user doesn't provide any class of observations, and the algorithm tries to find structure in his input and discover patterns from data on its own.
4. Reinforcement learning: It is used for robotics, gaming and navigation. With reinforcement learning, the algorithm discovers the actions which yield the greatest results.

Another categorization of machine learning depends on the desired output that user wants from a machine-learned system:

- In classification, each input is divided into two or more classes, and the model assigns the inputs to their classes based on decision learning. This categorization is often been used for supervised learning techniques.
- In regression, the outputs, and inputs are continuous variables. Regression procedure is part of supervised learning methods.

- In clustering, inputs divided into groups. The classes of samples are unknown, and this is an unsupervised learning method.
- Dimensionality reduction simplifies inputs by keeping only the most informative variables.

Common machine learning algorithms are: 1) Neural networks, 2) Decision trees, 3) Random forests, 4) Associations and sequence discovery, 5) Gradient boosting and bagging, 6) Support vector machines, 7) Nearest-neighbor mapping, 8) K-Means clustering, 9) Self-organizing maps, and 10) Principal component analysis, among others.

Data mining and machine learning often confuse the user because they apply the same methods. They can be distinguished as follows:

1. Machine learning focuses on prediction, based on pre-defined classes which the user gives for data.
2. Data mining focuses on the discovery of unknown patterns and knowledge from the data.

In our work, we use the random forest algorithm. Random forests are an ensemble of learning trees for classification and regression problems. It belongs to both fields of machine learning and data mining. In random forests the user decides to work with supervised or unsupervised learning forests by providing or not the classes of observations. In chapter 2 we will discuss in more depth how random forests work. In addition, we will introduce in chapter 3 the variable selection method, which is used to our main research. The variable selection method is based on the random forest algorithm and achieves to detect the most informative CpG sites which have good predictive accuracy as well. Finally, in chapter 4, we will present our real data application and the methods which have been used in this study.

2. Random Forests

2.1 Introduction

Random Forests are an ensemble of tree predictors such that each tree votes the best class, and the forest takes as class the majority of trees classes. Each tree in the forest is constructed by bootstrap sampling from the original dataset. Random Forests, developed by Leo Breiman (Breiman 2001), are shown to both build models with high accuracy when tested on high dimensional data and handle noisy variables. Besides, random forests are able to excavate knowledge referring to correlation between variables and interactions among them. In this chapter, we will describe how the random forests algorithm works, and present the two different methods of it. Random forests provide important metrics, which we will present and explain further down in the text. Random forests are suitable for microarray data because we have a large number of predictors and a small number of samples. On the other hand, classical statistical techniques cannot be applied directly to microarray datasets because of their high dimensionality. The dimensions of our dataset in a matrix form are 40x463442. Our motivation is to discover the most important DNA methylation sites, which have the ability to predict the actual class the patient belongs. The problem with the high dimensionality of our data is confronted with random forests algorithm. Random forests have advantages in microarray data mining problems for several reasons:

1. The classification trees are non-parametric and do not make assumptions about the underlying distribution.
2. It performs excellently with noisy variables.
3. It used commonly when the number of variables is larger than the number of the samples in the dataset.
4. It returns important measures such as variable importance, which can be used for variable selection methods.
5. Achieves excellent predictive accuracy for high dimensional genomic data.
6. Can be used for two-class and multi-class classification problems.

According to Leo Breiman (Breiman 2001), the definition regarding classification random forests is:

A random forest is a classifier consisting of a collection of structured tree classifiers $\{h(x, \Theta_k), k = 1, ...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree votes for the most popular class at input x .

According to Leo Breiman (Breiman 2001), regression random forests are formed by growing trees depending on equally distributed random vectors Θ such that the tree predictor *h*(x,Θ) takes as inputs numerical values. The output values are numerical, and we assume that the training set is independently drawn from the distribution of the random vector *Y*, X. The mean-squared generalization error for any numerical predictor *h*(x) is

$$E_{X,Y}\left(Y - h(x)\right)^2.$$

The random forest predictor is formed by averaging k trees $\left\{h\left(x, \Theta_k\right)\right\}$.

In problems such as regression, we attempt to predict the values of a continuous variable from one or more continuous variables and understand the relationship between them, e.g. we may want to predict the time when a patient gets sick again. This is a continuous variable, and we want to predict when the patient gets sick again based on some variables which have been tested.

In problems such as classification, we attempt to predict values of a categorical dependent variable (class, group membership) from one or more continuous variables and understand the relationship between them, e.g. we may want to predict if a patient is in high risk to get sick or not based on his age and weight.

To predict the value of y for a new value of x, we have to build a model based on:

- Expected mean squared error (for regression).
- Expected out of bag error estimate (for classification).

To sum up, if we want to make a prediction for a continuous variable, then we will use regression random forests. If we want to make a prediction for a categorical variable, then we will use classification random forests. This is one of their differences. The other difference that these two methods have is the splitting criterions.

- Regression: residual sum of squares

$$\mathrm{RSS} = \sum_{left} \left(y_i - y_L^{*} \right)^2 + \sum_{right} \left(y_i - y_R^{*} \right)^2$$

where y_L^{*} = mean y-value for left node

y_R^{*} = mean y-value for right node

- Classification: Gini criterion

$$\mathrm{Gini} = \mathrm{n}_L \sum_{k=1}^{K} p_{kL} \left(1 - p_{kL} \right) + \mathrm{n}_R \sum_{k=1}^{K} p_{kR} \left(1 - p_{kR} \right)$$

where p_{kL} = proportion of class k in left node

p_{kR} = proportion of class k in right node

2.2 Random Forests algorithm definition

In this section the algorithm of random forest will be introduced and an illustration of it in Figure 1.

- Let the number of samples be N, and the number of variables in the classifier be M.
- The number of input variables, m, is used to determine the decision at a node of a tree; m should be much less than M.
- Choose a training set for this tree by choosing N times with replacement (bootstrap sample) from all N available cases. Use the remaining cases to estimate the error of the tree, by predicting their classes.
- For each node of the tree, randomly choose m variables from M on which to base the decision at that node. Calculate the best split based on these m variables in the training set.

- Each tree is fully grown and not pruned.

The choice of bootstrap sampling is made because the entire process has to be random. In this way each tree is produced by a random sampling with replacement which is of equal size to and independent of the input vectors. The samples that are not included in the construction of a tree are approximately 1/3 of the total amount of samples.

The algorithm of random forests is the same for both regression and classification problems. The difference is that in classification we combine trees by voting and in regression by averaging.

We will explain how random forests work for both classification and regression problems.

- Understand how out-of-bag (OOB) error been estimated.
- Analyze the splitting criterions for classification and regression.
- Determine the best value for number of trees (*ntree*) in the forest.
- Which is the best value for m predictor variables (*mtry*) in each split.
- Understand the role of variable importance.

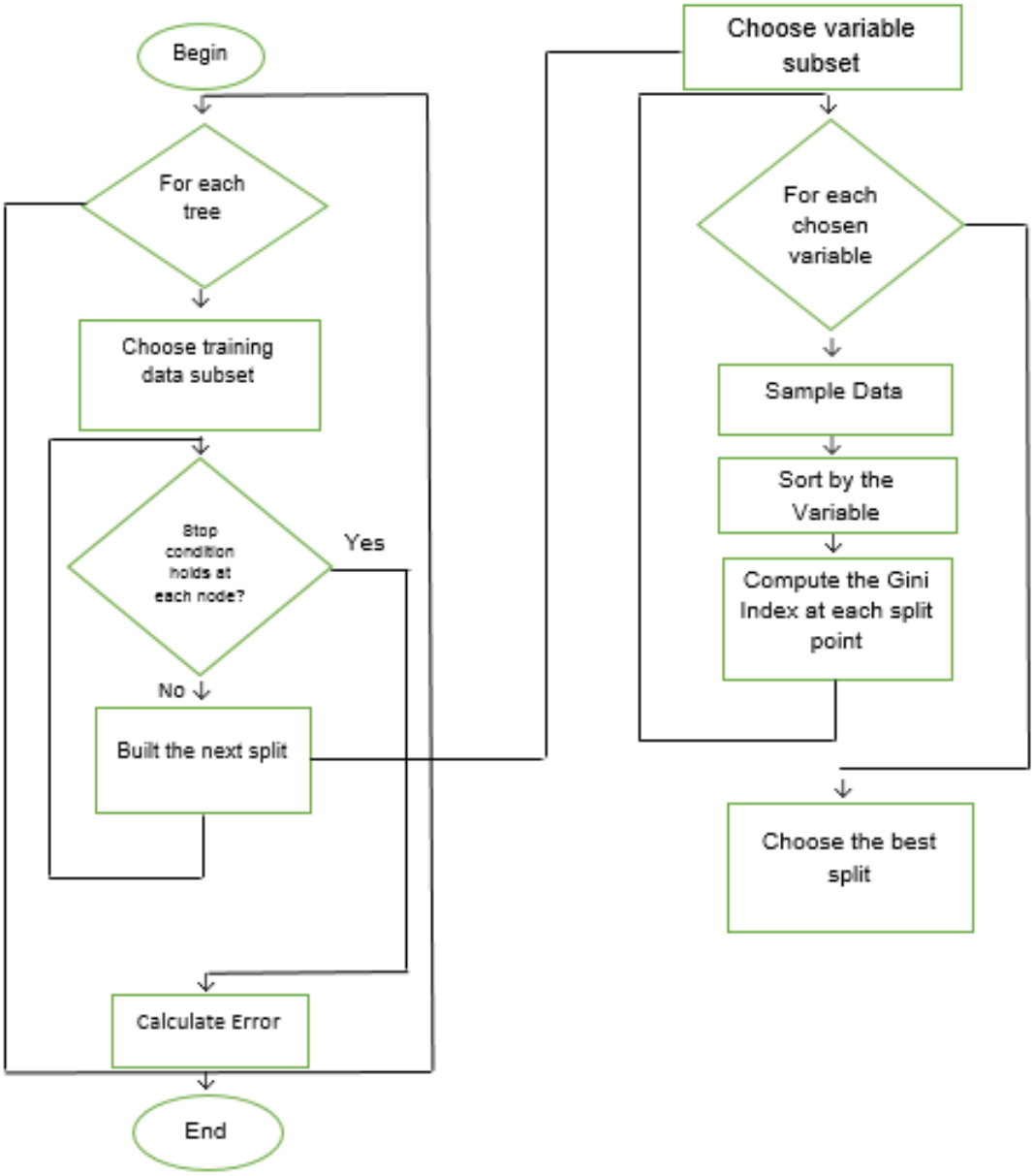


Figure 1: Algorithm Illustration.

2.3 Measured Errors

2.3.1 Out of bag error for classification forests

When the procedure begins, the N samples are separated in training and testing set. In random forests, the user isn't obligated to determine from the beginning which is the training and the test set. As we mentioned before, the bootstrap sampling method with replacement, every time creates an input vector for each tree. From the N samples, approximately the 2/3 are used for inputs in each tree, and approximately the other 1/3 of samples are not selected as input. The samples that are not included in the construction of each tree represent the testing set, and the other samples represent the training set. Hence, random forest algorithm creates from the beginning the training and the test sets.

This 1/3 of samples that are not used for the construction of the tree are used for the calculation of out of bag (OOB) error. For the error calculation we follow three steps:

- The algorithm takes a sample and searches the trees that this sample is out of bag. Then the algorithm fits in these trees the sample and these trees classify it to a class. This procedure is made for each sample.
- Let j be the class with the most votes for a sample from the classification and tc the true class of the sample.
- The percentage of times that j≠tc represents the out of bag error estimate.

The out of bag error estimate is important for a random forest because the user can understand how accurate the model is, e.g. if the OOB error is 12% it means that when the resulting model is applied on new observations, then these observations will be classified with error 12%. This means that our model is 88% accurate, which is a reasonably good model. Lower OOB error rate means that our model is more accurate to classify an observation to the right class. We have to mention that OOB error estimate depends on the value of the number of trees and the value of the m predictors. The default value of *ntree*=500 and *mtry*=$\sqrt{\text{variables}}$. Moreover, the OOB error estimate is dependent of the strength and the correlation between the trees of a forest. If two trees are correlated, then the OOB error estimate of the forest is increased. But this is very difficult to happen because with bootstrap sampling the input vector of each tree is different from the other. This leads to a

forest with non-correlated trees and a small OOB error rate. The strength of a tree means how good a classifier is the exact tree. A tree is a good classifier when the OOB error rate is small. So when we have a forest which is a good classifier then the OOB error rate should be small. In Figure 7, we present the random forest procedure and how the training and testing sets are defined.

2.3.2 Mean Squared Error (MSE) for regression forests

In regression forests the accuracy of a random forest's prediction can be obtained from the OOB data but not with the same way as classification. In regression forests OOB calculated by:

$$\mathrm{OOBMSE} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{y_i} - \hat{\mathrm{y}}_{\mathrm{iOOB}})^2,$$

where $\hat{\mathrm{y}}_{\mathrm{iOOB}}$ denotes the average prediction from ith observation from all trees where observation i was out of bag.

Analogously to linear regression the overall sum of squares is calculated by

$$\mathrm{SST} = \sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{y_i} - \bar{\mathrm{y}})^2,$$

and OOBR^2 can be obtained by

$$\mathrm{OOBR}^2 = \frac{1 - (\mathrm{OOBMSE})}{\mathrm{SST}}.$$

2.4 Splitting Criterion

2.4.1 Classification

There are three different measures chosen for classification splitting criterions, the misclassification error, information gain and the Gini Index. In random forests the Gini Index of node impurity chosen most for classification problems in contrast with measures of misclassification error and information gain. We will explain how Gini Index works and give an example of it.

Gini Index shows each variable's contribution to the homogeneity of the nodes and leaves in the resulting random forest. Homogeneity is represented with 0 and heterogeneity with 1. Higher decrease in Gini means that a particular predictor variable plays a greater role in partitioning the data into the defined classes.

If a dataset X contains examples from n classes, Gini (X) is defined as:

$$Gini(X) = 1 - \sum_{j=1}^{n} (p_j)^2 \ , \qquad (1)$$

where p_j is the relative frequency of class j in X.

If a dataset X is split into two subsets X_1 and X_2 with sizes N_1 and N_2 respectively, then this matrix contains a number of records of each class. The Gini Index X defined as:

$$Ginisplit(X) = \frac{N_1}{N} gini(X_1) + \frac{N_2}{N} gini(X_2) \ , \qquad (2)$$

the value that achieves the smallest split Gini (X) is chosen to split the node.

Example:

In this example, we will explain how Gini Index procedure works to determine the best split.

Table 1: Positions and classes of each patient.

Patients	Position1	Position2	Class
Patient1	13	10	1
Patient2	8	4	0
Patient3	29	1	0
Patient4	18	38	1
Patient5	35	44	0

From Table 1 begin by choosing the first attribute to be split which is Position1 attribute. The possible splits of the Position1 attribute in the dataset are Position1≤ 8, Position1≤ 13, Position1≤18, Position1≤ 29 and Position1≤ 35. Take the first split and calculate the Gini Index as follows:

The partitions after the binary split of Position1 ≤ 8 are given in Table 2.

Table 2: Number of records.

Attributes	Number of Records		
	Zero (0)	One(1)	N=5
Position1 ≤ 8	1	0	n_1=1
Position1 > 8	2	2	n_2=4

Then the calculations of equations (1) and (2) yields the following results.

$$Gini(\text{Position}1 \le 8) = 1 - \left(1^2 + 0^2\right) = 0 .$$

$$Gini(\text{Position}1 > 8) = 1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right) = 0.5 .$$

$$Ginisplit = \frac{1}{5} \cdot 0 + \frac{4}{5} \cdot 0.5 = 0.4 .$$

In the next step, the partitions of Position1≤ 13 after the binary split are given in Table 3.

Table 3: Number of records.

Attributes	**Number of Records**		
	Zero (0)	One (1)	N=5
Position1≤ 13	1	1	n_1=2
Position1> 13	2	1	n_2=3

With the same way, the calculations of equations (1) and (2) yields to the following results.

$$Gini(\text{Position} \le 13) = 1 - \left(\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = 0.5 .$$

$$Gini\left(\text{Position} > 13\right) = 1 - \left(\left(\frac{2}{3}\right)^2 + \left(\frac{1}{3}\right)^2\right) = 0.444 .$$

$$Ginisplit = \frac{2}{5} \cdot 0.5 + \frac{3}{5} \cdot 0.444 = 0.466 .$$

This procedure continues until we calculate the Gini from the remaining splits. The Table 4 has all the records of Gini Index for each split.

Table 4: Gini Index records.

Gini Split	Value
Ginisplit(Position1≤8)	0.4
Ginisplit(Position1≤13)	0.466
Ginisplit(Position1≤18)	0.266
Ginisplit(Position1≤29)	0.4
Ginisplit(Position1≤35)	0.48

The lowest value for Gini Split is 0.266. So we choose as split point the Position1≤18. Because the values of Position1 attribute are continuous the best way to decide the split point is to take the midpoint of every pair of consecutive values. So the split point is

$$\text{Position1} = \frac{(18+29)}{2} = \frac{47}{2} = 23.5.$$

The decision tree after the selection of the first split point is shown in Figure 2:

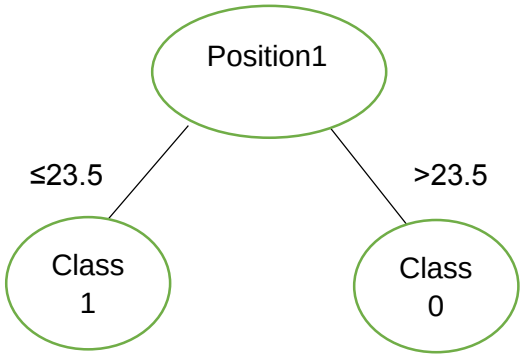


Figure 2: Decision tree after the first split.

This procedure is repeated for each attribute. The next step is to calculate the Gini Index of the attribute Position2. The procedure that we perform is the same as before. We select the lowest Gini Index as the best split for the attribute Position2 and after calculations, the value of Position2 split point is 7.

The final decision tree is shown below in figure 3:

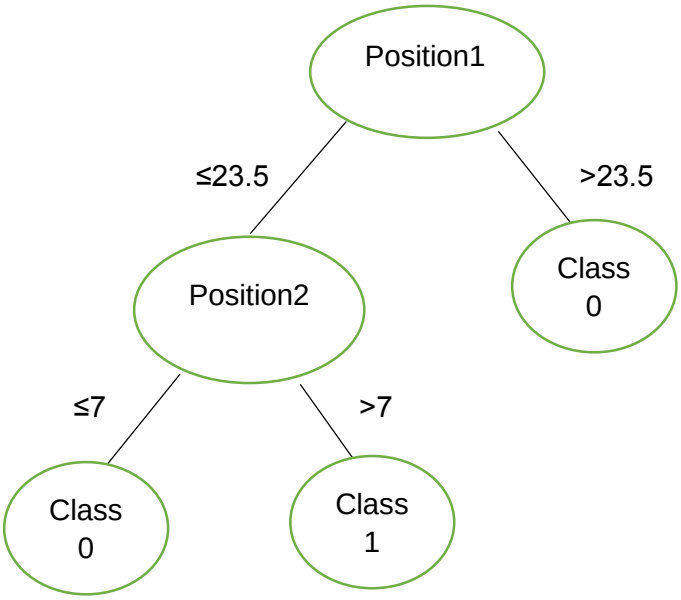


Figure 3: Final decision tree.

This is a single tree construction with the Gini Index method for classification. Random forest algorithm uses this method to find the best splits and constructs many trees using different sets of attributes for each tree. The procedure that random forests follows is random because the samples are sampled with bootstrap sampling (sampling with replacement) and the variables which are chosen to find the best split are chosen randomly from the entire set of variables.

2.4.2 Regression

The trees that the random forest algorithm produce are constructed in the same way as the classification and regression trees (CART). For the regression procedure, the split is based on the reduction of the residual sum of squares. The steps that the algorithm is following are:

- Beginning with the root node, CART finds the best variable to split the node into two child nodes. The splits of the predictor variables are binary with $X_j \leq s$ and $X_j > s$.

- In order to find the best variable, all variables and all values of them been checked. The selected variable has to minimize the impurity measure in the nodes, the residual sum of squares.
- Recursively continue with step one and two on the descendant nodes until the homogeneity of the nodes cannot be improved any more.

For regression problems, the terminal nodes are calculated by averaging the response variables.

Example:

In this example, we will explain how residual sum of squares works to determine the best split for a variable in Tabe 5.

Table 5: Positions and time to relapse of patients.

Patients	Position1	Position2	Time to relapse
Patient1	2	8	11
Patient2	4	9	3
Patient3	5	6	5
Patient4	7	5	9

At first, we begin with the variable Position1, and try to find the best value which minimizes the residual sum of squares.

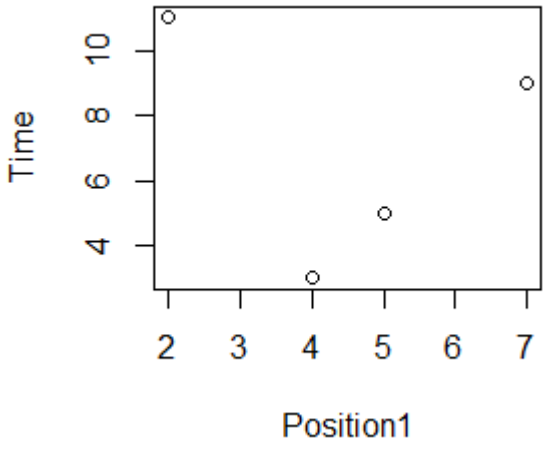


Figure 4: A plot of Position1 and Time to relapse values.

We start with Position1=2 then the residual sum of squares is:

$$\mathrm{RSS} = \sum_{x \le 2} (y_i - \bar{y})^2 + \sum_{x > 2} (y_i - \bar{y})^2,$$

$$\mathrm{RSS} = (11 - 11)^2 + (3 - 5.6)^2 + (5 - 5.6)^2 + (9 - 5.6)^2 = 18.68.$$

For Position1=4 the residual sum of squares is:

$$\mathrm{RSS} = \sum_{x \le 4} (y_i - \bar{y})^2 + \sum_{x > 4} (y_i - \bar{y})^2,$$

$$\mathrm{RSS} = (11 - 7)^2 + (3 - 7)^2 + (5 - 7)^2 + (9 - 7)^2 = 40.$$

For Position1=5 the residual sum of squares is:

$$\mathrm{RSS} = \sum_{x \le 5} (y_i - \bar{y})^2 + \sum_{x > 5} (y_i - \bar{y})^2,$$

$$\mathrm{RSS} = (3 - 6.3)^2 + (5 - 6.3)^2 + (11 - 6.3)^2 + (9 - 9)^2 = 34.67.$$

For Position1=7 the residual sum of squares is:

$$\mathrm{RSS} = \sum_{x \le 7} (y_i - \bar{y})^2 + \sum_{x > 7} (y_i - \bar{y})^2,$$

$$\mathrm{RSS} = (11 - 7)^2 + (3 - 7)^2 + (5 - 7)^2 + (9 - 7)^2 = 40.$$

Table 6: Residual Sum of Squares for each variable.

RSS Split	RSS
Position1=2	18.68
Position1=4	40
Position1=5	34.67
Position1=7	40

Therefore, we conclude from Table 6 that the variable Position1=2 achieves the best split because it minimizes the residual sum of squares. The decision tree after the selection of the first split is shown in Figure 5:

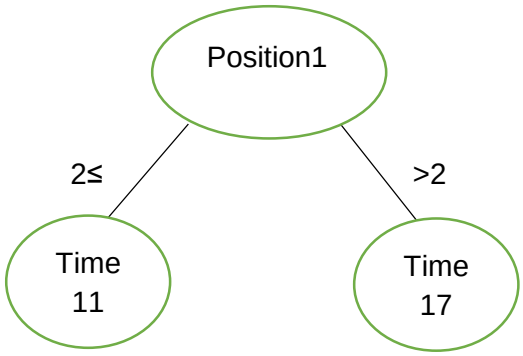


Figure 5: Decision tree after the first split.

This procedure is repeated for each attribute. The next step is to calculate the Residual Sum of Squares of the attribute Position2. The procedure that we perform is the same as before. We select the lowest Residual Sum of Squares as the best split for the attribute Position2 and after calculations, the value of Position2 split point is 5. The final decision tree is shown in the Figure 6:

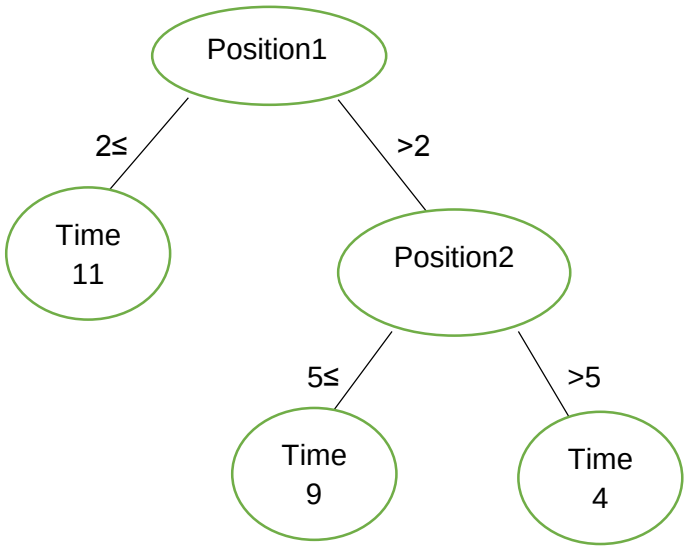


Figure 6: Final decision tree.

This procedure stops when the nodes become leafs. One node is terminal when:

- All the training data in the node are of the same class. This class becomes the class of this node and the node named pure node.
- During the tree construction, in some depth of the tree, the training instances which we test can't be split further because all available splits have already

be used up on the path from the root to the final node. Then the node's class is the majority of the classes of the variables belong to it.

- A given threshold for the minimum number of observations left in a node is reached. For classification, the threshold of samples in a node is 1 and in regression is 5.

2.5 Variable Importance

2.5.1 Classification

The variable importance indicates the importance of a variable in the classification or regression procedure on all trees. Leo Breiman (Breiman 2001) proposes two different measures for variable importance, the mean decrease in accuracy and the Gini Index (impurity) for the classification procedure. The original random forest variable importance proposed by Breiman (Breiman 2001) is the mean decrease in accuracy on the OOB samples when the predictor's values are randomly permuted. The rationale is the following:

By randomly permuting the values of the predictor variable X_j , then any association with the response variable Y is lost. The permuted predictor variable, together with the non-permuted predictor variables are used to predict the class of the OOB samples. The prediction accuracy is expected to decrease if the permuted predictor variable X_j is associated with the response variable Y. Breiman (Breiman 2001) defines the variable importance as the difference in prediction accuracy before and after permuting X_j , averaged over all trees.

Let $\overline{B}(t)$ be the out-of-bag sample for a tree t with $t \in \{1, \dots, ntree\}$. Then the variable importance of the permuted predictor variable X_j in tree t is:

$$VI^{(t)}(Xj) = \frac{\sum_{i \in \overline{B}(t)} I\left(y_i = \hat{y_i}^{(t)}\right)}{|\overline{B}(t)|} - \frac{\sum_{i \in \overline{B}(t)} I\left(y_i = \hat{y}_{i,\pi_j}{}^{(t)}\right)}{|\overline{B}(t)|},$$

where I is the indicator function, $\hat{y_i}^{(t)}$ is the predicted class for observation i before and $\hat{y}_{i,\pi_j}{}^{(t)}$ is the predicted class for observation i after permuting the values of the predictor variable X_j.

If the permuting predictor variable X_j is not in tree t then:

$$\mathrm{VI}^{(\mathrm{t})}(\mathrm{Xj}) = 0.$$

The overall importance of the permuted variable X_j is then computed as the average importance over all trees:

$$\mathrm{VI}(\mathrm{X_j}) = \frac{\sum_{\mathrm{t=1}}^{\mathrm{ntree}} \mathrm{VI}^{(\mathrm{t})}(\mathrm{Xj})}{\mathrm{ntree}}.$$

From this importance score, a standardized importance score can be computed. If each variable importance $\mathrm{VI}^{(\mathrm{t})}$ has standard deviation (s.d.) σ, then the average importance from *ntree* replications has standard error $\frac{\sigma}{\sqrt{\mathrm{ntree}}}$. The standardized importance score or scaled importance is:

$$\mathrm{z}(\mathrm{X_j}) = \frac{\mathrm{VI}(\mathrm{X_j}) \cdot \sqrt{\mathrm{ntree}}}{\sigma}.$$

However, the results of studies (Diaz-Uriarte 2007, Strobl and Zeileis 2008) , indicate that unscaled importance has better statistical properties than scaled importance.

On the other hand, Gini Index uses the decrease of Gini Index (impurity) after a node split. As we mentioned earlier when a node splits, the split is made by the variable with the smallest Gini Index. A variable is most informative when it has the larger decrease of impurity after a split. The overall Gini importance is calculated by averaging the Gini Index of each variable over all trees in a random forest.

2.5.2 Regression

For regression forests, there are two methods to measure the variable importance. The first is the reduction of mean squared error and the second is the decrease in node impurities. Impurity in regression is measured by the residual sum of squares. Impurity is calculated only at the node at which that variable is used for that split. Breiman (Breiman 2001) suggested the reduction in mean squared error (MSE) when permuting a variable. For tree *t*, the OOB mean squared error is calculated as:

$$\mathrm{OOBMSE_t} = \frac{1}{\mathrm{n_{OOB,t}}} \sum_{\substack{\mathrm{i=1} \\ \mathrm{i} \in \mathrm{OOB_t}}}^{\mathrm{n}} \left(\mathrm{y_i} - \hat{\mathrm{y}}_{\mathrm{i,t}}\right)^2,$$

where $\mathrm{OOB_t} = \{\text{observation i is out of bag in tree t}\}$ and

$n_{OOB,t} = \text{number of OOB observations in tree t.}$

If the variable X_j does not have association with the response variable Y, then after permuting the predictor variable X_j the difference should be small. Then the type for the permuted predictor variable X_j is the following:

$$OOBMSE_t(X_j \text{ permuted}) = \frac{1}{n_{OOB,t}} \sum_{\substack{i=1 \\ i \epsilon OOB_t}}^{n} \left(y_i - \hat{y}_{i,t}(X_j \text{ permuted})\right)^2,$$

should not be larger than $OOBMSE_t$ if the variable X_j isn't associated with the continuous response variable Y.

The MSE reduction over all trees in the forest is:

$$OOBMSE = \frac{1}{n}\sum_{t=1}^{n}(OOBMSE_t) - \left(OOBMSE_t(X_j \text{ permuted})\right),$$

where n=number of trees in the forest.

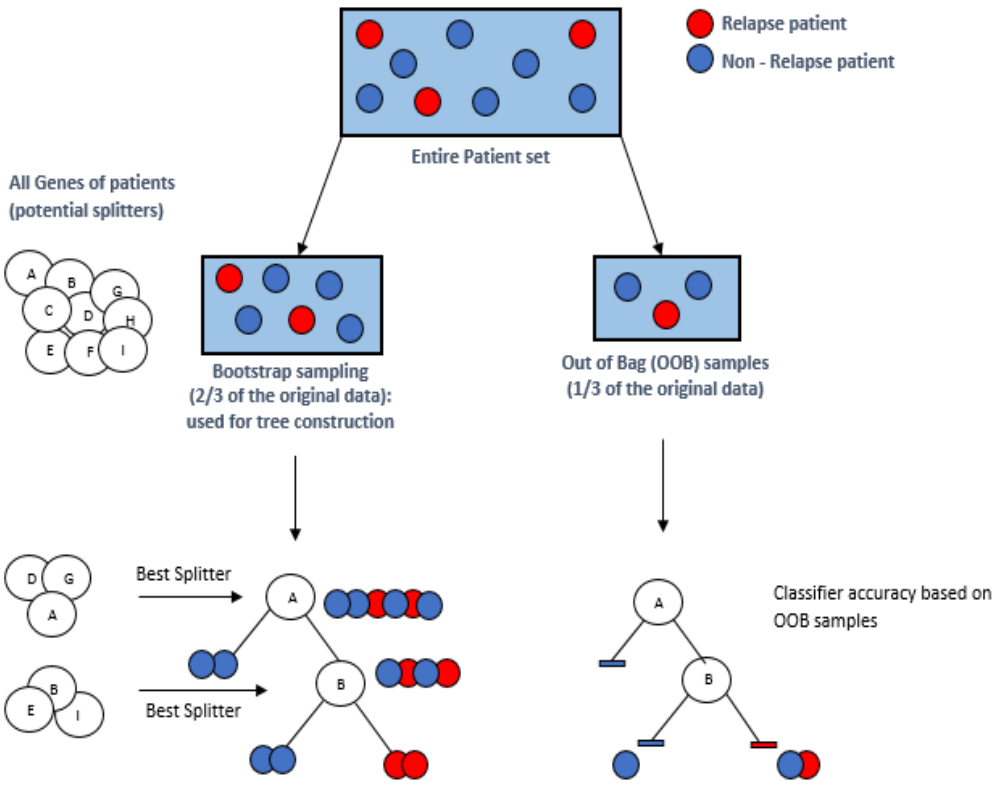


Figure 7: Random forest procedure.

2.6 Tuning critical parameters in random forest

Two critical parameters that the user should appropriately define before running the random forest algorithm are the number of trees and the number of variables randomly chosen as candidates at each split. Leo Breiman (Breiman 2001) suggests that default parametrization of *ntree* and *mtry* values leads to excellent results. The default parameters are for *ntree*=500 and for *mtry*=$\sqrt{\mathrm{M}}$. *Mtry* can take several values with range from 1 to M. In high dimensional data like microarrays, the default parameter choice isn't always good because the algorithm crashes. This happens because the memory, that the random forest algorithm needs to work, exceeds the existing memory capacity. In order to limit the random forest working memory, the user has to reduce the number of trees to be used for the construction of the forest. Zhang and Wang (Chen, Wang et al. 2011) demonstrated that it is not necessary to use the default parameters for satisfying prediction performance. In their study Genuer, Poggi et al. (Genuer, Poggi et al. 2008) suggest that the *mtry* value has to be large enough in order to have high probability to capture important variables in high dimensional problems. Liaw and Wiener (Liaw and Wiener 2002) suggest that if the number of genes is large and the truly informative genes are few, choosing large number of *mtry* gives better performance. Nicoletta Dessì, et al. (Dessì, Milia et al. 2012) in their study come to the following conclusions:

- When a forest with a small number of trees is selected, then a higher value to mtry is chosen, in order to increase the probability of randomly selecting informative variables.
- The random forest method which is applied to the most informative variables has prediction performance better than the random forest in the whole dataset.

Oshiro, et al. (Oshiro, Perez et al. 2012) show in their study that sometimes a large number of *ntree* in random forests only increases its computational cost and has no significant performance gain. After experiments, they conclude that for a number of trees in a range from 64 to 128, it is possible to obtain good balance between performance, processing time and memory usage.

In high dimensional data, we have to reduce the number of variables and take the most informative and important. Moreover, the number of *ntree* and *mtry* depends on the dimensionalities of our dataset. In biological datasets like gene expression, which have thousands of genes, a random forest will use the most of the genes, even

if not all are important. We conclude that with the default values of the parameters the computational cost and the memory usage would be larger. On the other hand, if we set a small number of trees then the computational cost would be less.

We believe a good way to work with high dimensional data specifically microarrays, is to select the most informative or important variables by filtering the entire dataset. Then apply the random forest algorithm without computational difficulties due to memory usage and processing.

2.7 Algorithm Output

One of the most important outputs that random forest for classification provides is the OOB error estimate. This measure is followed by a confusion matrix that records the disagreement between the final model's prediction and the actual outcomes of the observations. The actual observations are the rows of the matrix, and the columns that represent the models predictions for the samples, e.g. if we want to predict the presence of a disease we have the following confusion matrix.

Table 7: Confusion matrix.

N=150	**Predicted**	
Actual	No	Yes
No	86	11
Yes	4	49

In Table 7, there are two possible predicted classes: "yes" and "no". If we were predicting the presence of a disease, "yes" means that patient have the disease, and "no" means that patient don't have the disease. The classifier made 150 predictions. Out of those 150 cases, predicted "yes" 49 times and "no" 86 times. In reality, 53 patients in the sample have the disease and 97 patients do not.

From the confusion matrix in Table 8, we can take important common performance metrics calculated from it.

Table 8: Performance metrics.

N=150	**Predicted**		
Actual	No (PN)	Yes(PY)	**Row totals**
No (AN)	TN=86	FP=11	97
Yes (AY)	FN=4	TP=49	53
Column totals	90	60	N=150

The most commonly rates that are computed by a confusion matrix are the following:

- Accuracy: How often the classifier is correct.

$$acc = \frac{TP + TN}{N}.$$

- Misclassification Rate**:** How often the classifier is wrong.

$$error = \frac{FP + FN}{N}.$$

- True Positive Rate**:** When it's actually yes, how often it predicts yes.

$$tp = \frac{TP}{TP + FN}.$$

- False Positive Rate**:** The actuall class is no, how often the model predicts yes.

$$fp = \frac{FP}{FP + TN}.$$

- Specificity**:** When it's actually no, how often it predicts no.

$$specificity = \frac{TN}{N}.$$

- Precision**:** The model predicts yes, how often is correct.

$$presicion = \frac{TP}{TP + FP}.$$

2.8 Receiver operating characteristics (ROC) curves

A receiver operating characteristics (ROC) graph is a technique which allows researchers to understand how good the performance of their model is. Recently, many researchers apply ROC graphs, to measure the performance of their machine learning algorithms. In this way, they can compare their algorithms based on their performance measure. ROC graphs also can be applied to algorithms which involve decision making between two classes, i.e., a Yes or No on each observation. Some classifiers like neural networks or random forests produce a probability score of each instance which represents the probability that each instance will be classified in a class. A classification model is always followed by a confusion matrix. The confusion matrix of a model is a mapping of instances between the actual and predicted classes of them.

In Table 8 we have a confusion matrix with the predicted and actual outcomes of a model. In Table 8, we have four possible outcomes. A true negative is the instance which actually belongs in class no and the model predicts that it belongs to class no. A false positive is the instance which actually belongs in class no and the model predicts that it belongs to class yes. A false negative is the instance which actually belongs in class yes and the model predicts that it belongs to class no then this is a false negative. A true positive is the instance which actually belongs in class yes and the model predicts that it belongs to class yes this is a true positive. From this confusion matrix, we can take several performance metrics as false positive and true positive rate. ROC graphs are two-dimensional graphs which have in y axis the true positive rates and in x axis the false positive rates.

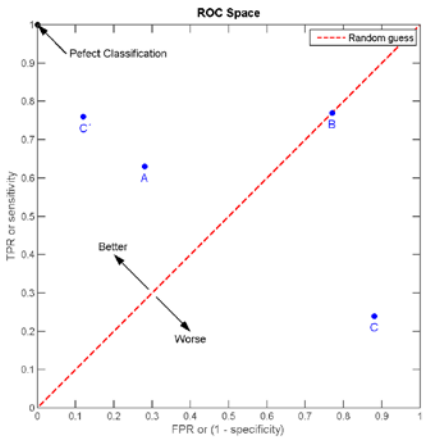


Figure 8: A ROC graph showing four discrete classifiers.

In Figure 8 the point (0, 0) means that a classifier commits no false positive errors but also no true positive gains. On the other hand, the point (1, 1) is exactly the opposite of the point (0, 0) in which we have positive classifications. The point (0, 1) which is shown by the arrow in the figure 8 represents perfect classification, and the point (1, 0) represents a flawed classification. Classifiers which belong to the upper triangle separated by the dashed line are better than the others. Also the diagonal line y=x represents the random guessing. The classifier B (0.78, 0.78) represents the strategy in which the classifier guesses the positive class 78% of the time but also make false negative errors 78% of the time. Classifier C is the worst of the four classifiers because it makes more false positives than true positives. Therefore, classifier C is worse than random guessing, in this case if we want to predict the class of an instance, it is better to flip a coin than to rely on classifier C.

In our study, we work with the random forest algorithm. The random forest provides a matrix which gives the number of OOB votes of each sample. Hence, we have for each sample the probability to be classified in each of the two classes. This ranking classifier is used to produce the thresholds. In the beginning, we start with threshold$+\infty$. In *ROCR* package in R, each threshold consists of the predicted scores, from the vote's matrix, produced by random forest algorithm. If the samples above the threshold are classified in the correct class, these are true positives. On the other hand, if a sample above the threshold is misclassified in respect with actual class it belongs, then this is a false positive. Each threshold produces a point in the ROC graph. After the ROC graph calculation, we find the area under the curve (AUC) which represents our model performance. The AUC is a numerical summary that represents the probability that the classifier will classify correctly a sample to the actual class that it belongs. Classifiers that have higher AUC scores represent perfect classifiers and perfect classifiers give models with better prediction accuracy. Models with AUC scores approximately to 70% are considered average classifiers and AUC scores approximately to 80% are considered good classifiers. If the AUC score of the model is bigger than 90%, then this model is a perfect classifier.

2.9 Advantages

In summary, a random forest is a good choice of model building for a number of reasons:

1. Very little pre-processing of the data needs to be performed.
2. Same idea for regression and classification except of the split criterion.
3. Handle categorical and continuous predictors.
4. Quick to fit even for large problems.
5. Automatic variable selection.
6. Many trees are built using two levels of randomness, the one level is from the bagging procedure and the other from variable selection. In this way each tree is an independent model and the resulting model tends not to overfit to the training dataset.
7. It produces very accurate classifiers and learning fast.
8. It offers an experimental method for detecting variable interactions.
9. No need of pruning in trees.

3. Variable Selection using Random Forests (varSelRF)

3.1 Introduction

In machine learning and statistics, variable selection is the process of selecting a subset of relevant features (variables/predictors) to include in the model construction. Variable selection is used to reduce the number of redundant variables. In microarrays, the number of variables (genes) is much larger than the number of samples (patients), in this case, we want to keep only the most important variables which have the capacity of separating classes of patients. We want to keep a small number of variables because from all only few are correlated with the pathogenesis of a disease. For this reason, it's difficult for several classification methods to classify and find the most important genes.

3.2 Algorithm Description

In our study, we have a microarray gene expression dataset with dimensions 40x463442. Our goal is to find a small subset of CpG sites, that classifies the patients correctly to their pre-defined classes, and achieve good predictive

performance as well. Due to the high dimensionality of our dataset (40x463442), it was very difficult to classify correctly our patients to their pre-defined classes applying the random forest algorithm to the entire dataset. Before the selection of this method, we applied several statistical methods to our dataset which we will explain in chapter 4, in order to find a small number of CpG sites with good predictive accuracy. However, none of these methods could find a small number of CpG sites with good predictive accuracy. Therefore, after a lot of research we conclude that in order to have a good prediction model we have to reduce our CpG sites, and keep only the relevant CpG sites according to the random forest classification procedure. We used the method proposed by Díaz-Uriarte et al. (Diaz-Uriarte 2007). We applied this method, because it is suitable for microarray datasets and keeps only the relevant CpG sites according to the random forest classification procedure. In their research, they investigate the use of random forest for classification of microarray data and propose a new method of gene selection, which is based on the variable importance of random forest and the OOB error. Variable importance and OOB error rate are two important measures of random forests.

Díaz-Uriarte et al. (Diaz-Uriarte 2007), describe a backward elimination procedure using random forests for selecting genes from microarray data. The steps of this method are the following:

- Fit a random forest model to all data and rank in decreasing order all genes based on their variable importance.
- Iteratively fit random forests and in each step remove 20% of genes which have the lowest variable importance.
- Select the group of genes corresponding to the random forest reaching the smallest OOB error rate.
- Estimate the prediction error rate based on the fact that the OOB error rate is biased down due to the recursive variable elimination and it can't be used to assess the overall error rate of the approach.

In their study, in order to calculate the prediction error rate they applied the ".632+" bootstrap method to the complete procedure (the remaining variables who achieve the lowest OOB error rate, after the variable selection procedure). The ".632+" bootstrap method was proposed by Efron and Tibshirani (Efron and Tibshirani 1997). The ".632+" method adds weights both to resubstitution error and to the leave-one-out bootstrap error. The resubstitution error represents the classifier's (random forest) error when applied to training data. On the other hand, the leave-

one-out bootstrap method represents the error on samples (patients) that are not used to train the classifier or in the variable selection procedure. To give a better understanding, if we want to predict Y with X using our prediction model *f* (in our case is the random forest model) then the resubstitution error would be:

$$\overline{\mathrm{err}} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{L}\left(\mathrm{y_i}, \hat{\mathrm{f}}(\mathrm{x_i})\right),$$

where $\mathrm{L}\left(\mathrm{x}, \hat{\mathrm{f}}(\mathrm{y})\right) = \left(\mathrm{y} - \hat{\mathrm{f}}(x)\right)^2$ is a function such as squared error and N is the number of samples. In addition x is a vector of input data (CpG sites) of each sample and y is the actual class of each sample. The resubstitution error isn't a good estimator, because it uses the same data to construct and assess a prediction rule. The resubstitution error decreases when the model complexity is increased. When the model complexity is increased enough then $\overline{\mathrm{err}} = 0$.

For the leave-one-out bootstrap method suppose that the training data is $\mathrm{Z} = (\mathrm{z_1}, \mathrm{z_2}, \ldots, \mathrm{z_N})$ where $\mathrm{z_i} = (\mathrm{x_i}, \mathrm{y_i})$. From this set we can take B bootstrap samples with replacement $\mathrm{M_1}, \mathrm{M_2}, \ldots, \mathrm{M_B}$, where each $\mathrm{M_i}$ has the same size N. Hence, the leave-one-out bootstrap estimator is given by the following equation:

$$\mathrm{Err_{boot}} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{1}{|\mathrm{K^i}|}\sum_{\mathrm{b}\in\mathrm{K^i}} \mathrm{L}\left(\mathrm{y_i}, \hat{\mathrm{f}}_{\mathrm{b}}(\mathrm{x_i})\right),$$

where $\hat{\mathrm{f}}_{\mathrm{b}}(\mathrm{x_i})$ is the predicted value from the random forest model corresponding to the b-th bootstrap sample (b=1, …, $|\mathrm{K^i}|$). $\mathrm{K^i}$ is the set of indices of bootstrap samples that don't contain observation i, and $\left|\mathrm{K^i}\right|$ is the number of such bootstrap samples. The resubsitution error tends to underestimate the prediction error. On the other hand, the leave-one-out bootstrap error tends to overestimate the prediction error. For this reason, in their study Efron and Tibshirani (Efron and Tibshirani 1997) proposed the ".632" method, which adds weights both to resubstitution error and to the leave-one-out bootstrap error based on the following equation:

$$\mathrm{Err}_{.632} = 0.368\overline{\mathrm{err}} + 0.632\mathrm{Err_{boot}}.$$

Coefficient 0.632 is multiplied with $\mathrm{Err_{boot}}$ based on the argument that each bootstrap sample contains approximately 2/3≅0.632 of the complete sample set. Hence, the resubstitution error will be multiplied with 1-0.632≅0.368. We note that

this method assigns more weight on the leave-one-out bootstrap error estimation than on the resubstitution error estimation.

The $\mathrm{Err}_{.632}$ performs better than other competitors, but in highly overfit rules where $\overline{\mathrm{err}} = 0$, the $\mathrm{Err}_{.632}$ tends to be downwardly biased. In order to improve the $\mathrm{Err}_{.632}$ estimator $\hat{\gamma}$ is defined to be the no-information error rate as:

$$\widehat{\gamma} = \frac{1}{\mathrm{N}^2}\sum_{\mathrm{i}=1}^{\mathrm{N}}\sum_{\mathrm{i}'=1}^{\mathrm{N}} \mathrm{L}\left(y_i, \hat{\mathrm{f}}(x_{i'})\right).$$

The estimator $\hat{\gamma}$ is the error rate of our prediction rule if the inputs and class labels were independent. The estimator $\hat{\gamma}$ is obtained by evaluating the prediction rule on all possible combinations of targets $\mathrm{y_i}$ and predictors $\mathrm{x_{i'}}$. The relative overfitting rate is defined then as:

$$\widehat{\mathrm{R}} = \frac{\mathrm{Err}_{\mathrm{boot}} - \overline{\mathrm{err}}}{\widehat{\gamma} - \overline{\mathrm{err}}}.$$

This quantity ranges from 0 ($\mathrm{Err}_{\mathrm{boot}} = \overline{\mathrm{err}}$) to 1 ($\mathrm{Err}_{\mathrm{boot}} = \widehat{\gamma}$). Finally the ".632+" estimator is defined as:

$$\widehat{\mathrm{Err}}^{(.632+)} = (1 - \mathrm{w}) \cdot \overline{\mathrm{err}} + \mathrm{w} \cdot \mathrm{Err}_{\mathrm{boot}}, \qquad (3)$$

$$\text{with } \mathrm{w} = \frac{.632}{1 - .368\widehat{\mathrm{R}}}.$$

The weight w ranges from .632 if $\widehat{\mathrm{R}} = 0$ to 1 if $\widehat{\mathrm{R}} = 1$. Therefore, $\widehat{\mathrm{Err}}^{(.632+)}$ ranges from $\mathrm{Err}_{.632}$ to $\mathrm{Err}_{\mathrm{boot}}$. We can also express equation (3) as:

$$\widehat{\mathrm{Err}}^{(.632+)} = \mathrm{Err}_{.632} + (\mathrm{Err}_{\mathrm{boot}} - \overline{\mathrm{err}})\frac{.368 \cdot .632 \cdot \widehat{\mathrm{R}}}{1.632 \cdot \widehat{\mathrm{R}}}, \qquad (4)$$

which emphasizes that $\widehat{\mathrm{Err}}^{(.632+)}$ exceeds $\mathrm{Err}_{.632}$ by an amount depending on $\widehat{\mathrm{R}}$. In cases where $\widehat{\gamma} \leq \overline{\mathrm{err}}$ or $\overline{\mathrm{err}} < \widehat{\gamma} \leq \mathrm{Err}_{\mathrm{boot}}$ the relative overfitting rate fall outside of bounds $[0,1]$. In order to deal with this problem, we need to modify $\mathrm{Err}_{\mathrm{boot}}$ and $\widehat{\mathrm{R}}$:

$$\mathrm{Err}_{\mathrm{boot}}{}' = \min(\mathrm{Err}_{\mathrm{boot}}, \widehat{\gamma})$$

and

$$\widehat{\mathrm{R}}' = \begin{cases} \dfrac{(\mathrm{Err}_{\mathrm{boot}} - \overline{\mathrm{err}})}{(\widehat{\gamma} - \overline{\mathrm{err}})}, & \text{if } \mathrm{Err}_{\mathrm{boot}}, \widehat{\gamma} > \overline{\mathrm{err}}, \\ 0, & \text{otherwise.} \end{cases}$$

After these modifications the equation (4) yields the following equation:

$$\widehat{\mathrm{Err}}^{(.632+)} = \mathrm{Err}_{.632} + (\mathrm{Err}_{\mathrm{boot}}{}' - \overline{\mathrm{err}})\frac{.368 \cdot .632 \cdot \widehat{\mathrm{R}}'}{1 - .368 \cdot \widehat{\mathrm{R}}'}.$$

To calculate the ".632+" bootstrap method, Diaz-Uriarte et al. (Diaz-Uriarte 2007) applied this method to the complete procedure. The leave-one-out bootstrap error is calculated by using the samples that are not used in the random forest construction or in the variable selection procedure. According to Jiang and Simon (Jiang and Simon 2007) for microarray datasets with n<p, where n= number of samples (patients) and p=number of variables (CpG sites), the overifitting problem exists and the resubstitution error is often close to zero due to the complexity of the model. When the resubtitution error is zero and w=.632 then $\widehat{\mathrm{Err}}^{(.632+)} = .632\mathrm{Err}_{\mathrm{boot}}$ is systematically downwardly biased where there are no class differences (Breiman, Friedman et al. 1984, Efron and Tibshirani 1997). In this case, the ".632+" method puts too much weight on the leave-one-out bootstrap error, in order to deal with this problem. In the case where n>p, the ".632+" bootstrap method often performs well in classification problems and is very popular for having low variability and only moderate bias.

There are two different types of variable importance, the importance based on the mean decrease of classification accuracy and the Gini Index. In their study Diaz-Uriarte et al. (Diaz-Uriarte 2007) used the importance based on the mean decrease of classification accuracy. Moreover, they applied this method in nine microarray datasets, and in each dataset they find small sets of genes with high predictive accuracy. In addition, they proved that this method's results are equally efficient with other existing methods of gene selection. This method can only be applied for supervised random forest classification problems and not for regression random forests and unsupervised problems. In Figure 9 we present the variable selection procedure explained before.

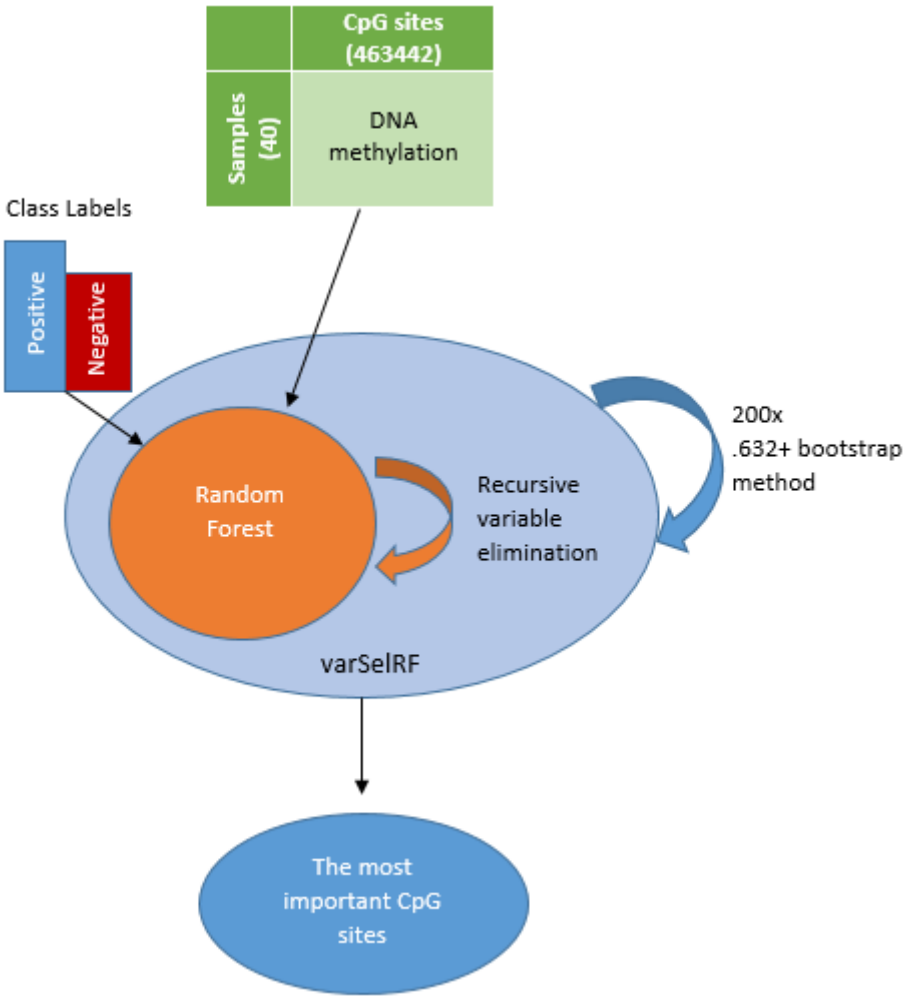


Figure 9: Variable selection procedure.

4. Real-data Application

4.1 Methodology and data description

In our study, we have a set of 40 patients with chronic lymphocytic leukemia. Each patient has two states, one state before the relapse and the other after the relapse. 37/40 patients were treated with the same treatment (FCR). Each patient has 463442 CpG sites with range of their values from 0-1. Hence, we have two matrices with dimensions 40x463442. In our main research, we have chosen to work with the first state, in which each patient is checked before the relapse. We worked with this state because we wanted to know if the patients responded to the treatment. If they didn't, i.e. the time to relapse is short, then, it might have been better to consider they

receive another treatment. Moreover, we have a file with the clinical-biological data of each patient, which contains the actual classes of patients we want to classify.

The aim of our study is to detect the most important CpG sites, which have the ability to separate the patients to their pre-defined classes based on their clinical-biological data before the relapse.

We started by “cleaning” the data and synchronize the order of patients from the annotation file with the matrix of dimensions 40x463442. Our first attempts were to detect the most important CpG sites with standard statistical techniques. We applied the Wilcoxon rank sum test to our data. The Wilcoxon rank sum test is appropriate because we have samples from two different states, and our data didn’t follow the normal distribution. The null hypothesis of Wilcoxon rank sum test was H_0: two samples come from the same population against an alternative hypothesis, especially that a particular population tends to have larger values than the other.

After the evaluation of the Wilcoxon rank sum test we kept only the CpG sites with p-value<0.05. Continuing, we evaluated the false discovery method due to the multiple comparisons that the Wilcoxon rank sum test made. Specifically, we used the method proposed by Benjamini-Hocheberg (Benjamini and Hochberg 1995) with threshold 0.1 because it is less conservative and more appropriate than the method of Benjamini-Yekutieli (Benjamini and Yekutieli 2001). In addition, the evaluation of Benjamini-Yekutieli method yields in zero CpG sites with the same threshold and seems to be more restrictive and conservative.

The evaluation of the false discovery method yields again to approximately 100 remaining CpG sites. In addition these CpG sites didn’t classify correctly our patients to their pre-defined classes. The number of CpG sites should be small enough because, then, CpG sites can be easier checked by biologists in order to be used as a prognostic signature. In another attempt, we applied the random forest algorithm to the entire dataset, but the algorithm crashed because of the high dimensionality of our data (40x463442). Although our work take place in INAB and in a server with large memory capacity, the random forest algorithm requirements exists the memory capacity of the server.

All the above methods didn’t work as we expected. In order to reduce the number of CpG sites, we computed the s.d. of the entire dataset for each CpG and we kept only CpG sites with s.d.≥0.3. The threshold 0.3 is an empirical cutoff. We calculated the s.d. at each CpG site, to choose those where the patients exhibited

have high variance. This possibly would reflect their dissimilarity. The number of the remaining CpG sites was 6721. Therefore, in these CpG sites, we applied the method of Díaz-Uriarte et al. (Diaz-Uriarte 2007), which can find the most important CpG sites based on the random forest algorithm. The variable selection method was necessary because we wanted a small set of CpG sites which achieved good predictive performance. In addition, the variable selection method detected the most relevant CpG sites to the classification procedure and eliminated the irrelevant CpG sites. Moreover, we applied the random forest algorithm to the matrix of dimensions 40x6721 which follows as a result of the s.d. filtering but the results weren't satisfying. Therefore, probably the random forest algorithm should be applied to a smaller set of CpG sites, in order to be more accurate.

The variable selection algorithm of Díaz-Uriarte et al. (Diaz-Uriarte 2007) was suitable for our problem and, in addition, in their study they applied the algorithm in nine microarray data sets. In our study, we examined the supervised learning method of random forests. Initially, we separated the patients in two groups based on their clinical-biological data. In the first grouping, patients separated based on their differential methylated CpG sites in two classes "High" and "Low". In the class "Low" belonged the patients who had few changes on their CpG sites comparing the two states of them. On the other hand, in class "High" belonged the patients who had big changes on their CpG sites comparing the two states of them. We have to mention, that small number of differential methylated CpG sites is related to short time to relapse. The second grouping was separated based on the months each patient relapsed. For this group, we had two classes of patients where the first class was for patients who relapsed in less than 24 months ("Ultra High risk") and the other class was for patients who relapsed in more than 24 months ("Others"). It is important to note that the period less than 24 months was followed by aggressive disease after recurrence.

In the first grouping, we had 40 patients. In the first class "High" belonged 23 from 40 patients and in the second class "Low" belonged 17 patients. In the matrix of dimensions 40x6721 we applied the variable selection algorithm. The algorithm starts by creating a large random forest, with a number of trees 5000 and the number of predictors tried at each split is the square root of 6721. After the random forest evaluation, the algorithm sorts by decreasing order the CpG sites based on their variable importance. Next, it removes 20% of the CpG sites with the lowest variable importance. The variable importance is computed based on the mean decrease of accuracy and not the Gini Index. Iteratively the variable selection algorithm fits

random forests, but, now, with a number of trees equal to 2000 and the same number of predictors as before is evaluated at each split. The number of trees is selected to be large enough because we want to increase the stability of our results. When the variable selection method finishes, we can see the most important CpG sites of the complete procedure which achieve at best to classify the patients correctly to their classes and the confusion matrix which shows how the model classifies the patients to their classes.

In Table 9, the first random forest is presented, which is applied to 6721 CpG sites with number of trees 5000 and number of predictors 81 at each split. The above random forest classification procedure wasn't satisfying. We observed that the OOB error rate is 50%. As we mentioned earlier the confusion's matrix columns represent the model predictions and the rows the actual class of each patient.

Table 9: First random forest confusion matrix for classes "High" and "Low".

Type of random forest: classification			
Number of trees: 5000			
Number of variables tried at each split: 81			
OOB estimate of error rate: 50%			
Confusion matrix:			
Predicted / Actual	High	Low	Classification error
High	19	4	0.173
Low	16	1	0.941

In Table 9, the model classified correctly 19/23 patients in class "High" and wrongly 4/23 in class "Low" with classification error 17.3%. The model classified correctly 1/17 patient in class "Low" and it classified wrongly 16/17 patients in class "High" with classification error 94.1%. Only one patient of class "Low" classified correctly based on his actual class. For class "High" the model predicted that 4 patients belonged to class "Low" but they actually belonged in class "High". For class "Low" the model predicted that 16 patients belonged to class "High" but they actually belonged in class "Low". This model before the variable selection wasn't satisfying. In

the following matrix in Table 10 we present the final random forest which was constructed by the remaining variables from the variable selection procedure.

These remaining CpG sites had the biggest variable importance and the ability to classify the patients correctly to their classes. In the end of the variable selection algorithm the remaining and most important CpG sites were 11. This was very important because we found a small set of CpG sites which achieved very good predictive accuracy. As we mentioned earlier the confusion's matrix columns represents the model predictions and the rows the actual class of each patient. In Table 10 the model classified correctly 23/23 patients in class "High" with classification error 0%. Each patient of class "High" classified correctly based on his actual class. The model classified correctly 16/17 patients in class "Low" and it classified wrongly one patient in class "High" with classification error 5.8%. This means that the model classified the patient in class "High" but actually this patient belonged in class "Low".

Table 10: Final random forest confusion matrix for classes "High" and "Low".

Type of random forest: classification			
Number of trees: 2000			
Number of variables tried at each split: 3			
OOB estimate of error rate: 2.5%			
Confusion matrix:			
Predicted / Actual	High	Low	Classification error
High	23	0	0.000
Low	1	16	0.058

It is obvious that after the variable selection the random forest improved for 11 CpG sites than for 6721 CpG sites. Also these 11 CpG sites seem to achieve good predictive accuracy instead of 6721.

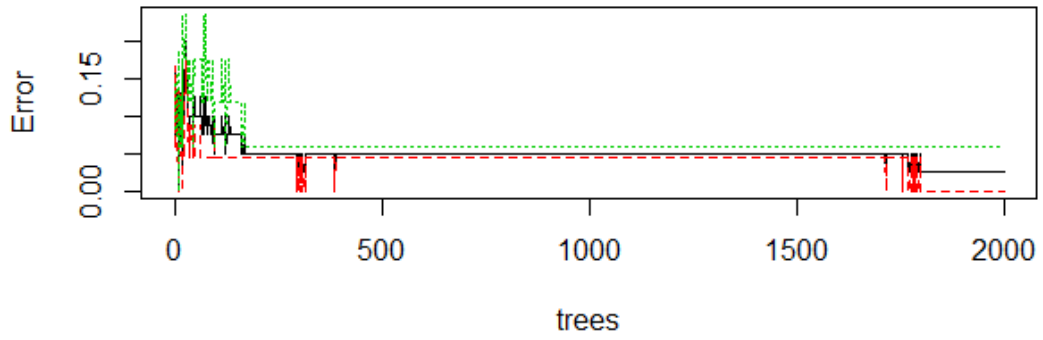


Figure 10: The OOB error rate is displayed for classes "High" and "Low" of 11 CpG sites for different number of trees.

In Figure 10 we see the error of each class and the OOB error of this procedure for different number of trees. The red dashed line represents the error rate of class “High” in each tree, green dashed line represents the error rate of class “Low” in each tree and the black line represents the OOB error rate in each tree. We note, that as the number of trees growing the OOB error is reduced. . Note that the OOB error remained unchanged, approximately between 150 and 1750 trees.

In their study, Díaz-Uriarte et al. (Diaz-Uriarte 2007) note that the OOB error rate is biased down due to the backward elimination. In order to handle the bias problem and find the prediction error rate, they applied the “.632+” bootstrap method of Efron and Tibshirani (Efron and Tibshirani 1997) to the complete procedure (the remaining variables who achieve the lowest OOB error rate, after the variable selection procedure). In addition, they note that “*the error rate of the variable selection procedure, estimated using the .632+ bootstrap method, indicates that the variable selection procedure does not lead to overfitting, and can achieve the objective of aggressively reducing the set of selected genes*”.

Hence, after the evaluation of the variable selection procedure we applied the “.632+” bootstrap method to the remaining 6 CpG sites, in order to understand the specificity of our predictors (CpG sites). Specifically, we applied this method five different times for 200 bootstrap samples and took the average prediction error of them in order to compare our results with the results of Díaz-Uriarte et al.(Diaz-Uriarte 2007). In Table 11, we present the prediction error rate values for each different run.

Table 11: The prediction error rate is displayed, which was calculated with the ".632+" bootstrap method of classes "High" and "Low" of 11 CpG sites.

Number of runs	Prediction error rate
1	0.1436
2	0.1415
3	0.1232
4	0.1250
5	0.1346

The average prediction error rate is 0.1335. For example in the first run, the resubstitution error was equal to zero, the leave-one-out bootstrap was equal to 0.1940, and the weight was equal to 0.7402. We note that the ".632+" method puts more weight in order to handle the bias problem. The OOB error rate can be viewed as a non-smooth estimator compared to the leave-one-out bootstrap error. This may happen because the OOB error rate employs a majority vote on all trees for sample i, while the leave-one-out bootstrap method takes an average on errors of these predictions. Hence, due to the variable elimination procedure, we expected that the OOB error rate would be biased down and every time would be less than the leave-one-out bootstrap estimator.

Based on bibliography, our results are reasonably good. Moreover, we used these 11 CpG sites to evaluate two other well-known methods in order to see how they respond. The first method is called heatmap and the second principal component analysis. We evaluated these methods for the set of 6721 CpG sites and for the set of 11 CpG sites resulted by the random forest in order to see their differences before and after the selection procedure.

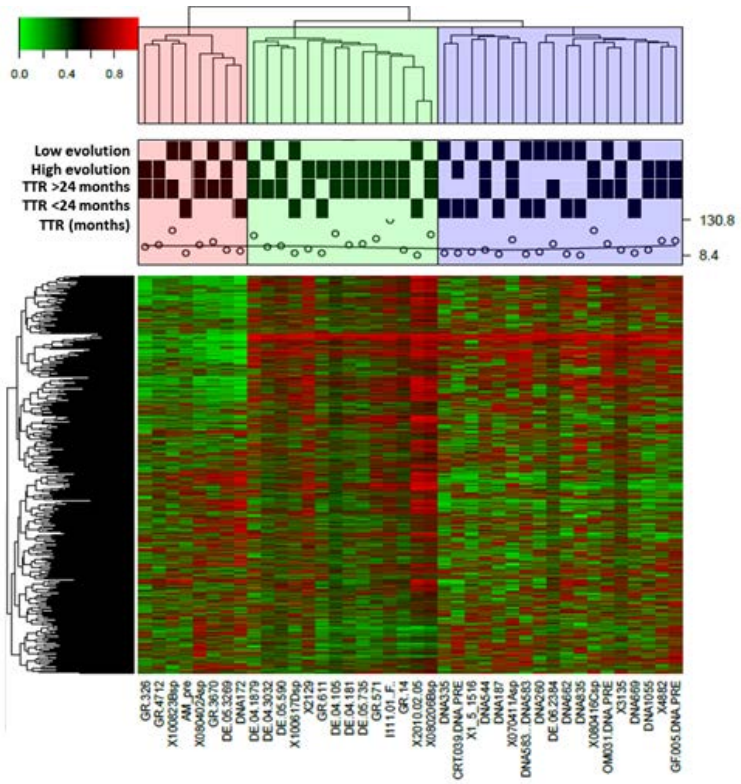


Figure 11: Heatmap of 6721 CpG sites.

In Figure 11 is a heatmap of 6721 CpG sites, in which was applied hierarchical clustering both in patients (columns) and in CpG sites (rows). The dendrogram shows the clusters of the patients and the CpG sites that have been made after the hierarchical clustering. The elements under the top dendrogram are the clinical-biological data for each patient taken from another file. The green color in the above figure depicts that the DNA methylation value of a CpG site in a patient is less than 0.3 and this CpG site with such value is hypomethylated. On the other hand, red color depicts that the DNA methylation value of a CpG site is hypermethylated in a patient and its value is bigger than 0.7. Finally, the black color range from (0.3, 0.7) represents a DNA methylation value of a CpG site which is not hyper or hypo methylated in a patient.

After the hierarchical clustering of patients and the separation of them in clusters, we observed that they couldn't manage to group based on their clinical-biological data and the classes that we wanted.

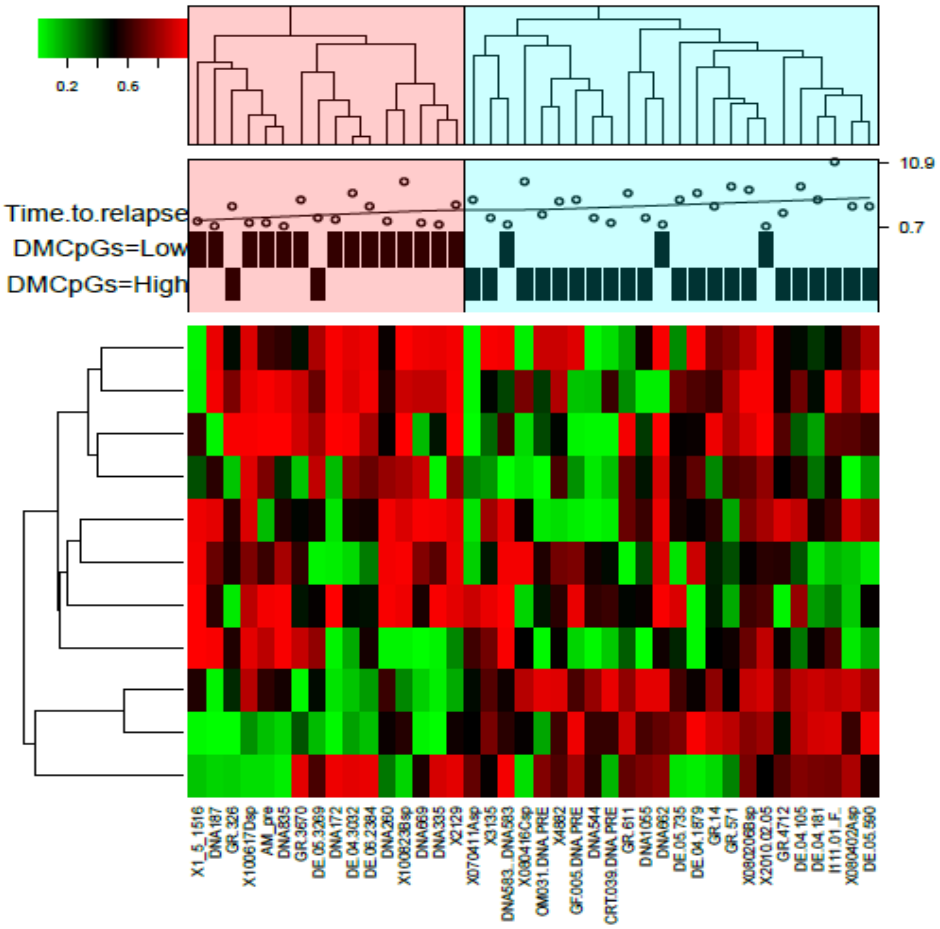


Figure 12: Heatmap of 11 CpG sites.

The above heatmap in figure 12 evaluated based on the selected 11 CpG sites, which have been taken after the variable selection procedure. In this heatmap we notice that the patients who belonged in the same cluster had almost the same clinical-biological data. In the first cluster of patients (left), only 2 of 16 belonged to the other group. In the second cluster of patients (right), only 3 of 24 belonged to the other group. Also we observed that between these two clusters of patients revealed a slightly change in the volatility of patients DNA methylation values.

Hence, we concluded that after the variable selection method these 11 CpG sites yield to a better grouping of patients than the grouping made with 6721 CpG sites. Also we detected changes between DNA methylation values of CpG sites in two clusters.

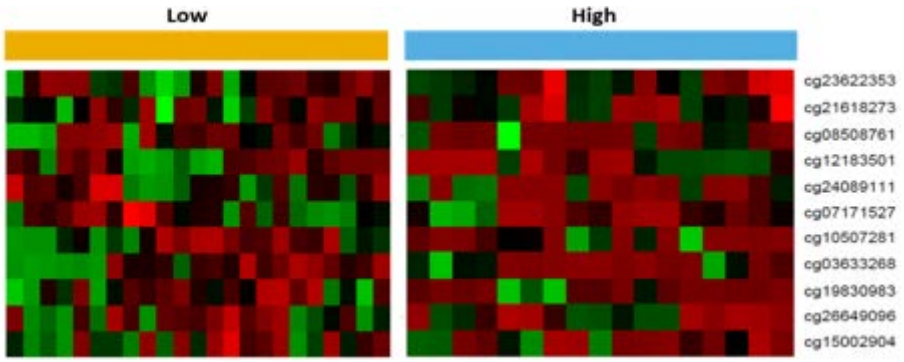


Figure 13: Classes “High”, “Low” of 11 CpG sites.

In order to give a better understanding according to the volatility of the DNA methylation values of patients we separated the two classes. We note, that in class “Low” we had more green cells than in class “High” in Figure 13.

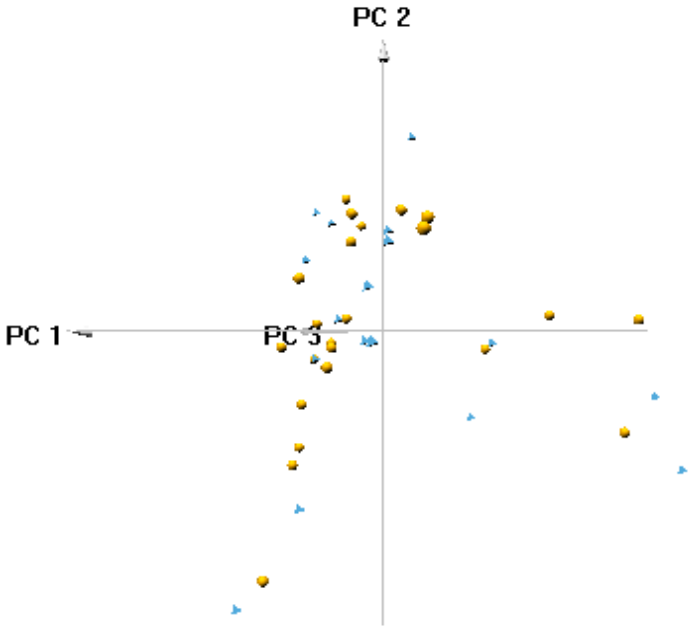


Figure 14: Principal component analysis of 6721 CpG sites of classes “High” and “Low”.

The above principal component analysis in Figure 14 was based on 6721 CpG sites. Each point represents a patient. Yellow points represent the patients who belonged in class “Low” and light blue points represent the patients who belonged in class “High”. We observed that the patients couldn’t manage to group based on their clinical-biological data and we can’t see a clear separation between them.

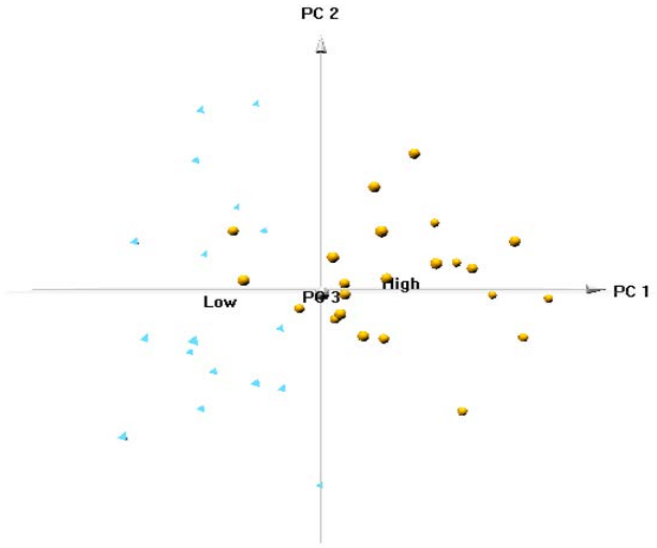


Figure 15: Principal component analysis of 11 CpG sites of classes “High” and “Low”.

The principal component analysis of Figure 15 was evaluated was based on 11 CpG sites taken from the variable selection procedure which was applied before. The difference between these two plots of figures 14 and 15 was very large. In figure 15, patients tended to group together and only 3 patients of class “High” were in the wrong side.

Therefore, we concluded that the heatmap and the principal component analysis methods achieved better performance of grouping patients to their classes for these 11 CpG sites than for the 6721 CpG sites.

Finally, in order to check the performance of our classifier, we evaluated ROC analysis which shows us the performance of our classifier in binary problems.

Figure 16: ROC plot of 11 CpG sites.

In Figure 16 we see the performance of our classifier. We note, that the area under curve (AUC) was equal to 0.99744. This means that our classifier was perfect. Values bigger than 0.9 means that the classifier's performance is perfect. True positive and false positive rates calculated from the table's votes produced by the random forest algorithm.

A table of votes contains in rows patients and in columns their classes. For each patient was given the probability to be classified in the one or in the other class by the random forest model. These probabilities for each class and patient were calculated by the number of (OOB) votes from the random forest. This method was evaluated in R and the package was ROCR. In Table 12 is given the matrix of votes produced by the random forest algorithm, the true class of each patient, the false positive rate (FPR), and the true positive rate (TPR).

Table 12: Votes of classes "High" and "Low" from the last random forest procedure.

Patients	Probability "High"	Probability "Low"	True class	FPR	TPR
				0.00000000	0.00000000
1	0.03984064	0.96015936	"Low"	0.00000000	0.05882353
2	0.11961057	0.88038943	"Low"	0.00000000	0.11764706
3	0.12275862	0.87724138	"Low"	0.00000000	0.17647059
4	0.19350474	0.80649526	"Low"	0.00000000	0.23529412
5	0.19464034	0.80535966	"Low"	0.00000000	0.29411765
6	0.20726783	0.79273217	"Low"	0.00000000	0.35294118
7	0.20728291	0.79271709	"Low"	0.00000000	0.41176471
8	0.20827586	0.79172414	"Low"	0.00000000	0.47058824
9	0.21014493	0.78985507	"Low"	0.00000000	0.52941176
10	0.23915900	0.76084100	"Low"	0.00000000	0.58823529
11	0.26266667	0.73733333	"Low"	0.00000000	0.64705882
12	0.28922237	0.71077763	"Low"	0.00000000	0.70588235
13	0.31504923	0.68495077	"Low"	0.00000000	0.76470588
14	0.33736559	0.66263441	"Low"	0.00000000	0.82352941
15	0.39337176	0.60662824	"Low"	0.00000000	0.88235294
16	0.44157609	0.55842391	"Low"	0.00000000	0.94117647
17	0.50210379	0.49789621	"High"	0.04347826	0.94117647

18	0.50464807	0.49535193	“Low”	0.04347826	1.00000000
19	0.61696306	0.38303694	“High”	0.08695652	1.00000000
20	0.63966480	0.36033520	“High”	0.13043478	1.00000000
21	0.68731988	0.31268012	“High”	0.17391304	1.00000000
22	0.70637119	0.29362881	“High”	0.21739130	1.00000000
23	0.71303075	0.28696925	“High”	0.26086957	1.00000000
24	0.71832884	0.28167116	“High”	0.30434783	1.00000000
25	0.73719677	0.26280323	“High”	0.34782609	1.00000000
26	0.78160920	0.21839080	“High”	0.39130435	1.00000000
27	0.80195258	0.19804742	“High”	0.43478261	1.00000000
28	0.84232365	0.15767635	“High”	0.47826087	1.00000000
29	0.84366577	0.15633423	“High”	0.52173913	1.00000000
30	0.85694823	0.14305177	“High”	0.56521739	1.00000000
31	0.87719298	0.12280702	“High”	0.60869565	1.00000000
32	0.88504155	0.11495845	“High”	0.65217391	1.00000000
33	0.88808140	0.11191860	“High”	0.69565217	1.00000000
34	0.93085106	0.06914894	“High”	0.73913043	1.00000000
35	0.94636015	0.05363985	“High”	0.78260870	1.00000000
36	0.95714286	0.04285714	“High”	0.82608696	1.00000000
37	0.97503285	0.02496715	“High”	0.86956522	1.00000000
38	0.98412698	0.01587302	“High”	0.91304348	1.00000000
39	0.99301676	0.00698324	“High”	0.95652174	1.00000000
40	0.99469496	0.00530504	“High”	1.00000000	1.00000000

We note that we started by sorting the column “Low” in decreasing order, and a list of cutoffs defined by the numbers of column “Low”. Each time for a given cutoff we observed if the above patient’s true classes were the same with the classes that the random forest model predicted. We observed that all patients with true classes “Low”, achieved higher probabilities to be classified in the class “Low” than in the class “High” by the random forest model except from the patient 18. Specifically, the probability that the patient 18 would be classified to the class “High” was only marginally higher than the probability of the class “Low” but the actual class of this

patient was “Low”. In this case, the performance of our model decreased, because the model’s prediction in the case of patient 18 wasn’t the same with the true class of this patient and the FPR was calculated as FPR=1/23.

Now we will evaluate the same method but for classes “Ultra High risk” and “Others”. In the second group we had the same 40 patients. In the first class “Ultra High risk” belonged 26 from 40 patients and in the second class “Others” belonged 14 patients. In the matrix of dimensions 40x6721 we applied the variable selection algorithm with the same parameters as before.

In Table 13 is presented the first random forest which applied to 6721 CpG sites with number of trees 5000 and number of predictors 81. We observed that the OOB error rate is 32.5%. In Table 13 the model classified correctly 26/26 patients in class “Ultra High risk” with classification error 0%. The model classified correctly 1/14 patient in class “Others” and it classified wrongly 16/17 patients in class “Ultra High risk” with classification error 92.8%. Only one patient of class “Others” classified correctly based on his actual class. The model predicted correctly that all patients belonged to class “Ultra High risk”.

Table 13: First random forest confusion matrix for classes “Ultra High risk” and “Others”.

Type of random forest: classification			
Number of trees: 5000			
Number of variables tried at each split: 81			
OOB estimate of error rate: 32.5%			
Confusion matrix:			
Predicted / Actual	Ultra High risk	Others	Classification error
Ultra High risk	26	0	0.000
Others	13	1	0.928

Furthermore, for class “Others” the model predicted that 13 patients belonged to class “Ultra High risk” but they actually belonged in class “Others”. This model

before the variable selection wasn't satisfying. In Table 14, we present the final random forest which produced by variable elimination.

Table 14: Final random forest confusion matrix for classes "Ultra High risk" and "Others".

Type of random forest: classification			
Number of trees: 2000			
Number of variables tried at each split: 2			
OOB estimate of error rate: 7.5%			
Confusion matrix:			
Predicted / Actual	Ultra High risk	Others	Classification error
Ultra High risk	24	2	0.076
Others	1	13	0.071

In the end of the variable selection algorithm the remaining and most important CpG sites were 6. This was a reasonably good result because we found a very small set of CpG sites which achieved very good predictive accuracy. In Table 14, the model classified correctly 24/26 patients in class "Ultra High risk" with classification error 7.6%. Only two patients of the original class "Ultra High risk" misclassified. The model classified correctly 13/14 patients in class "Others" and it classified wrongly one patient in class "Ultra High risk" with classification error 7.1%. This means that the model classified the patient in class "Ultra High risk" but actually this patient belonged in class "Others".

It is obvious that after the variable selection the random forest improved for 6 CpG sites than for 6721 CpG sites. Also these 6 CpG sites seems to achieve good predictive accuracy instead of 6721.

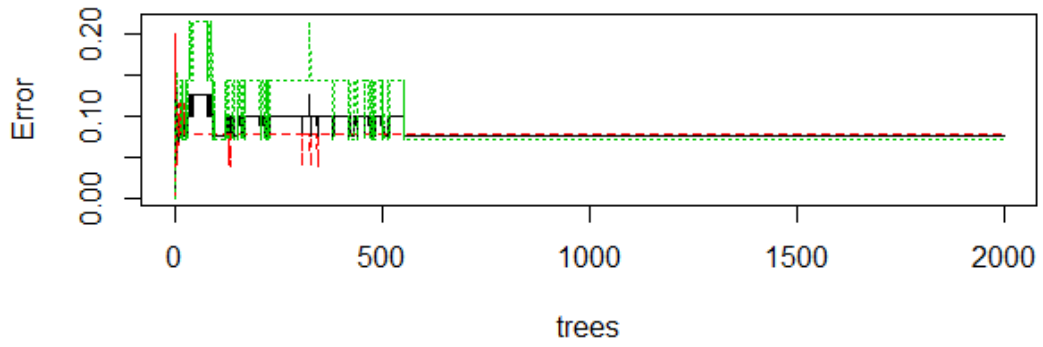


Figure 17: The OOB error rate is displayed of classes "Ultra High risk" and "Others" of 6 CpG sites for different number of trees.

In Figure 17 is represented the error of each class and the OOB error of this procedure for each tree. The red dashed line represents the error rate of class “Ultra High risk” in each tree, green dashed line represents the error rate of class “Others” in each tree and the black line represents the OOB error rate in each tree. We observe that after ≈600 trees the OOB error rate is stabilized.

In order to find the prediction error rate, we evaluated the “.632+” bootstrap method, for the remaining 6 CpG sites, five different times for 200 bootstrap samples and took the average prediction error rate of them. In the Table 15, we present the prediction error rate values for each different run.

Table 15: The prediction error rate is displayed, which was calculated with the “.632+” bootstrap method for classes "Ultra High risk" and "Others" of 6 CpG sites.

Number of runs	Prediction error rate
1	0.1247
2	0.1274
3	0.1374
4	0.1269
5	0.1250

The average prediction error rate has value 0.1282. For example in the first run the resubstitution error was equal to zero, the leave-one-out bootstrap was equal to 0.1701, and the weight was equal to 0.7330. We note that the leave-one-out bootstrap error, and the weight are less than in the previous prediction error rates. This means that this procedure achieved better the objective of aggressively reducing the set of selected genes.

Based on bibliography this was a reasonably good and comparable value. We have to mention, that these methods were applied to a real dataset with big heterogeneity and the results achieved to have good predictive accuracy.

Moreover, we used these 6 CpG sites as before to evaluate two other methods in order to see how they respond. The below heatmap in figure 18 evaluated based on the selected 6 CpG sites, which have been taken after the variable selection procedure for classes “Ultra High risk” and “Others”. In this heatmap, we note that the patients who belonged in the first cluster (left) have the same clinical-biological data. On the other hand, the patients who belonged to the second cluster (right), couldn’t manage to group satisfying based on their clinical-biological data. Furthermore, we observed that between these two clusters of patients there are completely different DNA methylation values.

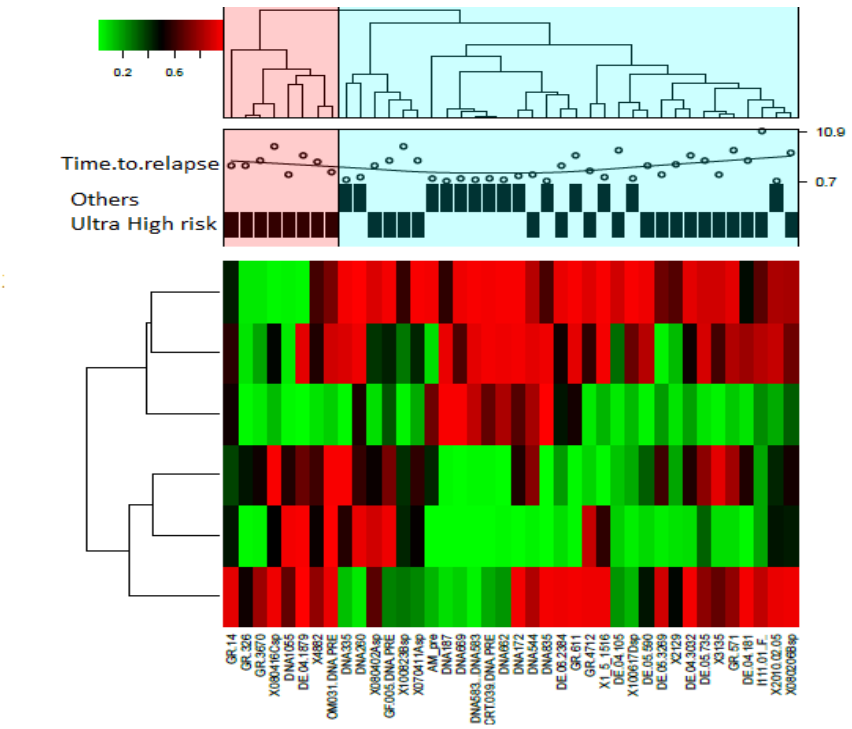


Figure 18: Heatmap of 6 CpG sites.

Hence, we concluded that after the variable selection method these 6 CpG sites yield to a better grouping of patients than the grouping made with 6721 CpG sites. Also we find changes between DNA methylation values of CpG sites in two clusters.

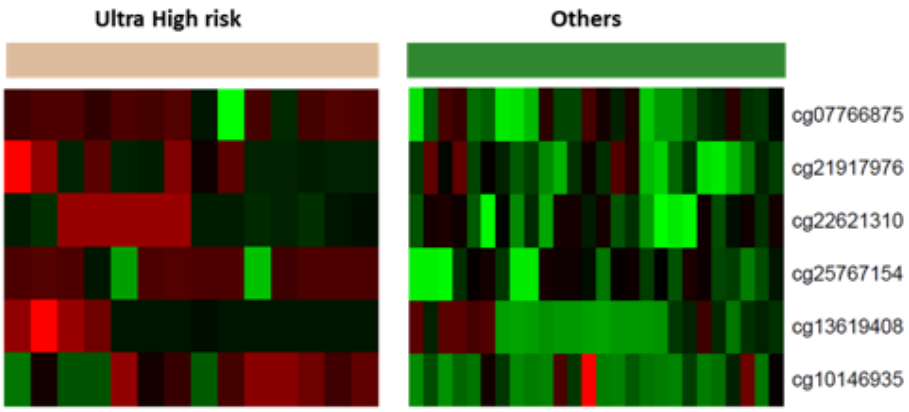


Figure 19: Classes “Ultra High risk”, “Others” of 6 CpG sites.

In order to see the difference between the DNA methylation values of patients we separated the patients based on their classes. As we observe in Figure 19, the patients who belonged in class “Ultra High risk” had most of their CpG sites hypermethylated. On the other hand, the patients who belonged in class “Others” had their CpG sites hypomethylated. This was very important because for the same CpG sites, we found different DNA methylation values on patients of different classes. As before, the last method which remain to be evaluated, is the method of principal component analysis.

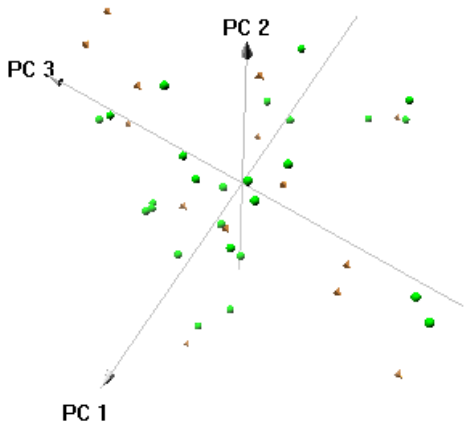


Figure 20: Principal component analysis of 6721 CpG sites for classes “Ultra High risk” and “Others”.

The above principal component analysis in figure 20 was based on 6721 CpG sites and on classes “Ultra High risk” and “Others”. Each point represents a patient. Beige points represent the patients who belonged in class “Ultra High risk” and green points represent the patients who belonged in class “Others”. As we can see, the patients couldn’t manage to group based on their clinical-biological data and we couldn’t see a clear separation between them.

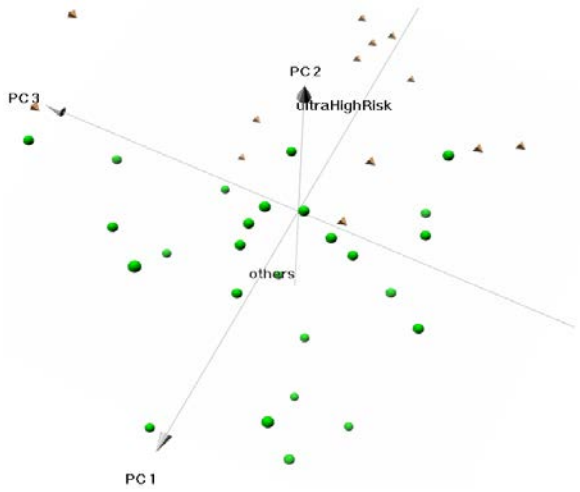


Figure 21: Principal component analysis of 6 CpG sites for classes “Ultra High risk” and “Others”.

The principal component analysis in Figure 21 was based on 6 CpG sites taken from the variable selection procedure which was applied before. The difference between these two plots of figures 20 and 21 was very large. In Figure 21, the patients tended to group together and only 1 patient of class “Ultra High risk” and 1 patient of class “Others” were in the wrong side.

Therefore, we concluded that heatmap and principal component analysis methods achieved better performance of grouping patients to their classes for these 6 CpG sites than 6721 CpG sites.

Finally, in order to check the performance of our classifier, we evaluated ROC analysis which shows us the performance of our classifier in binary problems.

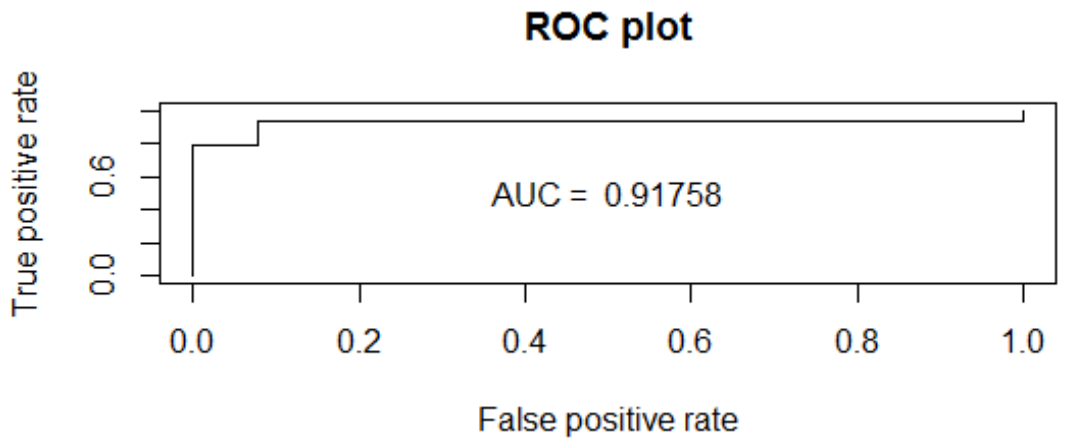


Figure 22: ROC plot of 11 CpG sites.

In Figure 22 we see the performance of our classifier. We note that the area under curve (AUC) ≈ 91.7%. This means that our classifier was perfect. Values bigger than 0.9 means that the classifier's performance is perfect.

In Table 16 we present the votes of classes "Ultra High risk" and "Others", the true class of each patient, the FPR and TPR. We note that we started by sorting the column "Others" in decreasing order and the cutoffs are defined to be the numbers of column "Others". We observed that the patients 12 and 13, who have true class "Ultra High risk", achieved higher probability to be classified in the class "Others" than in the class "Ultra High risk". In this case, the FPR of the patient 12 was calculated as FPR=1/26, and for the patient 13 was calculated as FPR=2/26.

Table 16: Votes of classes "Ultra High risk" and "Others".

Patients	**Probability "Ultra High risk"**	**Probability "Others"**	**True Class**	**FPR**	**TPR**
				0.00000000	0.00000000
1	0.00131406	0.998685940	"Others"	0.00000000	0.07142857
2	0.02567568	0.974324324	"Others"	0.00000000	0.14285714
3	0.04949239	0.950507614	"Others"	0.00000000	0.21428571
4	0.12500000	0.875000000	"Others"	0.00000000	0.28571429
5	0.15006821	0.849931787	"Others"	0.00000000	0.35714286

6	0.16666667	0.833333333	“Others”	0.00000000	0.42857143
7	0.19333333	0.806666667	“Others”	0.00000000	0.50000000
8	0.22544952	0.774550484	“Others”	0.00000000	0.57142857
9	0.26153846	0.738461538	“Others”	0.00000000	0.64285714
10	0.27052490	0.729475101	“Others”	0.00000000	0.71428571
11	0.31400283	0.685997171	“Others”	0.00000000	0.78571429
12	0.41374663	0.586253369	“Ultra High risk”	0.03846154	0.78571429
13	0.44460432	0.555395683	“Ultra High risk”	0.07692308	0.78571429
14	0.45481050	0.545189504	“Others”	0.07692308	0.85714286
15	0.47008547	0.529914530	“Others”	0.07692308	0.92857143
16	0.58310249	0.416897507	“Ultra High risk”	0.11538462	0.92857143
17	0.70661157	0.293388430	“Ultra High risk”	0.15384615	0.92857143
18	0.73772791	0.262272090	“Ultra High risk”	0.19230769	0.92857143
19	0.74229692	0.257703081	“Ultra High risk”	0.23076923	0.92857143
20	0.74668435	0.253315650	“Ultra High risk”	0.26923077	0.92857143
21	0.79382889	0.206171108	“Ultra High risk”	0.30769231	0.92857143
22	0.82030178	0.179698217	“Ultra High risk”	0.34615385	0.92857143
23	0.82065217	0.179347826	“Ultra High risk”	0.38461538	0.92857143
24	0.82421875	0.175781250	“Ultra High risk”	0.42307692	0.92857143
25	0.82794118	0.172058824	“Ultra High risk”	0.46153846	0.92857143
26	0.86046512	0.139534884	“Ultra High risk”	0.50000000	0.92857143
27	0.86856369	0.131436314	“Ultra High risk”	0.53846154	0.92857143
28	0.88811189	0.111888112	“Ultra High risk”	0.57692308	0.92857143

29	0.91292876	0.087071240	“Ultra High risk”	0.61538462	0.92857143
30	0.92408377	0.075916230	“Ultra High risk”	0.65384615	0.92857143
31	0.93108108	0.068918919	“Ultra High risk”	0.69230769	0.92857143
32	0.96986301	0.030136986	“Ultra High risk”	0.73076923	0.92857143
33	0.97010870	0.029891304	“Ultra High risk”	0.76923077	0.92857143
34	0.97503285	0.024967148	“Ultra High risk”	0.80769231	0.92857143
35	0.98424069	0.015759312	“Ultra High risk”	0.84615385	0.92857143
36	0.98770492	0.012295082	“Ultra High risk”	0.88461538	0.92857143
37	0.98850575	0.011494253	“Ultra High risk”	0.92307692	0.92857143
38	0.99007092	0.009929078	“Ultra High risk”	0.96153846	0.92857143
39	0.99316940	0.006830601	“Ultra High risk”	1.00000000	0.92857143
40	0.99709724	0.002902758	“Others”	1.00000000	1.00000000

Additionally, we investigated the behavior of the variable selection method (varSelRF) for different values of *ntree* and *mtry.* Specifically in Table 17, we made 32 different runs in the same dataset for *ntree*= 64, 128, 500, 1000, 2000(default), 3000, 4000, 5000 and for each *ntree* values we chosen *mtry*= 1, $\sqrt{\mathrm{M}}/2$, $\sqrt{\mathrm{M}}$(default), $\sqrt{\mathrm{M}} * 2$, where M was the number of variables (CpG sites) included in the dataset and we reported the OOB error rate for each procedure. These trials have been made to the same dataset, which concluded 40 patients with 6721 CpG sites with classes “Ultra High risk” and “Others”.

We have to note, that this error isn’t the prediction error of the procedure. This error was downwardly biased, and represents the error of the last random forest, which evaluated with the remaining variables. Finally, the column variables selected represents the number of the remaining variables that achieved the best OOB error rate.

Table 17: Iterations results.

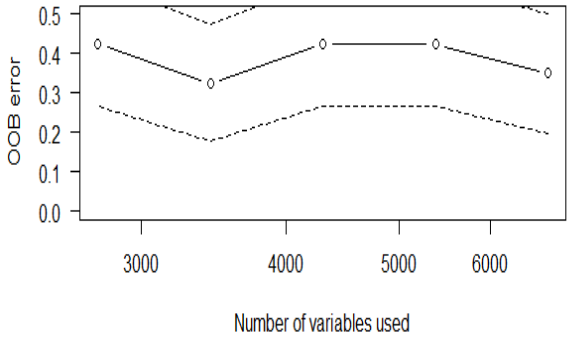
Ntree	Mtry	Variables Selected	OOB
64	1	3442	0.325
	$\sqrt{\mathrm{M}}/2$	5377	0.25
	$\sqrt{\mathrm{M}}$	152	0.075
	$\sqrt{\mathrm{M}} * 2$	3442	0.20
128	1	2754	0.30
	$\sqrt{\mathrm{M}}/2$	1410	0.15
	$\sqrt{\mathrm{M}}$	2754	0.30
	$\sqrt{\mathrm{M}} * 2$	4302	0.30
500	1	237	0.10
	$\sqrt{\mathrm{M}}/2$	62	0.025
	$\sqrt{\mathrm{M}}$	26	0.05
	$\sqrt{\mathrm{M}} * 2$	2	0.10
1000	1	32	0.025
	$\sqrt{\mathrm{M}}/2$	152	0.05
	$\sqrt{\mathrm{M}}$	40	0.025
	$\sqrt{\mathrm{M}} * 2$	32	0.075
2000	1	14	0.05
	$\sqrt{\mathrm{M}}/2$	40	0.025
	$\sqrt{\mathrm{M}}$	14	0.05
	$\sqrt{\mathrm{M}} * 2$	2	0.10
3000	1	17	0.125
	$\sqrt{\mathrm{M}}/2$	62	0.025
	$\sqrt{\mathrm{M}}$	2	0.10
	$\sqrt{\mathrm{M}} * 2$	2	0.10
4000	1	40	0.05
	$\sqrt{\mathrm{M}}/2$	26	0.025

	$\sqrt{\mathrm{M}}$	3	0.075
	$\sqrt{\mathrm{M}} * 2$	3	0.10
5000	1	21	0.05
	$\sqrt{\mathrm{M}}/2$	4	0.05
	$\sqrt{\mathrm{M}}$	2	0.075
	$\sqrt{\mathrm{M}} * 2$	2	0.10

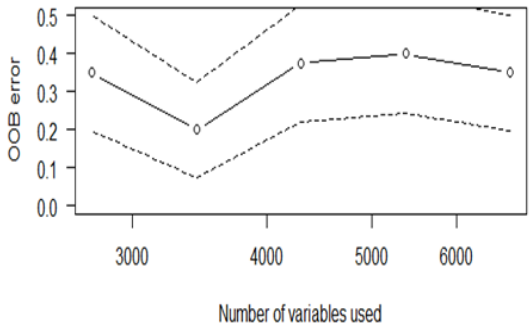
We concluded that after these trials, highest number of *ntree* and *mtry* values yields to better results. The desired result for the model, was to conclude to a small number of CpG sites which can achieve a small OOB error rate. This may happens, because with highest *ntree* number we increased the stability of variable importance. This was important because the variable selection method (varSelRF) eliminates at each step the variables based on their variable importance. Also with highest number of *mtry* values, the algorithm searches more variables for the best split. In this way, the probability of choosing the more informative variable for the split was increased, and the results tended to be better than for lower *mtry* values.

In Figures 23 and 24 below represent the variable selection procedure for specific *ntree* and *mtry* values. Because the number of runs was very high, we will only present a few figures of selected procedures. In the following figures, our claim that the results are better for higher *ntree* and *mtry* values is verified. We achieve to select only few variables with good predictive accuracy. In addition, for highest values of *ntree* and *mtry* the OOB error rate stability is increased. We note that the OOB error rate is biased down, and the prediction error rate is calculated in Table 16. In Figure 23, we present the change in OOB error with the number of variables used at each iteration. In Figure 24, we present the change in OOB error with the number of trees used at each iteration.

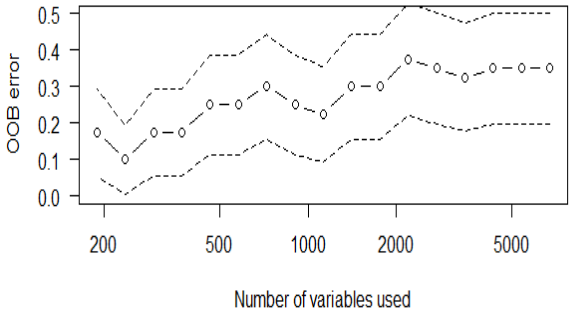
ntree=64, *mtry*=1



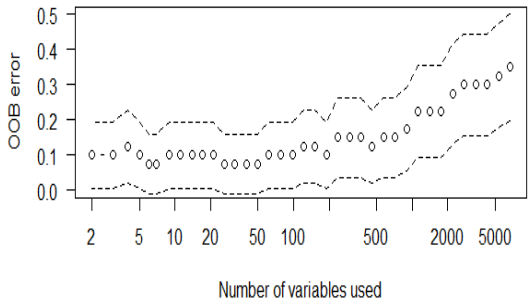
ntree=64, *mtry*=$\sqrt{\mathrm{M}} * 2$



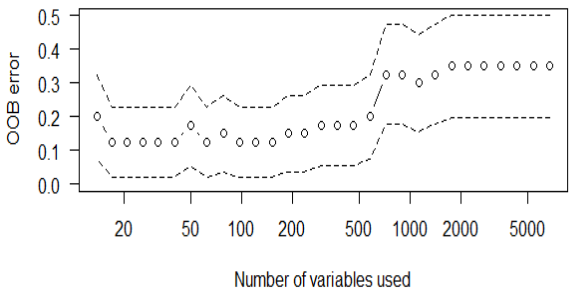
ntree=500, *mtry*=1



ntree=500, *mtry*=$\sqrt{\mathrm{M}} * 2$



ntree=3000, *mtry*=1



ntree=3000, *mtry*=$\sqrt{\mathrm{M}} * 2$

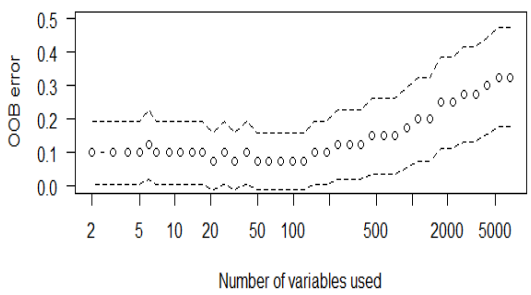
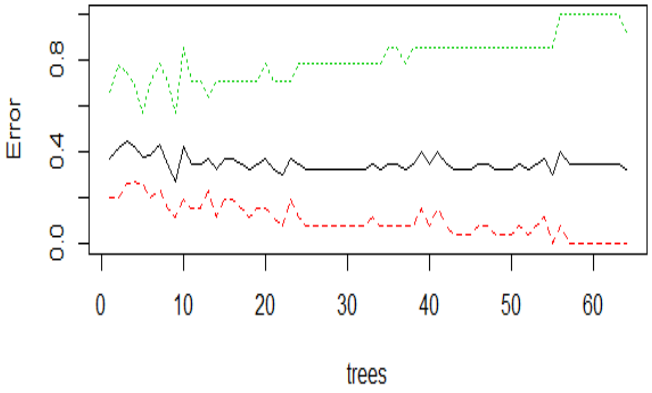
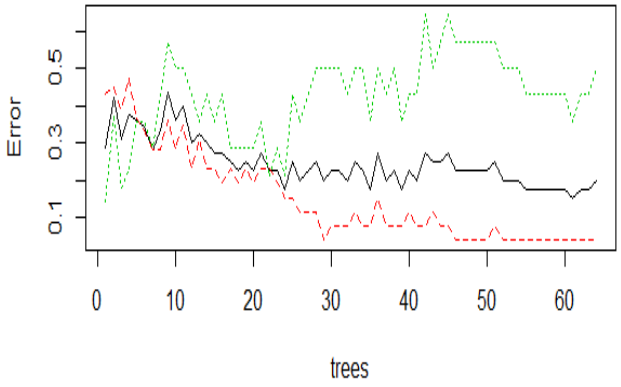


Figure 23: Number of variables used from right to left vs OOB error.

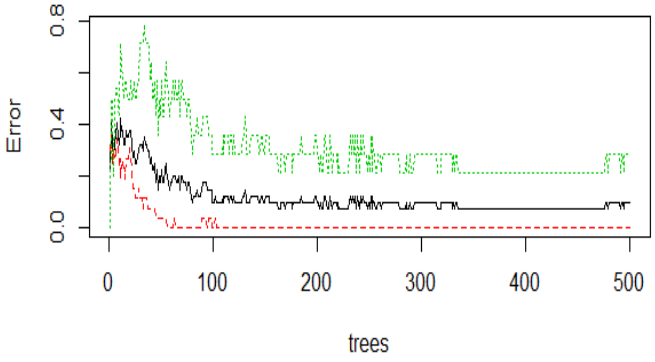
ntree=64, *mtry*=1



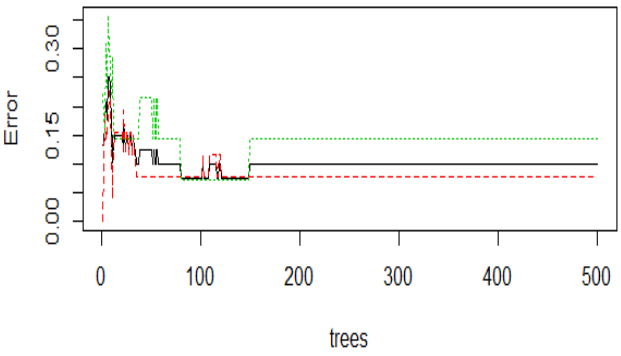
ntree=64, *mtry*=$\sqrt{\mathrm{M}} * 2$



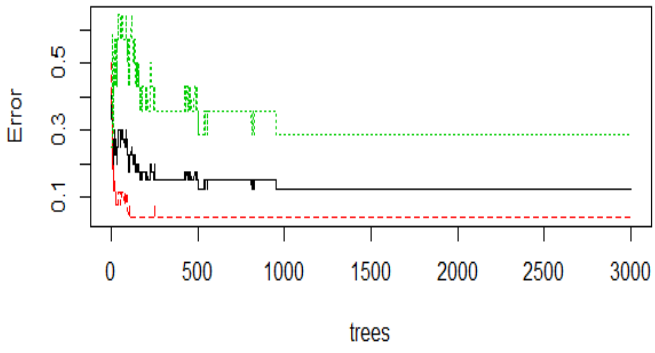
ntree=500, *mtry*=1



ntree=500, *mtry*=$\sqrt{\mathrm{M}} * 2$



ntree=3000, *mtry*=1



ntree=3000, *mtry*=$\sqrt{\mathrm{M}} * 2$

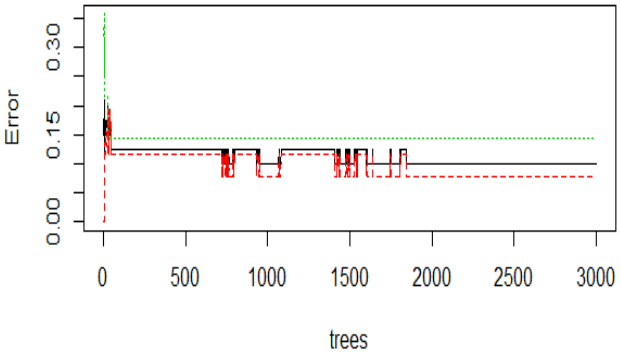


Figure 24: The OOB error is displayed for different number of trees.

Finally, in order to investigate the behavior of the most informative variables we introduce the following graph, which represents the variable importance of each of the six most informative variables for different *mtry* values. The six most informative CpG variables tend to appear as selected variables in other procedures with high variable importance too.

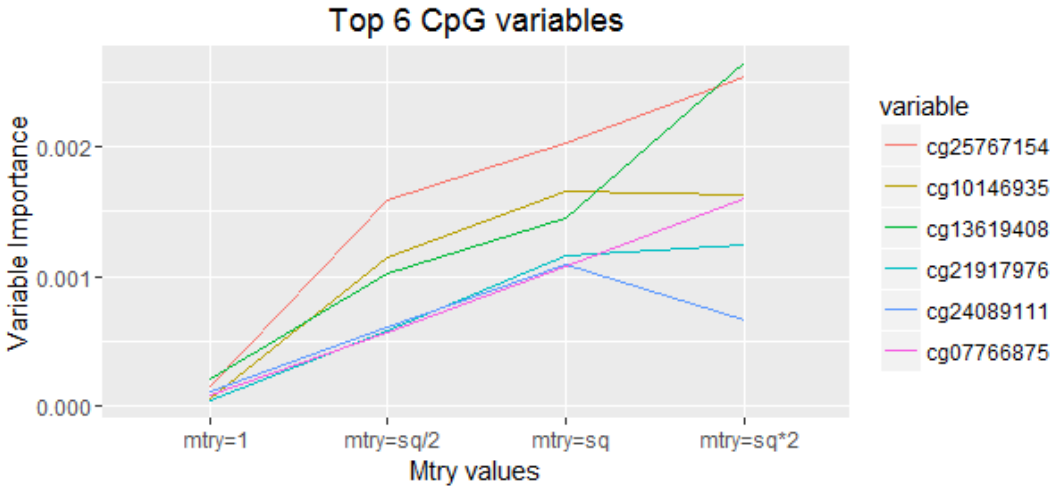


Figure 25: The six most informative CpG variables.

Based on figure 25, we concluded that for highest *mtry* values as previously the most informative variables were distinguished. The most informative variable was the one with the red and green line for *mtry*=$\sqrt{\mathrm{M}} * 2$. In Table 16, we present the prediction error estimate of our selected variable selection procedures.

Table 16: The prediction error rate is displayed, which was calculated with the ".632+" bootstrap method, for different numbers of *ntree* and *mtry* values.

Ntree / Mtry	64	500	3000
1	0.3462	0.2412	0.2429
$\sqrt{\mathrm{M}} * 2$	0.3305	0.1335	0.0978

As we can see, for *mtry*=$\sqrt{M} * 2$ the prediction error rate had the lowest values for all *ntree* values. We note, that the prediction error rate is the error obtained from the evaluation of the ".632+" bootstrap method which was mentioned and explained in Chapter 3. Moreover, we observed that for highest *ntree* values, the prediction error rate decreased for both *mtry* values. Finally, we present a plot in Figure 26, in order to see the behavior of the prediction error rate for the different values of *ntree* and *mtry*.

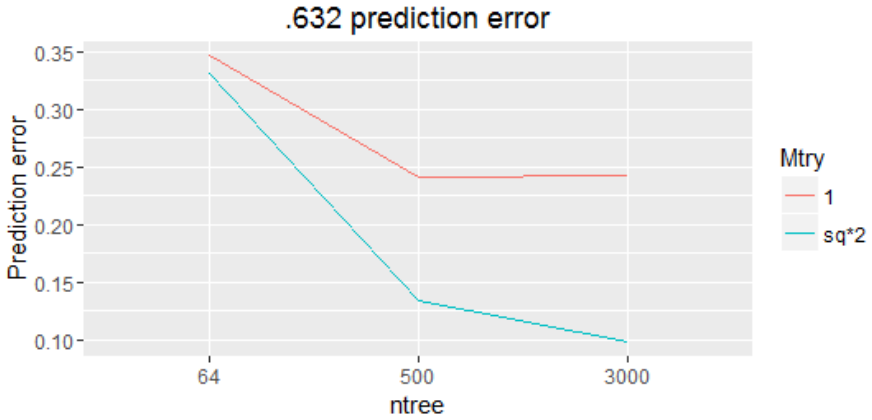


Figure 26: The prediction error estimate graph of Table 16.

The prediction error rate tended to decrease for highest *ntree* values. This may happen because with highest *ntree* and mtry values, the variable selection method which was based on the variable importance given from random forest algorithm was more stable than for lowest *ntree* values. Finally, we observed that for *ntree*=64 for both mtry values, the prediction error rates were too close, and this may happens because the variable selection method tends to be more unstable for low *ntree* values.

5. Conclusions

The aim of this study was to detect a small set of CpG sites among the overall number 463442, which achieved good predictive accuracy, and classify patients correctly to their pre-defined classes. Moreover, we investigated the importance of *ntree* and *mtry* values, by changing their values and executed the variable selection algorithm repeatedly.

After a lot of trials, we managed to have a set of 6 CpG sites for patients who belonged in the grouping which was based on the months each patient relapses and 11 CpG sites for patients who belonged in the grouping which was based on their differential methylated CpG sites. Moreover, these CpG sites from both groupings achieved good predictive accuracy and their roc curves achieved excellent performance. In addition, comparing the heatmap and principal component analysis plots, the selected CpG sites after the variable selection algorithm tended to group the patients in their pre-defined classes better than 6721 CpG sites. It is worth mentioning that random forest algorithm couldn't achieve good predictive accuracy and run in finite time for large sets of CpG sites.

Considering the parametrization of *ntree* and *mtry* values, after a lot of trials we concluded that for highest *ntree* and *mtry* values the variable selection algorithm improved, eliminated irrelevant CpG sites and selected only the most informative. Additionally, the variable selection algorithm resulted a small set of genes which achieved good predictive accuracy.

Further research could be done with the help of networks. The combination of networks and DNA methylation sites analysis is an undiscovered and uprising field, because with the help of networks we can discover groups, patterns and analyze several of nodes behaviors. It would be very useful in the future, to construct an algorithm, which has the ability to separate patients in their groups based on their DNA methylation values and uncover interesting patterns.

Bibliography

1. Benjamini, Y. and Y. Hochberg (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing." Journal of the royal statistical society. Series B (Methodological): 289-300.

2. Benjamini, Y. and D. Yekutieli (2001). "The control of the false discovery rate in multiple testing under dependency." Annals of statistics: 1165-1188.

3. Breiman, L. (2001). "Random forests." Machine learning **45**(1): 5-32.

4. Breiman, L., et al. (1984). Classification and regression trees, CRC press.

5. Cahill, N., et al. (2013). "450K-array analysis of chronic lymphocytic leukemia cells reveals global DNA methylation to be relatively stable over time and similar in resting and proliferative compartments." Leukemia **27**(1): 150-158.

6. Chen, X., et al. (2011). "The use of classification trees for bioinformatics." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **1**(1): 55-63.

7. Dessì, N., et al. (2012). "Pre-filtering Features in Random Forests for Microarray Data Classification." New Frontiers in Mining Complex Patterns (NFMCP 2012) **60**.

8. Diaz-Uriarte, R. (2007). "GeneSrF and varSelRF: a web-based tool and R package for gene selection and classification using random forest." BMC bioinformatics **8**(1): 328.

9. Efron, B. and R. Tibshirani (1997). "Improvements on cross-validation: the 632+ bootstrap method." Journal of the American Statistical Association **92**(438): 548-560.

10. Fayyad, U., et al. (1996). "From data mining to knowledge discovery in databases." AI magazine **17**(3): 37.

11. Genuer, R., et al. (2008). "Random Forests: some methodological insights." arXiv preprint arXiv:0811.3619.

12. Jiang, W. and R. Simon (2007). "A comparison of bootstrap methods and an adjusted bootstrap approach for estimating the prediction error in microarray classification." Statistics in medicine **26**(29): 5320-5334.

13. Landau, D. A., et al. (2014). "Locally disordered methylation forms the basis of intratumor methylome variation in chronic lymphocytic leukemia." Cancer Cell **26**(6): 813-825.

14. Liaw, A. and M. Wiener (2002). "Classification and regression by randomForest." R news **2**(3): 18-22.

15. Oakes, C. C., et al. (2014). "Evolution of DNA methylation is linked to genetic aberrations in chronic lymphocytic leukemia." Cancer discovery **4**(3): 348-361.

16. Oshiro, T. M., et al. (2012). How many trees in a random forest? MLDM, Springer.

17. Strobl, C. and A. Zeileis (2008). Danger: High power!–exploring the statistical properties of a test for random forest variable importance. Proceedings of the 18th International Conference on Computational Statistics, Porto, Portugal, Citeseer.

18. Chen, X. and H. Ishwaran (2012). "Random forests for genomic data analysis." Genomics **99**(6): 323-329.

19. Grömping, U. (2012). "Variable importance assessment in regression: linear regression versus random forest." The American Statistician.

20. Hastie, T., et al. (2005). "The elements of statistical learning: data mining, inference and prediction." The Mathematical Intelligencer **27**(2): 83-85.

21. Strobl, C., et al. (2008). "Conditional variable importance for random forests." BMC bioinformatics **9**(1): 1.

22. Strobl, C., et al. (2009). "An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests." Psychological methods **14**(4): 323.

23. Williams, G. (2011). Data mining with Rattle and R: The art of excavating data for knowledge discovery, Springer Science & Business Media.

24. Zhao, S., et al. (2014). "Advanced heat map and clustering analysis using heatmap3." BioMed research international **2014**.

25. R Core Team (2016). R: A language and environment for statistical computin
g. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.
R-project.org/.