ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΩΝ ΚΑΤΑΣΚΕΥΩΝ ΚΑΙ
ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

Ανάπτυξη Αλγορίθμου Επιβλεπόμενης Μάθησης για την Βελτιστοποίηση Στόχευσης Δραστικών Ουσιών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χαρίλαος Κουρουμπάς

Επιβλέπων: Λεωνίδας Αλεξόπουλος

Λέκτορας Ε.Μ.Π.

Αθήνα, Οκτώβριος 2011

Ευχαριστίες

Θεωρώ χρέος μου πριν ξεκινήσει η ανάπτυξη και η παρουσίαση του θέματος της Διπλωματικής μου εργασίας να ευχαριστήσω θερμά ορισμένα άτομα που με βοήθησαν και συνέβαλαν σημαντικά σε αυτή.

Κατ' αρχήν θα ήθελα να ευχαριστήσω τον πατέρα μου, τη μητέρα μου και τον αδερφό μου που με στήριξαν όλα τα προηγούμενα χρόνια σε όλες μου τις προσπάθειες και συνεχίζουν να με στηρίζουν.

Ευχαριστώ θερμά τον Λέκτορα Αλεξόπουλο Λεωνίδα, για την καθοδήγηση του και την συνεργασία μας καθ' όλη τη διάρκεια εκπόνησης της εργασίας.

Επίσης ευχαριστώ θερμά τον συνάδελφο και φίλο μου Γεώργιο Μανίκη, για τις πολύτιμες συμβουλές που μου έδωσε όποτε τις χρειάστηκα.

Κουρουμπάς Χαρίλαος

Περιεχόμενα

Περίληψη

Ο σκοπός αυτής της διπλωματικής εργασίας αφορά την ανάπτυξη αλγορίθμου επιβλεπόμενης μάθησης, και συγκεκριμένα Support Vector Machines, για την βελτιστοποίηση στόχευσης δραστικών ουσιών.

Αναλύονται τα στάδια εξέλιξης μιας δραστικής ουσίας, και οι βασικές ιδιότητες των SVM. Οι μετρήσεις που χρησιμοποιούνται από τον αλγόριθμο προέρχονται από κυτταρικές σειρές ηπατικού καρκίνου στις οποίες έχουν χορηγηθεί ουσίες με γνωστά αποτελέσματα, και νέες ουσίες για τις οποίες θα γίνει μια πρόβλεψη σχετικά με την αποτελεσματικότητα τους.

Για την ανάπτυξη του αλγορίθμου γράφτηκε κώδικας σε περιβάλλον MATLAB και το toolbox που χρησιμοποιήσαμε για τη διαδικασία της δυαδικής κατηγοριοποίησης, τόσο για την εκπαίδευση στην οποία δουλεύαμε με γνωστές ουσίες, όσο και για την πρόβλεψη με τις νέες ουσίες είναι το LIBSVM toolbox.

Λέξεις Κλειδιά

Εξέλιξη φαρμάκων, Ανακάλυψη φαρμάκων, Υπατικά κύτταρα, Κυτταρικά σήματα, κυτταροσειρές, Support Vector Machines, SVM, Κατηγοριοποίηση, Κανονικοποίηση, MATLAB, Εργαλειοθήκη LIBSVM

Abstract

The scope of this thesis involves the development of an algorithm concerning supervised machine learning methods, specifically Support Vector Machines, which is used for the optimization of active compound targeting.

We analyze the way an active compound is developed, and the basic functions of Support vector machines. The data that the algorithm utilizes come from hepatic cancer cell lines which have been treated to compounds, the results of which we are familiar with, and new compounds for which we will attempt to predict whether they will function properly or not.

The algorithm was compiled using MATLAB and the toolbox we used for the process of binary classification, both for training when working with known compounds and predicting when working with new compounds, is the LIBSVM toolbox.

Keywords

Drug development, Drug discovery, Hepatic cells, Cell signals, Cell lines, Support Vector Machines, SVM, Classification, Normalization, MATLAB, LIBSVM toolbox

1. Εισαγωγή

Η διαδικασία της εξέλιξης φαρμάκων είναι ιδιαίτερα χρονοβόρα και ακριβή, και για το λόγο αυτό διερευνώνται συνεχώς νέες μέθοδοι για να επιταχυνθεί η διαδικασία της εξέλιξης, βελτιστοποιώντας το κόστος της.

Είναι γεγονός ότι τα τελευταία χρόνια οι υπολογιστές έχουν εξελιχθεί σε χρήσιμα εργαλεία με άπειρες εφαρμογές σε όλους τους τομείς. Για να γίνουν πιο προσιτοί οι στόχοι που θέτει η διαδικασία της εξέλιξης φαρμάκων, η χρήση ηλεκτρονικών υπολογιστών και διάφορων υπολογιστικών μοντέλων είναι απαραίτητη.

Σε αυτή τη διπλωματική εργασία θα ερευνήσουμε την δυνατότητα και αποτελεσματικότητα της χρήσης των Support Vector Machines για την εξέλιξη φαρμάκων.

Τα SVM’s είναι ένα εργαλείο κατηγοριοποίησης δεδομένων σε όσες κατηγορίες επιθυμούμε εμείς. Στη συγκεκριμένη εργασία θα ασχοληθούμε αποκλειστικά με δυαδική κατηγοριοποίηση, όπου αρχικά θα εκπαιδεύσουμε τον αλγόριθμο χρησιμοποιώντας δεδομένα από έρευνες για υπάρχοντα φάρμακα το κλινικό αποτέλεσμα των οποίων είναι γνωστό, και μετά θα χρησιμοποιήσουμε αυτά τα δεδομένα για να κάνουμε μια αρχική πρόβλεψη σχετικά με την αποτελεσματικότητα νέων φαρμάκων σε κλινικές δοκιμές.

2. Εξέλιξη φαρμάκων

2.1. Εισαγωγή στην εξέλιξη φαρμάκων

Η εξέλιξη φαρμάκων είναι ένας γενικός όρος ο οποίος περιγράφει την απαραίτητη διαδικασία ώστε ένα φάρμακο ή μία συσκευή να φτάσει στην αγορά. Περιλαμβάνει την ανακάλυψη του φαρμάκου ή της συσκευής, προκλινικές δοκιμές (σε μικροοργανισμούς ή ζώα) και κλινικές δοκιμές (σε υγιείς εθελοντές ή ασθενείς).

Τα στάδια στα οποία χωρίζεται είναι τα εξής:

- Ανακάλυψη δραστικής ουσίας: Στη φάση αυτή γίνεται έρευνα και θέτονται οι πρωτεϊνικοί στόχοι τους οποίους θα σημαδέψουν τα υποψήφια φάρμακα. Εδώ γίνεται σύνθεση χημικών ουσιών, και όσες από αυτές δείξουν υποσχόμενα αποτελέσματα περνούν στη φάση των προκλινικών δοκιμών.
- Προκλινικές δοκιμές: Στη φάση αυτή οι δραστικές ουσίες που διακρίθηκαν στο προηγούμενο στάδιο υποβάλλονται σε βιολογικές δοκιμές, όπου εξετάζονται ως προς τη σύνθεση, τη σταθερότητα και την τοξικότητα, σε κυτταρικό επίπεδο αρχικά και μετά σε κατάλληλα πειραματόζωα. Πάλι όσες από αυτές δείξουν επιθυμητά αποτελέσματα λαμβάνουν έγκριση να περάσουν στη φάση των κλινικών δοκιμών.
- Κλινικές δοκιμές: Στη φάση αυτή οι ουσίες χορηγούνται σε υγιείς εθελοντές και ασθενείς, ώστε να εξεταστούν ως προς την αποτελεσματικότητα, την τοξικότητα, τη δοσολογία και τέλος τα αποτελέσματα από μακροχρόνια χρήση. Στο στάδιο αυτό γίνεται και η ονοματολογία της δραστικής ουσίας (και από δραστική ουσία ονομάζεται «φάρμακο»). Όσα φάρμακα επιτύχουν σε αυτή τη φάση, αν λάβουν έγκριση από τις αρμόδιες αρχές, θα παραχθούν και θα μπορούν να προσφερθούν σε ιατρούς και τους ασθενείς τους με σκοπό την αντιμετώπιση ασθενειών.
- Διάθεση του φαρμάκου στην αγορά και παρακολούθηση του.

2.2. Ανακάλυψη φαρμάκων

Στους τομείς της ιατρικής, βιοτεχνολογίας και φαρμακολογίας, η ανακάλυψη φαρμάκων είναι η διαδικασία κατά την οποία ένα φάρμακο ανακαλύπτεται και κατασκευάζεται.

Στο παρελθόν τα περισσότερα φάρμακα έχουν ανακαλυφθεί αναγνωρίζοντας κάποιο ενεργό συστατικό είτε μέσω παραδοσιακών μεθόδων ή λόγω τύχης. Οι σύγχρονες μέθοδοι όμως απαιτούν καλύτερη κατανόηση ώστε να ελεγχθεί η ασθένεια ή η μόλυνση σε μοριακό και φυσιολογικό επίπεδο, και να θέσουν συγκεκριμένους στόχους.

Η διαδικασία της ανακάλυψης φαρμάκων περιλαμβάνει την αναγνώριση των συστατικών, της σύνθεσης, του χαρακτηρισμού, του προσυμπτωματικού ελέγχου και των αναμενόμενων θεραπευτικών αποτελεσμάτων. Μόλις μια ουσία δείξει την αξία της στις παραπάνω δοκιμές τότε περνά στο στάδιο της εξέλιξης φαρμάκου.

2.3. Στόχος της δραστικής ουσίας

Αρχικά πρέπει να οριστεί ο στόχος του φαρμάκου, δηλαδή η κυτταρική ή μοριακή δομή στην οποία αυτό θα δράσει. Οι στόχοι χωρίζονται σε δύο διακριτές κατηγορίες: υπάρχοντες και νέοι.

Οι υπάρχοντες στόχοι είναι αυτοί για τους οποίους υπάρχει αρκετή αλλά όχι πλήρη επιστημονική κατανόηση, και αρκετή βιβλιογραφία σχετικά με τον τρόπο που ενεργούν φυσιολογικά και παθολογικά. Όσο περισσότερες γνώσεις υπάρχουν για τον στόχο τόσο μικρότερη χρηματική και χρονική επένδυση χρειάζεται για την εξέλιξη της θεραπείας.

Οι νέοι στόχοι είναι όλοι αυτοί για τους οποίους δεν υπάρχουν αρκετές γνώσεις και έτσι ξεκινά η διαδικασία της ανακάλυψης του φαρμάκου. Αυτά συνήθως περιλαμβάνουν πρωτεΐνες που ανακαλύφθηκαν πρόσφατα, ή πρωτεΐνες η λειτουργία των οποίων έγινε πρόσφατα γνωστή ως αποτέλεσμα βασικής επιστημονικής έρευνας.

2.4. Προσυμπτωματικός έλεγχος και σχεδιασμός (screening and design)

Η διαδικασία της ανακάλυψης ενός φαρμάκου ενάντια σε έναν επιλεγμένο στόχο για μια συγκεκριμένη ασθένεια συνήθως περιλαμβάνει προσυμπτωματικό έλεγχο, όπου μεγάλος αριθμός χημικών δοκιμάζεται ως προς τη δυνατότητα του να τροποποιήσει τον εκάστοτε στόχο. Άλλη μια λειτουργία του προσυμπτωματικού ελέγχου είναι να δείξει πόσο επιλεκτικά είναι τα χημικά για τον επιλεγμένο στόχο. Το ιδανικό είναι να βρεθεί μία ουσία η οποία αντιδρά μόνο με το συγκεκριμένο στόχο και όχι άλλους. Σε περίπτωση που η ουσία αντιδρά και με άλλους στόχους, είναι πολύ πιθανό να προκαλέσει μεγάλα επίπεδα τοξικότητας όταν το φάρμακο φτάσει στη φάση των κλινικών δοκιμών όπου θα δοκιμάζεται σε ανθρώπους.

Γενικά είναι πάρα πολύ δύσκολο, έως απίθανο, να βρεθεί το τέλειο φάρμακο, πόσο μάλλον από τη διαδικασία του προσυμπτωματικού ελέγχου. Συνήθως όταν παρατηρούμε κάποια δραστηριότητα, τότε γίνεται προσπάθεια βελτιστοποίησης του φαρμάκου αυξάνοντας τη δραστηριότητα του στον επιλεγμένο στόχο και μειώνοντας τη δραστηριότητα του σε άσχετους στόχους.

Η διαδικασία αυτή θα χρειαστεί επαναληπτικές δοκιμές, κατά τις οποίες, ευελπιστούμε ότι οι ιδιότητες των ουσιών που χρησιμοποιήθηκαν θα βελτιωθούν, και θα καταφέρουν να φτάσουν για δοκιμές in vitro (πειράματα που πραγματοποιούνται σε αυστηρά ελεγχόμενες συνθήκες έξω από τους ζωντανούς οργανισμού) και in vivo (πειράματα που πραγματοποιούνται σε ιστούς εντός ζώντος οργανισμού).

Ο σχεδιασμός του φαρμάκου, όπου μελετώνται οι βιολογικές και φυσικές ιδιότητες του, είναι εξίσου σημαντικός με την εξέλιξη του διότι μπορεί να γίνει μία πρόβλεψη του τύπου των χημικών που θα χρειαστούν. Μόλις επιλεγεί μια σειρά ουσιών με τα απαιτούμενα χαρακτηριστικά, μία ή δύο ουσίες θα προταθούν για το στάδιο της εξέλιξης του φαρμάκου.

Αξίζει να αναφερθεί ότι οι πηγές για καινοτόμες χημικές δομές που εξελίσσονται για την αντιμετώπιση κάποιων αντιβακτηριδιακών θεραπειών, συχνά προέρχονται από φυσικές ουσίες. Οι ουσίες αυτές μπορεί να είναι από φυτά, μικρόβια, θαλάσσια ασπόνδυλα, κτλ.

2.5. Προκλινικές δοκιμές

Το στάδιο προκλινικών δοκιμών, αφορά την έρευνα που λαμβάνει χώρα πριν τις κλινικές δοκιμές (όπου γίνονται δοκιμές σε ανθρώπους) και συλλέγονται σημαντικά στοιχεία σχετικά με τη σκοπιμότητα, την επαναληπτική δοκιμασία και την ασφάλεια.

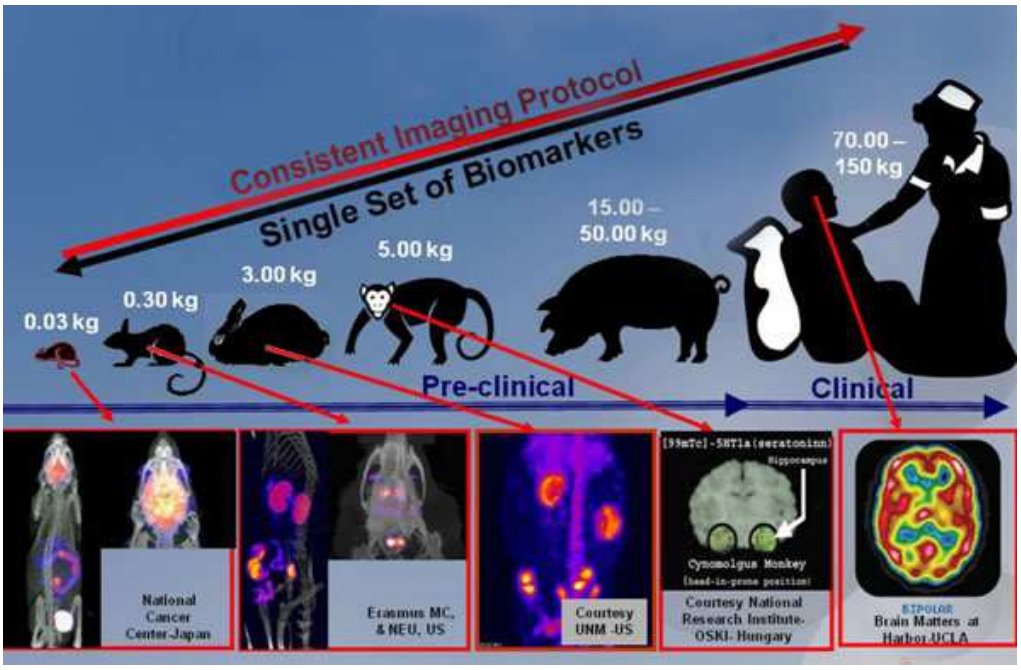
Εικόνα 2.1. Προκλινικές δοκιμές σε περιβάλλον εργαστηρίου.

Ο βασικός στόχος των προκλινικών δοκιμών είναι ο προσδιορισμός του προφίλ ασφαλείας του εκάστοτε προϊόντος. Τα προϊόντα αυτά μπορεί να είναι νέες ή βελτιωμένες ιατρικές συσκευές, φάρμακα, κτλ. Κάθε είδος προϊόντος διερευνάται με διαφορετικό τρόπο. Για παράδειγμα, τα φάρμακα υποβάλλονται σε δοκιμές φαρμακοδυναμικής (PD), φαρμακοκινητικής (PK) και τοξικότητας σε πειραματόζωα [1][2]. Τα στοιχεία που συλλέγονται επιτρέπουν στους ερευνητές να καθορίσουν μία ασφαλή αρχική δόση του φαρμάκου για χρήση στις κλινικές δοκιμές. Οι ιατρικές συσκευές που δεν συσχετίζονται με τη χορήγηση του φαρμάκου δεν χρειάζεται να υποβληθούν σε αυτές τις δοκιμές και περνούν κατευθείαν σε δοκιμές οι οποίες σχετίζονται με την ασφάλεια χρήσης της συσκευής και των υποσυστημάτων της.

Κάποιες ιατρικές συσκευές υποβάλλονται σε δοκιμές βιοσυμβατότητας οι οποίες βοηθούν να διαπιστωθεί αν η συσκευή και τα υποσυστήματα της είναι βιώσιμα σε ένα ανθρώπινο μοντέλο [3].

Στις περισσότερες περιπτώσεις γίνονται δοκιμές in vitro (πειράματα που πραγματοποιούνται σε αυστηρά ελεγχόμενες συνθήκες έξω από τους ζωντανούς οργανισμούς) και in vivo (πειράματα που πραγματοποιούνται σε ιστούς εντός ζώντος οργανισμού) [4] [5]. Οι έρευνες για την τοξικότητα ενός φαρμάκου συγκεντρώνονται κατά κύριο λόγο στα όργανα τα οποία στοχεύει το φάρμακο, καθώς και τις μακροχρόνιες παρενέργειες σχετικά με την καρκινογένεση.

Οι πληροφορίες οι οποίες αποκομίζονται από αυτές τις έρευνες είναι καθοριστικές για την έναρξη των δοκιμών στον άνθρωπο. Συνήθως οι δοκιμές σε πειραματόζωα περιλαμβάνουν τη χρήση δύο διαφορετικών ειδών. Αυτά που χρησιμοποιούνται συνήθως είναι ποντίκια και σκυλιά, ενώ σε κάποιες περιπτώσεις χρησιμοποιούνται πίθηκοι ή γουρούνια (Εικόνα 2.2). Η επιλογή του είδους σχετίζεται άμεσα με τις ομοιότητες που θα έχει η επίδραση του φαρμάκου στον άνθρωπο. Τα σκυλιά για παράδειγμα δεν είναι καλά πειραματόζωα για φάρμακα τα οποία χορηγούνται από το στόμα λόγω του πεπτικού τους συστήματος. Επίσης τα τρωκτικά δεν είναι κατάλληλα για αντιβιοτικά φάρμακα επειδή τους προκαλούν δυσμενή αποτελέσματα στην εντερική περιοχή [6].



Εικόνα 2.2. Χρήση πειραματόζωων στις προκλινικές δοκιμές και τελικά χορήγηση

στον άνθρωπο.

Αναλόγως τη λειτουργία του φαρμάκου, μπορεί να μεταβολιστεί με παρόμοιο ή εντελώς διαφορετικό τρόπο ανάμεσα στα διάφορα είδη, επηρεάζοντας τόσο την αποτελεσματικότητα όσο και την τοξικότητα. Οι περισσότερες έρευνες διεξάγονται σε πειραματόζωα μεγαλύτερου μεγέθους όπως σκύλοι, γουρούνια και πρόβατα, τα οποία επιτρέπουν τη δοκιμασία σε όργανα με μέγεθος παραπλήσιο των ανθρωπίνων.

Βασισμένα σε κλινικές δοκιμές, ορίζονται τα επίπεδα μη παρατηρήσιμων παρενεργειών (No Observable Effect Level, NOEL) τα οποία χρησιμεύουν στην εύρεση της δοσολογίας των κλινικών δοκιμών. Σε γενικές γραμμές περιλαμβάνεται ένα περιθώριο ασφαλείας της τάξεως του 1/100 το οποίο οφείλεται σε διαφορές μεταξύ των ειδών και των ασθενών.

Οι δοκιμές με χρήση πειραματόζωων έχουν περιοριστεί τα τελευταία χρόνια για ηθικούς και χρηματικούς λόγους. Πολλές έρευνες όμως συνεχίζουν να χρησιμοποιούν πειραματόζωα για τις δοκιμές τους, λόγω των απαραιτήτων για την εξέλιξη του φαρμάκου ανατομικών και φυσιολογικών ομοιοτήτων.

2.6. Κλινικές δοκιμές

2.6.1. Διεξαγωγή κλινικών δοκιμών

Οι κλινικές δοκιμές αφορούν ένα σύνολο διαδικασιών στην ιατρική έρευνα και την εξέλιξη φαρμάκων, οι οποίες θα προσφέρουν χρήσιμα δεδομένα συσχετισμένα με την ασφάλεια και την αποτελεσματικότητα κάθε ιατρικής παρέμβασης (Εικόνα 2.3). Οι δοκιμές αυτές λαμβάνουν χώρα μόνο αν η συλλογή πληροφοριών κατά τις προκλινικές δοκιμές είναι ικανοποιητική και λάβει έγκριση από τις αρχές της χώρας στην οποία εξελίσσεται το φάρμακο.

Εικόνα 2.3. Συνοπτική ανάλυση των φάσεων των κλινικών δοκιμών μέχρι δοθεί έγκριση για το φάρμακο.

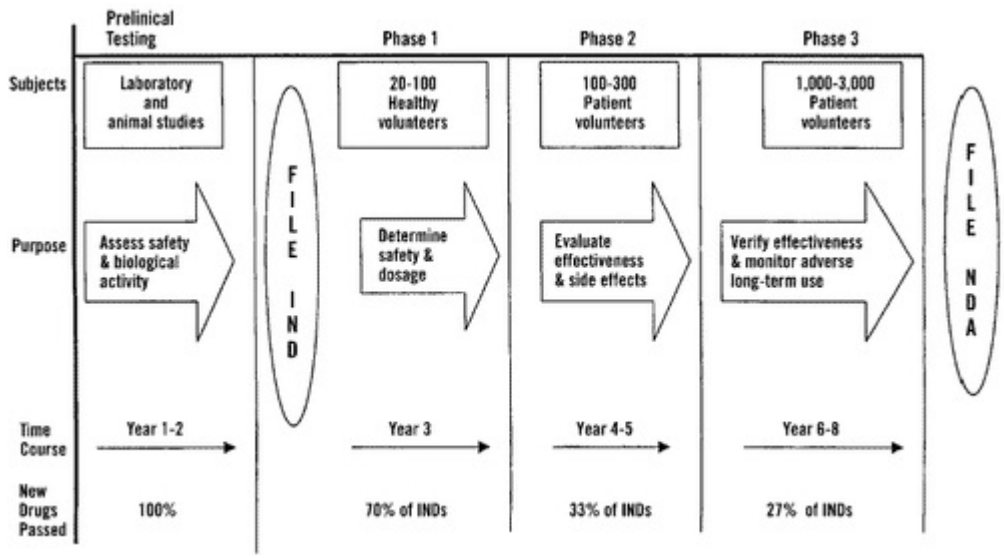
Αναλόγως το είδος του φαρμάκου και το στάδιο της εξέλιξης, οι ερευνητές επιστρατεύουν αρχικά λίγους υγιείς εθελοντές και/ή ασθενείς, στους οποίους δοκιμάζονται οι διάφορες λύσεις που έχουν προκύψει από τις προκλινικές δοκιμές. Εφόσον συλλέξουν θετικές πληροφορίες κατά τη χορήγηση του φαρμάκου σχετικά με την αποτελεσματικότητα και την ασφάλεια του, ο αριθμός των ασθενών αυξάνεται. Το

μέγεθος των κλινικών δοκιμών ποικίλει σημαντικά καθώς μπορεί να αποτελείται από ένα ερευνητικό κέντρο σε μία χώρα ή από πολλά ερευνητικά κέντρα παγκόσμια.

Συνήθως για τις κλινικές δοκιμές επιλέγονται ασθενείς με συγκεκριμένες παθήσεις, οι οποίοι επωφελούνται από τη χορήγηση θεραπειών οι οποίες δεν υπάρχουν ακόμα σε τελική μορφή. Στην πρώτη φάση χρησιμοποιούνται υγιείς εθελοντές οι οποίοι λαμβάνουν χρηματική ανταμοιβή για τις υπηρεσίες του. Κατά τις περιόδους δοσολογίας, οι εθελοντές παραμένουν στις ερευνητικές εγκαταστάσεις για διάστημα ενός έως τριάντα ημερών, και κάποιες φορές παραπάνω, για να γίνουν οι απαραίτητες μετρήσεις.

Κατά το σχεδιασμό μιας κλινικής δοκιμής, ο ερευνητής ξεκινά αναγνωρίζοντας τη φαρμακευτική αγωγή που θα χορηγηθεί. Εκεί διεξάγονται πιλοτικά πειράματα για τον σχεδιασμό της κλινικής δοκιμής, σχετικά με την αποτελεσματικότητα του φαρμάκου τόσο στην κλινική δοκιμή όσο και σε πραγματικές συνθήκες. Στις ΗΠΑ οι ηλικιωμένοι αποτελούν μόλις το 14% του πληθυσμού αλλά καταναλώνουν περισσότερο από το 1/3 των φαρμάκων [7]. Ακόμα και έτσι, συνήθως εξαιρούνται από τις δοκιμές επειδή τα προβλήματα υγείας τους και η συχνή χρήση φαρμάκων μπορεί να οδηγήσει σε αναξιόπιστα αποτελέσματα. Επίσης συνήθως εξαιρούνται γυναίκες, παιδιά και άτομα με παθήσεις διαφορετικές από αυτές που εξετάζονται [8].

Αφού προσδιοριστεί ο τύπος των ασθενών που θα επωφεληθούν από την φαρμακευτική αγωγή, επιστρατεύονται οι κατάλληλοι ασθενείς και συλλέγονται οι απαραίτητες πληροφορίες για αυτούς. Οι ασθενείς αυτοί είναι εθελοντές και συνήθως δεν πληρώνονται για τις δοκιμές. Οι πληροφορίες που συλλέγονται αφορούν τις ζωτικές ενδείξεις, την περιεκτικότητα του φαρμάκου στο αίμα και τη βελτίωση ή μη στην κατάσταση του ασθενή. Οι ερευνητές μετά στέλνουν τις πληροφορίες στους σπόνσορες οι οποίοι τις αναλύουν με στατιστικές μεθόδους διερευνώντας περαιτέρω την ασφάλεια και την αποτελεσματικότητα του φαρμάκου είτε μεμονωμένα για κάθε ασθενή είτε συγκρίνοντας μεταξύ ασθενών με την ίδια πάθηση.



Εικόνα 2.4. Στάδια προκλινικών και κλινικών δοκιμών στην εξέλιξη φαρμάκων.

2.6.2. Φάσεις κλινικών δοκιμών

Οι κλινικές δοκιμές συνήθως χωρίζονται σε 4 φάσεις και κάθε μία από αυτές αντιμετωπίζεται διαφορετικά. Η διαδικασία αυτή μπορεί να διαρκέσει πολλά χρόνια. Αν ένα φάρμακο περάσει τις πρώτες τρεις φάσεις (Εικόνα 2.4) θα πάρει έγκριση παραγωγή, ενώ η τέταρτη φάση αφορά έρευνες για το προϊόν οι οποίες λαμβάνουν χώρα μετά την παραγωγή.

- $1^{η}$ Φάση: Συνήθως επιλέγεται μια μικρή ομάδα (20-100 ατόμων) αποτελούμενη από υγιείς εθελοντές. Η φάση αυτή προσδιορίζει την ασφάλεια, την αντοχή, την φαρμακοκινητική και τη φαρμακοδυναμική του φαρμάκου. Μετά τη χορήγηση, οι εθελοντές παρακολουθούνται ανελλιπώς μέχρι να εξασθενήσει η επήρεια του φαρμάκου. Σταδιακά αυξάνεται η δοσολογία σε επίπεδα πολύ μικρότερα από αυτά που δημιουργούσαν προβλήματα κατά τη δοκιμή σε πειραματόζωα. Υπάρχουν και περιπτώσεις όπου γίνονται δοκιμές σε ασθενείς που πάσχουν από ασθένειες οι οποίες μέχρι στιγμής δεν έχουν αντιμετωπιστεί πλήρως (όπως το HIV). Ο βασικός σκοπός αυτών των δοκιμών είναι να ανακαλυφθεί η δοσολογία

μετά την οποία το φάρμακο είναι πολύ τοξικό για να χορηγηθεί [10]. Η πληρωμή για τέτοιου είδους δοκιμές μπορεί να φτάσει τα $6000 ανάλογα τη διάρκεια των δοκιμών.

- $2^{η}$ Φάση: Μετά την ολοκλήρωση της αρχικής έρευνας πρώτης φάσης, σχετικά με την ασφάλεια του φαρμάκου, αυξάνεται ο αριθμός των ασθενών (100-300 άτομα) και διερευνάται η αποτελεσματικότητα του φαρμάκου. Συνήθως η αποτυχία ενός φαρμάκου διαπιστώνεται κατά τη διάρκεια της δεύτερης φάσης όπου το φάρμακο κρίνεται αναποτελεσματικό ή τοξικό. Συνεπώς στη δεύτερη φάση γίνονται εκτιμήσεις για τη δοσολογία χορήγησης και την αποτελεσματικότητα του φαρμάκου.

- $3^{η}$ Φάση: Η τρίτη φάση αφορά τη χορήγηση του φαρμάκου σε μεγάλες ομάδες ασθενών (1.000 – 3.000+ ασθενείς) ώστε να διαπιστωθεί η αποτελεσματικότητα του φαρμάκου, και να συγκριθεί με την καλύτερη διαθέσιμη μέχρι στιγμής φαρμακευτική αγωγή. Οι δοκιμές τρίτης φάσης είναι οι πιο ακριβές, χρονοβόρες και δύσκολες δοκιμές ως προς τον σχεδιασμό και την εκτέλεση, ειδικά για θεραπείες χρόνιων ιατρικών παθήσεων. Είναι κοινή πρακτική να συνεχίζονται οι δοκιμές μέχρι η παραγωγή να εγκριθεί από τις αρχές, έτσι ώστε να μπορούν οι ασθενείς να προμηθεύονται φάρμακα τα οποία δεν υπάρχουν στην αγορά αλλά ίσως αποδειχθούν σωτήρια για αυτούς. Τα περισσότερα φάρμακα που τα καταφέρνουν στις δοκιμές τρίτης φάσης φτάνουν στην παραγωγή, αλλά σε περίπτωση δυσμενών παρενεργειών μπορεί να ανακαλεστούν από την αγορά.

- $4^{η}$ Φάση: Η τέταρτη φάση αφορά την παρακολούθηση του φαρμάκου και την ακατάπαυστη τεχνική υποστήριξη σε ένα φάρμακο από τη στιγμή που θα λάβει έγκριση για τη διάθεση του στην αγορά. Συνεπώς εντοπίζονται παρενέργειες ή άλλες επιπλοκές από μακροχρόνια χρήση του φαρμάκου, οι οποίες μπορούν εκ των υστέρων να απαγορέψουν την πώληση του ή να περιορίσουν τις εφαρμογές του.

2.6.3. Διάρκεια δοκιμών

Οι κλινικές δοκιμές αποτελούν ένα μικρό κομμάτι της έρευνας που χρειάζεται για την εξέλιξη μιας νέας θεραπείας. Για να φτάσει ένα φάρμακο στο στάδιο των κλινικών δοκιμών, πρέπει πρώτα να ανακαλυφθεί, να χαρακτηριστεί και να δοκιμαστεί σε εργαστήρια (σε κυτταρικό επίπεδο ή σε πειραματόζωα), και η όλη διαδικασία μπορεί να διαρκέσει από 6 έως 8 έτη.
Άλλος ένας παράγοντας που αυξάνει τον χρόνο των κλινικών δοκιμών είναι η έλλειψη εθελοντών ή ασθενών (Εικόνα 2.5), τόσο λόγω της ιδιαιτερότητας της ασθένειας που δοκιμάζεται, όσο επειδή πολλές φορές δεν δέχονται να λάβουν μέρος επειδή το φάρμακο δεν έχει σίγουρα αποτελέσματα. Στην περίπτωση των καρκινοπαθών, λιγότερο από 5% των ενηλίκων με καρκίνο συμφωνούν να συμμετάσχουν στις δοκιμές.

Εικόνα 2.5. Ζήτηση εθελοντών για δοκιμή φαρμάκων.

2.7. Κόστος εξέλιξης φαρμάκου

Έρευνες των diMasi et al, οι οποίες εκδόθηκαν το 2003, αναφέρουν ένα μέσο κόστος της τάξεως των 800 εκατομμυρίων δολαρίων προ φόρων μέχρι να φτάσει ένα φάρμακο την αγορά [11][12]. Μια δεύτερη έρευνα η οποία εκδόθηκε το 2006 εκτιμά ότι το προαναφερθέν κόστος κυμαίνεται μεταξύ 500 εκατομμυρίων και 2 δισεκατομμυρίων δολαρίων, το οποίο εξαρτάται άμεσα από το είδος της θεραπείας και την εταιρία που εξελίσσει το φάρμακο [13]. Πιο πρόσφατες έρευνες (2010) κάνουν λόγο για ένα μέσο κόστος το οποίο ανέρχεται στο 1.2 δισεκατομμύρια δολάρια [14].

2.8. Ρυθμός επιτυχίας

Για την αντιμετώπιση μιας ασθένειας, είναι υποψήφιες για χρήση 5000 έως 10000 χημικές ουσίες. Κατά μέσο όρο μόνο 250 από αυτές θα δώσουν καλές ενδείξεις ώστε να προχωρήσουν σε εργαστηριακές δοκιμές σε ποντίκια και άλλα πειραματόζωα. Σε γενικές γραμμές μόνο 10 από αυτά θα προκριθούν για δοκιμή σε ανθρώπους [15]. Σε μια μελέτη που έγινε από το Tufts Center for the Study of Drug Development, εκτιμήθηκε ότι μόνο το 21.5% από τα φάρμακα που έφτασαν στο στάδιο κλινικών δοκιμών, εγκρίθηκαν για παραγωγή και διάθεση στην αγορά [16].

3. Support vector machines και kernels

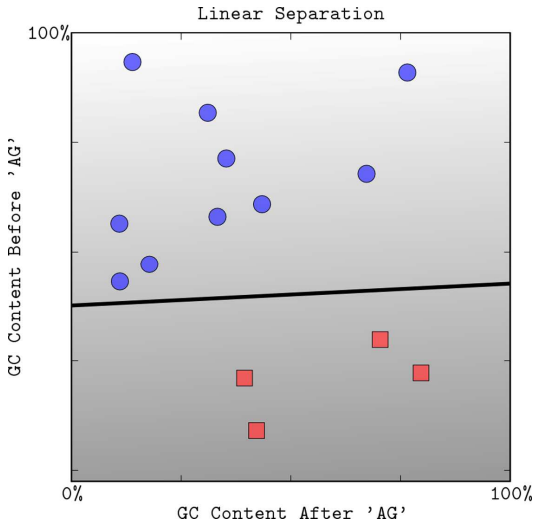
3.1. Εισαγωγή στα SVMs

Η αυξανόμενη αξία των βιολογικών δεδομένων τα οποία προκύπτουν από διάφορες εργαστηριακές διατάξεις και η συνεχής εξέλιξη νέων μεθόδων υψηλής απόδοσης για την παρακολούθηση των βιολογικών συστημάτων, απαιτούν ολοένα πιο ολοκληρωμένες υπολογιστικές προσεγγίσεις. Το πρώτο βήμα είναι να φτιαχτούν εύχρηστες βάσεις δεδομένων, αλλά η πλήρης εκμετάλλευση τους απαιτεί αλγορίθμους οι οποίοι εξάγουν αυτόματα πληροφορίες από τα δεδομένα, οι οποίες μπορούν να δώσουν επιθυμητά αποτελέσματα σε βιολογικό επίπεδο.

Πολλά από τα προβλήματα που αντιμετωπίζονται από την υπολογιστική βιολογία αφορούν προβλέψεις για τη δομή ενός γονιδίου, τη λειτουργία του, τις αλληλεπιδράσεις και το ρόλο του σε μια ασθένεια. Τα SVMs (Support Vector Machines) και οι σχετικές μέθοδοι στο επίπεδο των kernels, είναι πολύ ικανά στην επίλυση τέτοιου είδους προβλημάτων [17]-[19]. Τα SVMs χρησιμοποιούνται συχνά στην υπολογιστική βιολογία λόγω της υψηλής τους ακρίβειας, της ικανότητας τους να χειρίζονται πολυδιάστατες και μεγάλες βάσεις δεδομένων, και της ευελιξίας να μοντελοποιήσουν ποικίλες πηγές δεδομένων [18],[20]-[22].

3.2. Δυαδική κατηγοριοποίηση (Binary Classification)

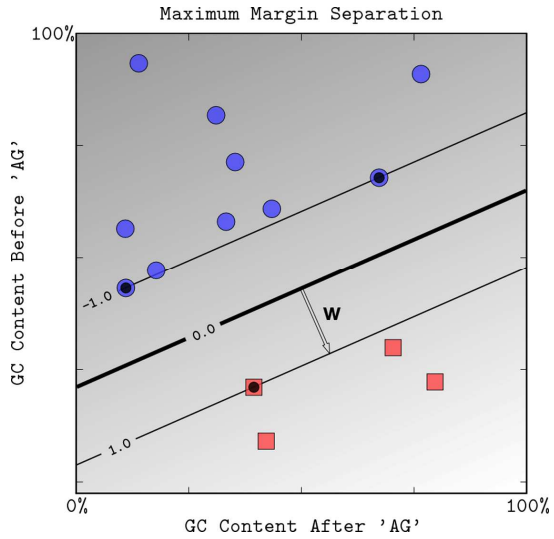
Η απλούστερη μορφή επίλυσης ενός προβλήματος πρόβλεψης είναι η δυαδική κατηγοριοποίηση (binary classification), όπου πρέπει να γίνει ένας διαχωρισμός σε αντικείμενα που ανήκουν σε μία από δύο κατηγορίες οι οποίες συμβολίζονται με έναν άσσο με θετικό (+1) ή αρνητικό (-1) πρόσημο. Τα SVMs χρησιμοποιούν δύο βασικές ιδέες για την επίλυση αυτού του προβλήματος: διαχωρισμός δεδομένων με μεγάλο περιθώριο (large margin separation) και πράξεις στο επίπεδο των kernels (kernel functions). Η ιδέα του large margin separation μπορεί να αναπαρασταθεί εύκολα όταν διαχωρίζονται σημεία σε δύο διαστάσεις (Εικόνα 3.1). Ένας απλός τρόπος να διαχωριστούν αυτά τα σημεία είναι να σχεδιάσουμε μια ίσια γραμμή που τα χωρίζει, και να αποκαλέσουμε τα σημεία που βρίσκονται στη μία πλευρά θετικά, και αυτά που βρίσκονται στην άλλη πλευρά αρνητικά. Αν τα δύο σύνολα χωριστούν σωστά, γίνεται προσπάθεια να σχεδιαστεί ξανά η γραμμή, αλλά αυτή τη φορά όσο πιο μακριά γίνεται από τα σημεία και των δύο συνόλων (Εικόνες 3.2 και 3.3). Αυτή η επιλογή σχεδίασης αποτελεί την ιδέα του διαχωρισμού με το μεγαλύτερο δυνατό περιθώριο (large margin separation).



Εικόνα 3.1. Ένας γραμμικός κατηγοριοποιητής ο οποίος διαχωρίζει δύο κατηγορίες σημείων (τετράγωνα και κύκλος), σχεδιασμένα σε δύο διαστάσεις [30].

Η διαχωριστική γραμμή χωρίζει τον χώρο σε δύο σύνολα ανάλογα με το πρόσημο. Η

απόχρωση του γκρι αναπαριστά την τιμή της διακρίνουσας εξίσωσης: σκούρο για χαμηλές τιμές και ανοιχτό για υψηλές.

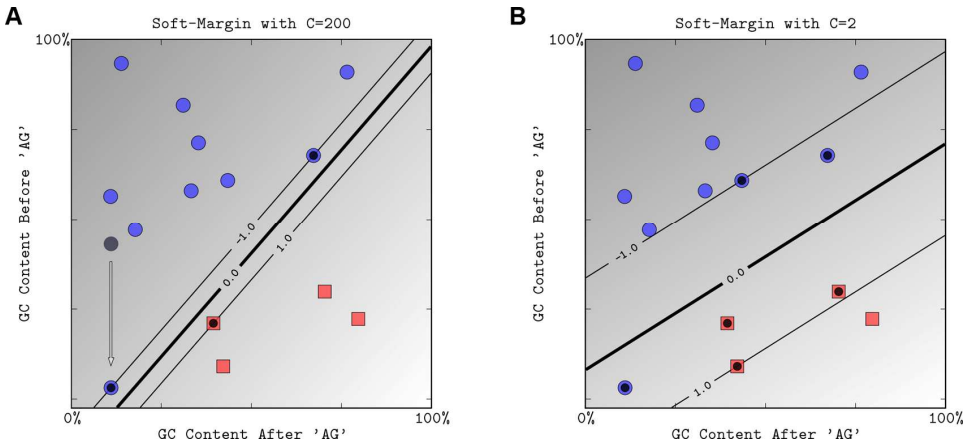


Εικόνα 3.2. Το μέγιστο περιθώριο όπως υπολογίζεται από ένα γραμμικό SVM [30].

Η περιοχή ανάμεσα στις δύο λεπτές γραμμές ορίζει την περιοχή του περιθωρίου (margin area) όπου,

$$-1 \leq \langle \mathbf{w}, \mathbf{x} \rangle + b \leq 1.$$

Τα σημεία των δεδομένων που έχουν μαύρα κέντρα είναι τα support vectors, δηλαδή τα σημεία που είναι κοντά στο σύνορο απόφασης (decision boundary). Αυτά ορίζουν το περιθώριο το οποίο χωρίζει τις δύο κατηγορίες. Στο παραπάνω σχήμα φαίνονται τρία support vectors πάνω στα σύνορα (όπου $f(\mathbf{x}) = -1$ or $f(\mathbf{x}) = +1$).

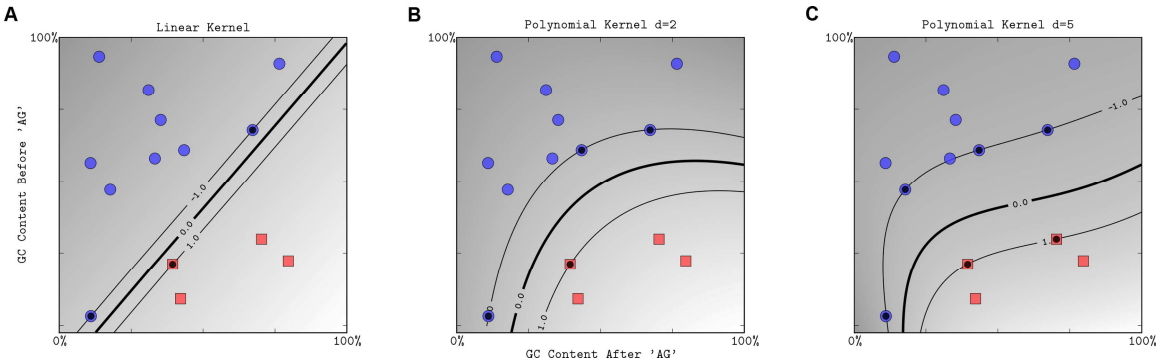


Εικόνα 3.3. Η επιρροή της σταθεράς του soft-margin, C, στο decision boundary [30].

Αν τροποποιήσουμε τα δεδομένα μετακινώντας το σημείο με γκρι σκίαση στη νέα θέση που δείχνει το τόξο, αυτό θα μειώσει σημαντικά το περιθώριο με το οποίο τα hard-margin SVM μπορούν να κατηγοριοποιήσουν τα δεδομένα. Η δημιουργία του περιθωρίου χρησιμοποιώντας μια πολύ μεγάλη τιμή του C, μιμείται τη συμπεριφορά του hard margin SVM, το οποίο μας δείχνει ότι κάποια σημεία που μπορεί να αποτελούν λάθη στην εκπαίδευση έχουν μεγάλο κόστος στο σχεδιασμό του περιθωρίου (Εικόνα 3.3.A). Μία μικρότερη τιμή του C μας επιτρέπει να αγνοήσουμε κάποια σημεία πολύ κοντά στο περιθώριο και έτσι έχει την ευχέρεια να μεγαλώσει το σύνορο (Εικόνα 3.3.B). Το περιθώριο επιλογής ανάμεσα στα θετικά και τα αρνητικά σημεία είναι η παχιά γραμμή που φαίνεται στο σχήμα, ενώ οι λεπτές γραμμές αντιπροσωπεύουν το σύνορο (-1 έως +1).

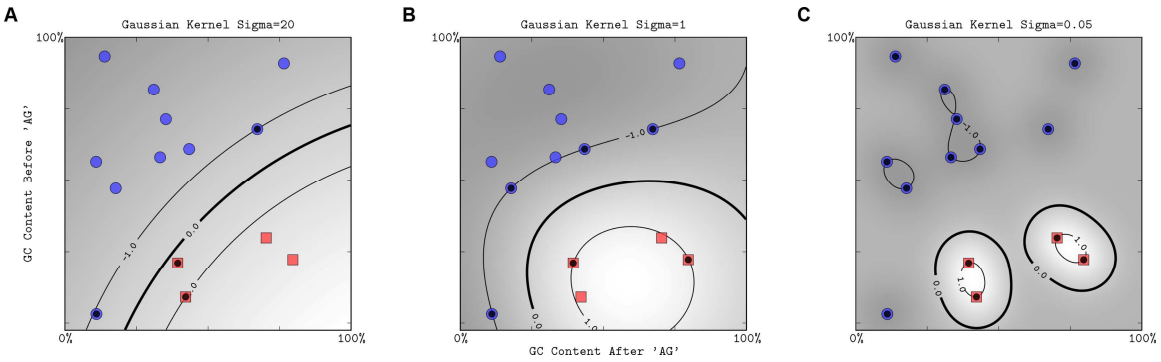
Αντί της ιδέας των σημείων στο χώρο, μπορούμε να σκεφτούμε ότι τα σημεία μας αναπαριστούν αντικείμενα, τα οποία αντιπροσωπεύονται από ένα σύνολο ιδιοτήτων το οποίο προήλθε από μετρήσεις σε κάθε ένα από αυτά. Για παράδειγμα στην περίπτωση των σχημάτων (Εικόνες 3.1 – 3.5) υπάρχουν δύο μετρήσεις για κάθε αντικείμενο και συμβολίζονται σημειακά σε ένα επίπεδο δύο διαστάσεων. Όταν χρησιμοποιήθηκε μεγάλο σύνορο, αποδείχτηκε ότι η σχετική θέση των σημείων ή η ομοιότητα των σημείων είναι πιο σημαντική από την ακριβή θέση τους. Στην απλούστερη περίπτωση της γραμμικής κατηγοριοποίησης, η ομοιότητα δύο αντικειμένων υπολογίζεται από το εσωτερικό γινόμενο των αντίστοιχων διανυσμάτων (vectors). Για να οριστούν άλλα μέτρα ομοιότητας που οδηγούν σε μη γραμμική κατηγοριοποίηση, μπορεί να επεκταθεί η ιδέα

του εσωτερικού γινομένου ανάμεσα στα σημεία με τη βοήθεια πράξεων σε επίπεδο kernel. Τα kernel υπολογίζουν την ομοιότητα ανάμεσα σε δύο σημεία και είναι η δεύτερη σημαντική ιδέα των SVMS και των πράξεων σε επίπεδο kernel [18],[23].



Εικόνα 3.4. Η επίδραση του βαθμού ενός πολυωνυμικού kernel [30].

Ένα πολυωνυμικό kernel με βαθμό 1 οδηγεί σε γραμμικό διαχωρισμό (Εικόνα 3.4.A). Πολυωνυμικοί kernel υψηλότερου βαθμού επιτρέπουν πιο ευέλικτο σύνορο διαχωρισμού (Εικόνα 3.4.B,C).



Εικόνα 3.5. Η επιρροή της παραμέτρου πλάτους του Gaussian kernel (σ) για μια σταθερή τιμή του C [30].

Για μεγαλύτερες τιμές του σ (Εικόνα 3.5.A), το σύνορο διαχωρισμού είναι σχεδόν γραμμικό. Όσο μικραίνει το «σ», η ευελιξία του συνόρου αυξάνεται (Εικόνα 3.5.B). Μικρές τιμές του «σ» οδηγούν σε over fitting (Εικόνα 3.5.C).

Η εγγενή γνώση του τομέα σε κάθε διαδικασία κατηγοριοποίησης, συλλαμβάνεται ορίζοντας ένα κατάλληλο kernel μεταξύ των αντικειμένων. Όπως θα δούμε αργότερα, αυτό έχει δύο πλεονεκτήματα: την ικανότητα να παράγει μη-γραμμικά σύνορα διαχωρισμού χρησιμοποιώντας μεθόδους για σχεδιασμένες για μη γραμμικούς κατηγοριοποιητές, και την πιθανότητα να εφαρμοστεί ένας κατηγοριοποιητής σε δεδομένα χωρίς προφανές διανυσματικό διάστημα.

3.3. Large Margin Separation

3.3.1. Γραμμικός διαχωρισμός με hyperplanes.

Σε αυτό το κομμάτι, παρουσιάζουμε την ιδέα των γραμμικών κατηγοριοποιητών. Τα SVM είναι ένα παράδειγμα ενός γραμμικού κατηγοριοποιητή δύο κλάσεων. Τα δεδομένα για ένα πρόβλημα εκμάθησης περιλαμβάνουν αντικείμενα τα οποία έχουν ένα από τα δύο LABEL: +1 (θετικά αντικείμενα) και -1 (αρνητικά αντικείμενα). Έστω το **x** αντιπροσωπεύει ένα διάνυσμα με M στοιχεία, όπου το x_j, $j = 1,\ldots,M$ αναπαριστά ένα σημείο σε ένα διανυσματικό πεδίο M διαστάσεων. Το $\mathbf{x}_i$ αναπαριστά το *i-οστό διάνυσμα των δεδομένων* $\{(\mathbf{x}_i,y_i)\}_{i=1}^{n}$, ενώ το y_i είναι η ετικέτα (label) το οποίο σχετίζεται με το αντικείμενο $\mathbf{x}_i$, και n ο αριθμός των αντικειμένων. Τα αντικείμενα $\mathbf{x}_i$ ονομάζονται πρότυπα, είσοδοι ή παραδείγματα.

Απαραίτητο για τον ορισμό ενός γραμμικού κατηγοριοποιητή (linear classifier) είναι το εσωτερικό γινόμενο ανάμεσα στα δύο διανύσματα:

$$\langle \mathbf{w},\mathbf{x} \rangle = \sum_{j=1}^{M} w_j x_j .$$

Ένας γραμμικός κατηγοριοποιητής βασίζεται σε μία γραμμική διακρίνουσα εξίσωση της μορφής:

$$f(\mathbf{x}) = \langle \mathbf{w},\mathbf{x} \rangle + b \quad (1)$$

Η διακρίνουσα εξίσωση $f(\mathbf{x})$ ορίζει ένα βαθμό για την είσοδο **x**, και χρησιμοποιείται για να ληφθεί η απόφαση ως προς τον τρόπο κατηγοριοποίησης. Το διάνυσμα **w** είναι γνωστό ως weight factor, και ο βαθμωτός *b* ονομάζεται *bias*. Σε πρόβλημα δύο διαστάσεων, τα σημεία που ικανοποιούν την εξίσωση:

$$\langle \mathbf{w},\mathbf{x} \rangle = 0 ,$$

αντιστοιχούν σε μια γραμμή η οποία περνά από την αρχή των αξόνων, σε ένα επίπεδο τριών διαστάσεων, και πιο γενικά ένα υπέρ-επίπεδο (hyperplane). Το bias μετατοπίζει το hyperplane σε σχέση με την αρχή των αξόνων.

Το hyperplane διαιρεί το χώρο σε δύο, και σύμφωνα με το πρόσημο της $f(\mathbf{x})$, γνωρίζουμε σε ποια πλευρά του hyperplane θα βρίσκεται κάθε σημείο(σχήμα 1): Αν η $f(\mathbf{x})>0$, τότε το σημείο βρίσκεται στη θετική πλευρά. Το σύνορο ανάμεσα στις δύο περιοχές οι οποίες έχουν διαχωριστεί ως θετικές ή αρνητικές ονομάζεται σύνορο απόφασης (decision boundary) του κατηγοριοποιητή. Το σύνορο απόφασης ορισμένο από ένα υπέρ-επίπεδο

θεωρείται γραμμικό, επειδή είναι γραμμικό στα δεδομένα. Ένας κατηγοριοποιητής με γραμμικό σύνορο απόφασης ονομάζεται γραμμικός κατηγοριοποιητής. Στο επόμενο κομμάτι, θα συζητηθεί ένας συγκεκριμένος γραμμικός κατηγοριοποιητής, το SVM, ο οποίος είναι από τους πιο αποτελεσματικούς όταν δουλεύει με μεγάλα σύνολα δεδομένων.

3.3.2. Κατηγοριοποίηση με μεγάλο περιθώριο (Classification with large margin)

Όταν έχουμε να δουλέψουμε με ένα σύνολο δεδομένων το οποίο μπορεί να διαχωριστεί γραμμικά, όπως αυτό του σχήματος 1, δεν υπάρχει μόνο ένα hyperplane το οποίο κατηγοριοποιεί σωστά όλα τα σημεία. Εδώ προκύπτει ο προβληματισμός επιλογής ενός μόνο hyperplane, το οποίο όχι μόνο χωρίζει τα σημεία σωστά, αλλά το κάνει με μεγάλο περιθώριο. Το περιθώριο ενός γραμμικού κατηγοριοποιητή, ορίζεται ως η απόσταση του κοντινότερου σημείο στο σύνορο απόφασης (Εικόνα 3.2).

Το hard-margin SVM, μπορεί να εφαρμοστεί σε όλα τα δεδομένα που χωρίζονται γραμμικά, και η ιδιότητα του είναι να κατηγοριοποιήσει σωστά όλα τα σημεία. Αυτή η ιδιότητα δίνει μεγάλη ακρίβεια στον CLASSIFIER, αλλά έχει χαμηλή απόδοση σε σχέση με τα SOFT MARGIN SVM.

3.4. Soft margin

Πρακτικά, τα δεδομένα δεν γίνεται πάντα να διαχωριστούν γραμμικά, και ακόμα και αν γίνεται μπορεί να επιτευχθεί μεγαλύτερο περιθώριο (margin) αν εσκεμμένα κατηγοριοποιηθούν λάθος κάποια σημεία (Εικόνα 3.3). Η θεωρία και τα αποτελέσματα έρευνας έχουν δείξει ότι το μεγάλο περιθώριο θα αποδώσει καλύτερα από το hard margin SVM. Για να επιτραπούν σφάλματα στην κατηγοριοποίηση, μετατρέπουμε την εξίσωση ως εξής:

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \text{ for } i = 1, \ldots, n,$$

Όπου $\xi_i \geq 0$ είναι οι μεταβλητές που επιτρέπουν σε ένα σημείο να τοποθετηθεί μέσα στο σύνορο ή σε λάθος κατηγορία.

Για να αποτραπεί η υπερβολική χρήση των λάθος τοποθετημένων σημείων, ορίζεται η

σταθερά C η οποία θέτει τους όρους για τη μεγιστοποίηση του περιθωρίου και την ελαχιστοποίηση των λάθος κατηγοριοποιήσεων. Η μέθοδος αυτή ονομάζεται soft-margin SVM [24].

Η επιρροή της μεταβλητής C φαίνεται στην εικόνα 3.3. Για μεγαλύτερες τιμές του C, δίνεται βάρος στην αποφυγή των λανθασμένος κατηγοριοποιήσεων, όπως φαίνεται στο σχήμα 3Α, όπου τα δύο σημεία που βρίσκονται κοντά στο hyperplane επηρεάζουν άμεσα τον προσανατολισμό του και το φέρνουν πολύ κοντά στα υπόλοιπα σημεία. Όταν η τιμή του C μειωθεί (Εικόνα 3.3Β), τα σημεία αυτά βρίσκονται μέσα στο περιθώριο, και για τα υπόλοιπα δεδομένα, το περιθώριο αυτό είναι πολύ μεγαλύτερο.

3.5. Κανονικοποίηση δεδομένων (Normalization)

Οι κατηγοριοποιητές οι οποίοι λειτουργούν με μεγάλο περιθώριο, είναι ευαίσθητοι ως προς τον τρόπο εισαγωγής των δεδομένων [25]. Για αυτό το λόγο είναι πολλές φορές απαραίτητο να γίνει κανονικοποίηση των δεδομένων. Η κανονικοποίηση μπορεί να πραγματοποιηθεί είτε σε επίπεδο δεδομένων ή σε επίπεδο kernel, ή και στα δύο. Όταν μετρώνται δεδομένα σε διαφορετικές κλίμακες, γίνεται προσπάθεια να καθοριστούν οι τιμές έτσι ώστε να είναι της ίδιας κλίμακας, π.χ. για κάθε τιμή αφαιρούμε τη μέση τιμή και τη διαιρούμε με την τυπική απόκλιση. Μία εναλλακτική του να κανονικοποιήσουμε κάθε αντικείμενο ξεχωριστά, είναι να μετατρέψουμε τα αντικείμενα σε μοναδιαία διανύσματα. Σε γενικές γραμμές η κανονικοποίηση συχνά προσφέρει βελτιωμένη απόδοση σε γραμμικούς και μη γραμμικούς Kernels, και μπορεί επίσης να οδηγήσει σε ταχύτερη σύγκλιση.

3.6. Χειρισμός μη ισορροπημένου αριθμού δεδομένων

Πολλά σύνολα δεδομένων που συναντάμε δεν είναι ισορροπημένα, δηλαδή η μία κατηγορία αποτελείται από πολύ λιγότερα αντικείμενα σε σχέση με την άλλη. Τέτοια σύνολα δεδομένων δυσκολεύουν πολύ τη δουλειά του κατηγοριοποιητή. Όταν ένα σύνολο δεδομένων δεν είναι ισορροπημένο, το κόστος της λάθος κατηγοριοποίησης δεν είναι και αυτό ισορροπημένο, μιας και ένα λάθος στην κατηγορία με τις λιγότερες τιμές θα κοστίσει πολύ περισσότερο από ένα λάθος στην άλλη κατηγορία. Για την αντιμετώπιση αυτού του προβλήματος δίνονται διαφορετικές παράμετροι για κάθε κατηγορία, ώστε να μην επιτρέπει να γίνονται εύκολα λάθη στην κατηγορία με τις λιγότερες τιμές [26].

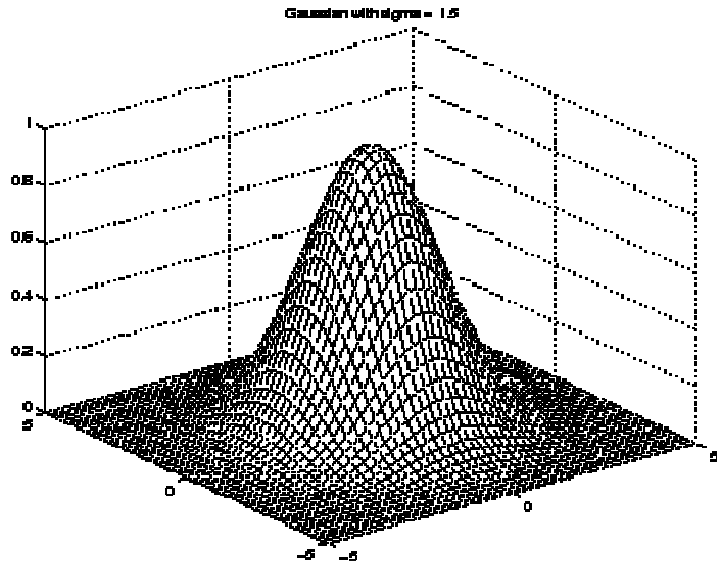
3.7. Επιλογή Kernel

Ο αλγόριθμος των SVM επιτρέπει τη χρήση kernels, δίνοντας έτσι τη δυνατότητα να υπολογιστούν εσωτερικά γινόμενα σε μη γραμμικό χώρο.

Δεν υπάρχει γενικός κανόνας για την επιλογή kernel, μιας και όλα εξαρτώνται από τα δεδομένα. Δοκιμάζονται πρώτα ένας γραμμικός kernel και μετά εξετάζεται αν ένας πολυωνυμικός kernel μπορεί να βελτιώσει την απόδοση. Μετά δοκιμάζονται διαφορετικές τιμές για τις παραμέτρους του κάθε kernel.

Τα kernels που χρησιμοποιούνται πιο συχνά είναι τα εξής:

1. Linear
2. Polynomial
3. Gaussian Radial Basis Function (Εικόνα 3.6)



Εικόνα 3.6. Γραφική αναπαράσταση ενός radial basis function kernel.

Πολλές φορές δεν δοκιμάζεται μόνο ένας kernel, αλλά συνδυασμός διαφορετικών kernels, με σκοπό να αυξηθεί η απόδοση του κατηγοριοποιητή [27]-[29].

3.8. Μέθοδοι για Cross-validation

Το cross-validation είναι μια τεχνική η οποία ελέγχει τον τρόπο που θα χρησιμοποιηθούν τα δεδομένα σε μια στατιστική ανάλυση. Χρησιμοποιείται σε περιπτώσεις όπου στόχος είναι η πρόβλεψη, και επιχειρείται να βρεθεί πόσο ακριβείς είναι τα αποτελέσματα. Έτσι ένα σύνολο δεδομένων χωρίζεται σε δύο ή περισσότερα υποσύνολα, χρησιμοποιώντας το ένα υποσύνολο για την εκπαίδευση του αλγορίθμου (training dataset) και το άλλο για αξιολόγηση του(testing dataset). Για πιο ασφαλή αποτελέσματα, γίνονται πολλές επαναλήψεις του cross-validation χρησιμοποιώντας διαφορετικά υποσύνολα κάθε φορά, και βγαίνει ο μέσος όρος από όλες τις επαναλήψεις.

Οι πιο κοινές μέθοδοι για cross-validation είναι οι εξής:

- ***K-fold cross-validation:*** Σε αυτή τη μέθοδο το αρχικό σύνολο δεδομένων χωρίζεται σε k υποσύνολα. Από τα k υποσύνολα ένα χρησιμοποιείται για επαλήθευση (test dataset) και τα υπόλοιπα (k-1) υποσύνολα χρησιμοποιούνται για την εκπαίδευση (training data). Η διαδικασία αυτή επαναλαμβάνεται k φορές (όσες και τα folds), όπου κάθε ένα από τα k υποσύνολα χρησιμοποιείται μία φορά σαν δεδομένα επαλήθευσης. Το μεγαλύτερο πλεονέκτημα αυτής της μεθόδου είναι ότι όλα τα αντικείμενα του συνόλου δεδομένων χρησιμοποιούνται και για εκπαίδευση αλλά και για επαλήθευση.
- ***2-fold cross-validation:*** Αυτή είναι η απλούστερη μορφή του k-fold cross-validation και όπως λέει ο τίτλος το σύνολο δεδομένων χωρίζεται σε δύο υποσύνολα. Κάθε fold λαμβάνει αντικείμενα με τυχαίο τρόπο ώστε και τα δύο folds να έχουν ίδιο αριθμό αντικειμένων. Έπειτα το ένα από τα δύο χρησιμοποιείται για εκπαίδευση και το άλλο για επαλήθευση.
- ***Leave-one-out cross-validation:*** Όπως δηλώνει και το όνομα η μέθοδος leave-one-out cross-validation (LOOCV), χρησιμοποιεί ένα μόνο αντικείμενο από το αρχικό σύνολο δεδομένων για επαλήθευση και όλα τα υπόλοιπα αντικείμενα χρησιμοποιούνται για εκπαίδευση. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να χρησιμοποιηθούν όλα τα αντικείμενα από μία τουλάχιστον φορά για επαλήθευση. Η διαδικασία είναι ίδια με αυτή του K-fold cross-validation, απλά στην προκειμένη περίπτωση ο αριθμός K (folds) είναι ίσος με τον αριθμό των αντικειμένων. Η μέθοδος leave-one-out συνήθως δίνει τα καλύτερα αποτελέσματα, αλλά έχει μεγάλο κόστος σε υπολογιστική ισχύ λόγω των πολλών επαναλήψεων που απαιτούνται για την ολοκλήρωση της εκπαίδευσης.

Για την εκπόνηση της διπλωματικής εργασίας χρησιμοποιήθηκε η μέθοδος Leave-one-out cross-validation.

3.9. Sensitivity και specificity

Τα sensitivity (ευαισθησία) και specificity (ειδικότητα) είναι στατιστικές μετρήσεις της απόδοσης μιας μεθόδου δυαδικής κατηγοριοποίησης.

Τα αποτελέσματα που λαμβάνουμε χωρίζονται σε τέσσερις κατηγορίες:

1. True positives: Θετικοί (+1) οι οποίοι επαληθεύτηκαν σωστά.
2. False positives: Αρνητικοί (-1) οι οποίοι επαληθεύτηκαν λανθασμένα ως θετικοί.
3. True negatives: Αρνητικοί (-1) οι οποίοι επαληθεύτηκαν σωστά.
4. False negatives: Θετικοί (+1) οι οποίοι επαληθεύτηκαν λανθασμένα ως αρνητικοί.

3.9.1. Sensitivity

Ο όρος sensitivity σχετίζεται με την ικανότητα της δοκιμής να αναγνωρίσει θετικά αποτελέσματα. Υπολογίζεται εύκολα με την παρακάτω εξίσωση:

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

3.9.2. Specificity

Ο όρος specificity ορίζεται ως εξής:

$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}}$$

3.10. Αλγόριθμοι εκπαίδευσης SVM και software.

Η καταλληλότητα των SVMs έχει οδηγήσει στην εξέλιξη μεγάλου αριθμού λύσεων για τη βελτιστοποίηση των SVMs. Υπάρχουν πολλές διαφορετικές υλοποιήσεις των SVMs αλλά τα πιο συχνά χρησιμοποιημένα toolbox για περιπτώσεις δυαδικής κατηγοριοποίησης είναι τα LIBSVM και SVMlight.

Για την εκπόνηση της εργασίας χρησιμοποιήθηκε το LIBSVM toolbox.

4. Ανάλυση δεδομένων και εξαγωγή αποτελεσμάτων

4.1. Εισαγωγή

Στο κεφάλαιο αυτό θα αναλύσουμε τα περιεχόμενα και τη δομή των συνόλων δεδομένων, τον τρόπο χειρισμού τους, την εξαγωγή, την ανάλυση και την επεξήγηση των αποτελεσμάτων του κώδικα. Για την εκπόνηση της εργασίας και την εξαγωγή αποτελεσμάτων χρησιμοποιήθηκε περιβάλλον MATLAB σε συνδυασμό με το LIBSVM toolbox (κώδικας στο Παράρτημα).

4.2. Περιεχόμενα των συνόλων δεδομένων

Τα σύνολα δεδομένων που χρησιμοποιήσαμε αποτελούνται από μετρήσεις 13 σημάτων, σε 4 διαφορετικές κυτταροσειρές. Τα σήματα που χρησιμοποιήσαμε για την εκπαίδευση των SVM, προέρχονται από δοκιμή 7 διαφορετικών φαρμάκων, το αποτέλεσμα των οποίων είναι γνωστό.

Οι τέσσερις κυτταροσειρές από τις οποίες έχουμε λάβει μετρήσεις είναι οι εξής:

- Human hepatoma Hep3B cell line
- Human hepatoma Hepg2 cell line
- Human hepatoma Huh7 cell line
- Mhc cell line

Σε αυτές τις κυτταροσειρές χορηγήθηκαν τα εξής φάρμακα τα αποτελέσματα των οποίων είναι γνωστά, και φαίνονται παρακάτω:

- Sunitinib: pass (label: +1)
- Sorafenib: pass (label: +1)
- Lapatinib: fail (label: -1)
- Gefitinib: fail (label: -1)
- Erlotenib: pass (label:+1)
- Bortezomib: fail (label:-1)
- DMSO: fail (label: -1)

Μετά τη χορήγηση των φαρμάκων, τα σήματα που μετρήθηκαν στα κύτταρα είναι τα εξής:

- AKT
- CREB
- ERK12
- HSP27
- ikb
- IRB
- IRS1
- JNK
- MEK1
- P38
- P70S6
- cmet
- igfr

4.3. Χειρισμός των dataset από τον κώδικα

Τα dataset χρησιμοποιήθηκαν με δύο διαφορετικούς τρόπους:

- ***Integrated dataset:*** περιέχει δεδομένα από τη δοκιμή φαρμάκων σε όλες τις κυτταροσειρές, και τα folds διαιρέθηκαν ανά φάρμακο. Τα folds χωρίστηκαν ανά φάρμακο, άρα κάθε fold περιέχει 4 γραμμές με δεδομένα, με το ίδιο φάρμακο αλλά διαφορετική κυτταροσειρά σε κάθε γραμμή των δεδομένων. Συνεπώς έχουμε 7 διαφορετικά folds. Σε κάθε τρέξιμο χρησιμοποιήσαμε 6 folds για εκπαίδευση(train dataset) και ένα για επαλήθευση(test dataset), όπου μετά από 7 loops του κώδικα όλα τα fold χρησιμοποιήθηκαν ακριβώς μία φορά για επαλήθευση.
- ***Ξεχωριστά datasets για κάθε κυτταροσειρά.*** Με αυτά τα dataset κάθε κυτταροσειρά εξετάστηκε μόνη της. Κάθε μία από τις 7 γραμμές του dataset αντιπροσωπεύει μετρήσεις από κάθε φάρμακο. Όπως και στο integrated dataset έχουμε 7 folds, με τη διαφορά ότι κάθε fold περιέχει μία μόνο γραμμή και αντιπροσωπεύει τις μετρήσεις από τη χορήγηση ενός φαρμάκου στην εξεταζόμενη κυτταροσειρά.

Κατά την εκτέλεση ο κώδικας έχει την ιδιότητα να δοκιμάζει χρησιμοποιώντας όλα τα σήματα του συνόλου δεδομένων αλλά και μερικό σύνολο αυτών. Επαναληπτικά αφαιρείται κάθε φορά μία στήλη, η οποία περιέχει τις μετρήσεις ενός σήματος, και μετά προκύπτουν δύο διαφορετικά σύνολα δεδομένων. Το ένα περιέχει μόνο τη στήλη(one-only, Πίνακες 2-6) που αφαιρέθηκε και το άλλο όλες τις υπόλοιπες στήλες (one-out, Πίνακες 7-11). Αυτό γίνεται για τους εξής λόγους:

- Για να δούμε πόσο σωστά γίνεται και αν επηρεάζεται η επαλήθευση όταν ένα σήμα αφαιρεθεί από τα δεδομένα
- Για να δούμε πόσο σωστά μπορεί να επαληθεύσει το μοντέλο με ένα μόνο σήμα.

4.4. Επιλογή dataset και εξαγωγή αποτελεσμάτων

Για να επιδιώξουμε τα καλύτερα δυνατά αποτελέσματα χρησιμοποιώντας τα παραπάνω δεδομένα, έλαβα 5 datasets, τα οποία προέκυψαν από διαφορετικό τρόπο κανονικοποίησης του αρχικού dataset, και στη συνέχεια δοκιμάστηκαν και συγκρίθηκαν ώστε να διαλέξουμε ένα με το οποίο θα συνεχίσουμε τις δοκιμές (Πίνακας 4.1)

Τα datasets είναι τα εξής:

- Bool_0_45
- Bool_0_60
- Relmax
- Relmax_diff
- Relmax_collapse_timepoints

Τα 5 datasets συγκρίθηκαν μεταξύ τους και αυτό που επιλέχθηκε για τη συνέχεια των δοκιμών είναι το "**bool_0_60**", ύστερα από συνεννόηση με τον επιβλέποντα καθηγητή.

Στους πίνακες των αποτελεσμάτων εμφανίζεται στις σειρές η κυτταροσειρά μαζί με το φάρμακο που έχει χορηγηθεί, ενώ στις στήλες εμφανίζονται το σήμα ή τα σήματα για τα οποία γίνεται η κατηγοριοποίηση.

4.5. Παρουσίαση πινάκων αποτελεσμάτων και περιεχόμενα

Ο χρωματικός διαχωρισμός στους πίνακες δείχνει τα αποτελέσματα της επαλήθευσης:

- Πράσινο: Σωστή επαλήθευση (true negative ή true positive)
- Κόκκινο: λανθασμένη επαλήθευση (false negative ή false positive)

Κάτω από κάθε πίνακα αναρτώνται χρήσιμα δεδομένα σχετικά με το kernel που χρησιμοποιήθηκε, την ακρίβεια, το sensitivity και το specificity της κατηγοριοποίησης.

Η διευκρίνιση των παραμέτρων είναι της μορφής “t/d/g” όπου:

- t: ο τύπος kernel που χρησιμοποιήθηκε (0: linear, 1: polynomial, 2: Gaussian rbf)
- d: Ο βαθμός του πολυωνύμου
- g: το gamma variation

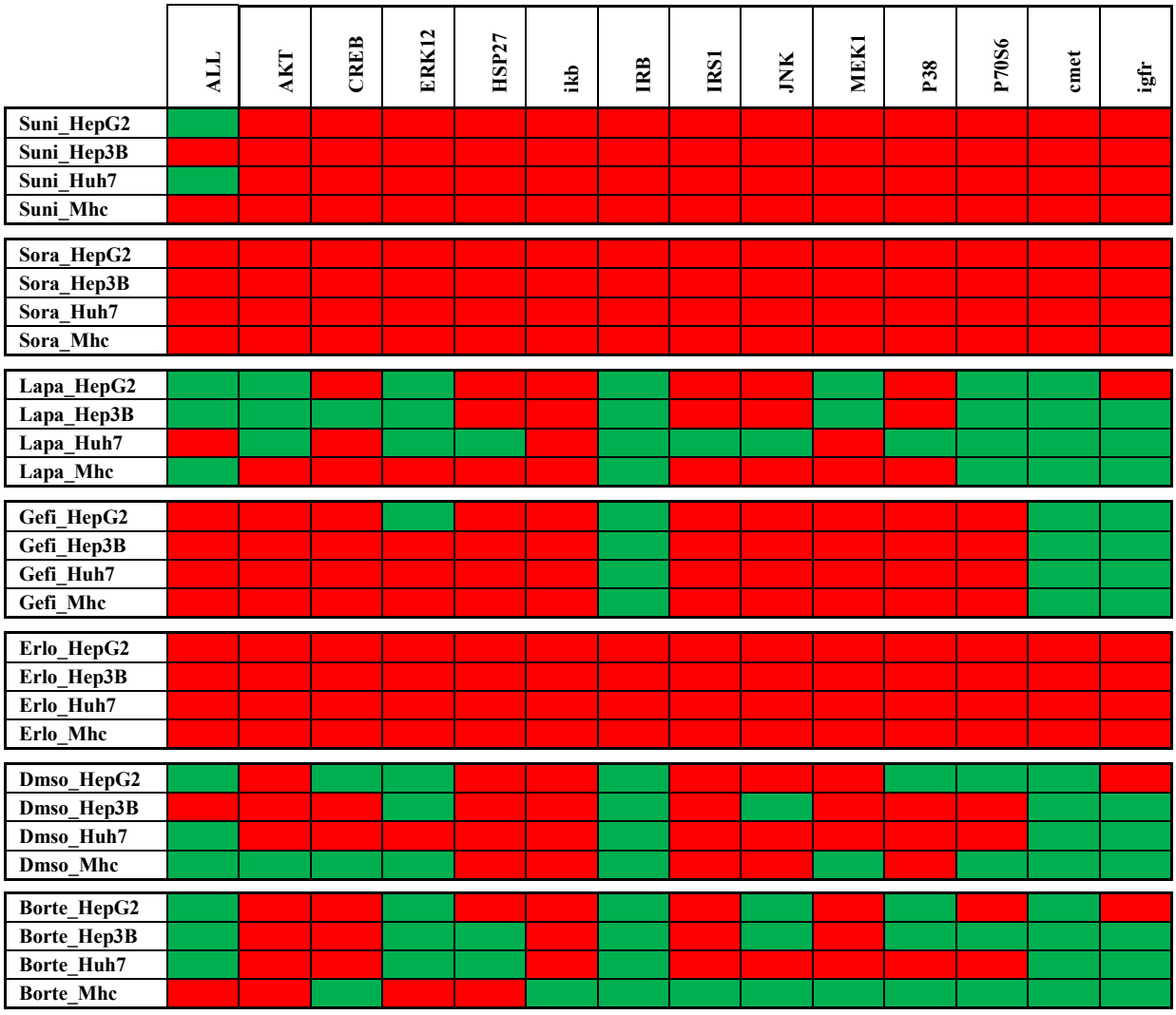
Οι πίνακες που θα συναντήσουμε παρακάτω είναι οι εξής:

- Πίνακας 1: Σύγκριση των 5 διαφορετικών dataset
- Πίνακες 2 - 6: Επαλήθευση δεδομένων σε integrated dataset αλλά και κάθε κυτταροσειρά μόνη της με τη μέθοδο one-only.
- Πίνακες 7 - 11: Επαλήθευση δεδομένων σε integrated dataset αλλά και κάθε κυτταροσειρά μόνη της με τη μέθοδο one-only.

4.6. Πίνακες αποτελεσμάτων εκπαίδευσης

	bool_0_45	bool_0_60	relmax	relmax_diff_1	relmax_col_t
Suni_HepG2					
Suni_Hep3B					
Suni_Huh7					
Suni_Mhc					
Sora_HepG2					
Sora_Hep3B					
Sora_Huh7					
Sora_Mhc					
Lapa_HepG2					
Lapa_Hep3B					
Lapa_Huh7					
Lapa_Mhc					
Gefi_HepG2					
Gefi_Hep3B					
Gefi_Huh7					
Gefi_Mhc					
Erlo_HepG2					
Erlo_Hep3B					
Erlo_Huh7					
Erlo_Mhc					
Dmso_HepG2					
Dmso_Hep3B					
Dmso_Huh7					
Dmso_Mhc					
Borte_HepG2					
Borte_Hep3B					
Borte_Huh7					
Borte_Mhc					
kernel	0	0	0	1/2/0,1	1/2/0,1
accuracy	0.321428571	0.392857143	0.357142857	0.571428571	0.571428571
sensitivity	0.4375	0.5625	0.5	1	1
specificity	0.166666667	0.166666667	0.166666667	0	0

Πίνακας 4.1: Σύγκριση των 5 διαφορετικών integrated datasets και επιλογή ενός από αυτά για χρήση στον κώδικα.



	ALL	AKT	CREB	ERK12	HSP27	ikb	IRB	IRS1	JNK	MEK1	P38	P70S6	cmet	igfr
Suni_HepG2														
Suni_Hep3B														
Suni_Huh7														
Suni_Mhc														
Sora_HepG2														
Sora_Hep3B														
Sora_Huh7														
Sora_Mhc														
Lapa_HepG2														
Lapa_Hep3B														
Lapa_Huh7														
Lapa_Mhc														
Gefi_HepG2														
Gefi_Hep3B														
Gefi_Huh7														
Gefi_Mhc														
Erlo_HepG2														
Erlo_Hep3B														
Erlo_Huh7														
Erlo_Mhc														
Dmso_HepG2														
Dmso_Hep3B														
Dmso_Huh7														
Dmso_Mhc														
Borte_HepG2														
Borte_Hep3B														
Borte_Huh7														
Borte_Mhc														
Kernel	0	0	0	0	0	0	2/2/0.1	0	0	0	0	0	1/2/.1	0
Accuracy	0.393	0.143	0.143	0.357	0.107	0.036	0.571	0.071	0.179	0.143	0.179	0.286	0.571	0.464
Sensitivity	0.563	0.25	0.25	0.625	0.188	0.063	1	0.125	0.313	0.25	0.313	0.5	1	0.813
Specificity	0.167	0	0	0	0	0	0	0	0	0	0	0	0	0

Πίνακας 4.2. Αποτελέσματα κατηγοριοποίησης με integrated dataset, χρησιμοποιώντας όλες της κυτταροσειρές και τη μέθοδο one-only, όπου τα folds χωρίστηκαν κατά φάρμακο.

	ALL	AKT	CREB	ERK12	HSP27	ikb	IRB	IRS1	JNK	MEK1	P38	P70S6	cmet	igfr
Suni_Hep3B														
Sora_Hep3B														
Lapa_Hep3B														
Gefi_Hep3B														
Erlo_Hep3B														
Dmso_Hep3B														
Borte_Hep3B														

Kernel	ALL	1/2/0.1	0	1/2/0.1	0	0	0	1/2/0.1	0	1/2/0.1	ALL	0	0	0
Accuracy	0	0.143	0.286	0.286	0.143	0.143	0.571	0.143	0.143	0.286	0	0.429	0.571	0.571
Sensitivity	0	0.25	0.5	0.5	0.25	0.25	1	0.25	0.25	0.5	0	0.75	1	1
Specificity	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Πίνακας 3: Αποτελέσματα κατηγοριοποίησης χρησιμοποιώντας δεδομένα από την κυτταροσειρά Hep3B και τη μέθοδο one-only, μετά από δοκιμή όλων των παραμέτρων και εύρεση των βέλτιστων.

	ALL	AKT	CREB	ERK12	HSP27	ikb	IRB	IRS1	JNK	MEK1	P38	P70S6	cmet	igfr
Suni_HepG2														
Sora_HepG2														
Lapa_HepG2														
Gefi_HepG2														
Erlo_HepG2														
Dmso_HepG2														
Borte_HepG2														

Kernel	0	0	1/2/0.1	0	0	0	0	0	1/2/0.1	1/2/0.1	1/2/0.1	1/2/0.1	0	0
Accuracy	0.714	0.143	0.286	0.286	0.571	0.429	0.571	0.429	0.571	0.143	0.429	0.286	0.571	0.143
Sensitivity	0.75	0.25	0.5	0.5	1	0.75	1	0.75	1	0.25	0.75	0.5	1	0.25
Specificity	0.667	0	0	0	0	0	0	0	0	0	0	0	0	0

Πίνακας 4: Αποτελέσματα κατηγοριοποίησης χρησιμοποιώντας δεδομένα από την κυτταροσειρά HepG2 και τη μέθοδο one-only, μετά από δοκιμή όλων των παραμέτρων και εύρεση των βέλτιστων.

	ALL	AKT	CREB	ERK12	HSP27	ikb	IRB	IRS1	JNK	MEK1	P38	P70S6	cmet	igfr
Suni_huh7														
Sora_huh7														
Lapa_huh7														
Gefi_huh7														
Erlo_huh7														
Dmso_huh7														
Borte_huh7														

Kernel	0	1/2/0.1	0	0	0	0	0	0	0	0	0	1/2/0.1	0	0
Accuracy	0.143	0.286	0.143	0.286	0.286	0.143	0.286	0.286	0.143	0.143	0.143	0.286	0.571	0.571
Sensitivity	0.25	0.5	0.25	0.5	0.5	0.25	0.5	0.5	0.25	0.25	0.25	0.5	1	1
Specificity	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Πίνακας 4.5: Αποτελέσματα κατηγοριοποίησης χρησιμοποιώντας δεδομένα από την κυτταροσειρά huh7 και τη μέθοδο one-only, μετά από δοκιμή όλων των παραμέτρων και εύρεση των βέλτιστων.

	ALL	AKT	CREB	ERK12	HSP27	ikb	IRB	IRS1	JNK	MEK1	P38	P70S6	cmet	igfr
Suni_mhc														
Sora_mhc														
Lapa_mhc														
Gefi_mhc														
Erlo_mhc														
Dmso_mhc														
Borte_mhc														

Kernel	0	1/2/0.1	0	0	0	0	0	0	0	0	0	0	0	0
Accuracy	0.714	0.286	0.571	0.143	0.143	0.571	0.571	0.143	0.429	0.429	0.571	0.286	0.571	0.571
Sensitivity	0.75	0.5	1	0.25	0.25	1	1	0.25	0.75	0.75	1	0.5	1	1
Specificity	0.667	0	0	0	0	0	0	0	0	0	0	0	0	0

Πίνακας 4.6: Αποτελέσματα κατηγοριοποίησης χρησιμοποιώντας δεδομένα από την κυτταροσειρά mhc και τη μέθοδο one-only, μετά από δοκιμή όλων των παραμέτρων και εύρεση των βέλτιστων.

	ALL	ALL – AKT	ALL – CREB	ALL – ERK12	ALL – HSP27	ALL – Ikb	ALL – IRB	ALL – IRS1	ALL – JNK	ALL – MEK1	ALL – P38	ALL – P70S6	ALL – cmet	ALL – igfr
Suni_HepG2														
Suni_Hep3B														
Suni_Huh7														
Suni_Mhc														
Sora_HepG2														
Sora_Hep3B														
Sora_Huh7														
Sora_Mhc														
Lapa_HepG2														
Lapa_Hep3B														
Lapa_Huh7														
Lapa_Mhc														
Gefi_HepG2														
Gefi_Hep3B														
Gefi_Huh7														
Gefi_Mhc														
Erlo_HepG2														
Erlo_Hep3B														
Erlo_Huh7														
Erlo_Mhc														
Dmso_HepG2														
Dmso_Hep3B														
Dmso_Huh7														
Dmso_Mhc														
Borte_HepG2														
Borte_Hep3B														
Borte_Huh7														
Borte_Mhc														
Kernel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Accuracy	0.393	0.357	0.357	0.357	0.429	0.393	0.214	0.393	0.321	0.429	0.357	0.393	0.393	0.429
Sensitivity	0.563	0.563	0.5	0.5	0.625	0.563	0.313	0.563	0.563	0.625	0.5	0.563	0.563	0.563
Specificity	0.167	0.083	0.167	0.167	0.167	0.167	0.083	0.167	0	0.167	0.167	0.167	0.167	0.25

Πίνακας 4.7. Αποτελέσματα κατηγοριοποίησης με integrated dataset χρησιμοποιώντας όλες της κυτταροσειρές και τη μέθοδο one-out, όπου τα folds χωρίστηκαν κατά φάρμακο.

	ALL	ALL – AKT	ALL – CREB	ALL – ERK12	ALL – HSP27	ALL – Ikb	ALL – IRB	ALL – IRS1	ALL – JNK	ALL – MEK1	ALL – P38	ALL – P70S6	ALL – cmet	ALL – igfr
Suni_Hep3B														
Sora_Hep3B														
Lapa_Hep3B														
Gefi_Hep3B														
Erlo_Hep3B														
Dmso_Hep3B														
Borte_Hep3B														

Kernel	ALL	ALL	1/2/0.1	ALL	0	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL
Accuracy	0	0	0.143	0	0.143	0	0	0	0	0	0	0	0	0
Sensitivity	0	0	0.25	0	0.25	0	0	0	0	0	0	0	0	0
Specificity	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Πίνακας 4.8: Αποτελέσματα κατηγοριοποίησης χρησιμοποιώντας δεδομένα από την κυτταροσειρά Hep3B και τη μέθοδο one-out, μετά από δοκιμή όλων των παραμέτρων και εύρεση των βέλτιστων.

	ALL	ALL – AKT	ALL – CREB	ALL – ERK12	ALL – HSP27	ALL – Ikb	ALL – IRB	ALL – IRS1	ALL – JNK	ALL – MEK1	ALL – P38	ALL – P70S6	ALL – cmet	ALL – igfr
Suni_HepG2														
Sora_HepG2														
Lapa_HepG2														
Gefi_HepG2														
Erlo_HepG2														
Dmso_HepG2														
Borte_HepG2														

Kernel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Accuracy	0.714	0.714	0.571	0.571	0.571	0.714	0.571	0.571	0.571	0.714	0.429	0.714	0.714	0.714
Sensitivity	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
Specificity	0.667	0.667	0.333	0.333	0.333	0.667	0.333	0.333	0.333	0.667	0	0.667	0.667	0.667

Πίνακας 4.9: Αποτελέσματα κατηγοριοποίησης χρησιμοποιώντας δεδομένα από την κυτταροσειρά HepG2 και τη μέθοδο one-out, μετά από δοκιμή όλων των παραμέτρων και εύρεση των βέλτιστων.

	ALL	ALL – AKT	ALL – CREB	ALL – ERK12	ALL – HSP27	ALL – Ikb	ALL – IRB	ALL – IRS1	ALL – JNK	ALL – MEK1	ALL – P38	ALL – P70S6	ALL – cmet	ALL – igfr
Suni_huh7														
Sora_huh7														
Lapa_huh7														
Gefi_huh7														
Erlo_huh7														
Dmso_huh7														
Borte_huh7														

Kernel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Accuracy	0.143	0.143	0.143	0.143	0.143	0.143	0.143	0.143	0.143	0.143	0.143	0.143	0.143	0.143
Sensitivity	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Specificity	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Πίνακας 4.10: Αποτελέσματα κατηγοριοποίησης χρησιμοποιώντας δεδομένα από την κυτταροσειρά huh7 και τη μέθοδο one-out, μετά από δοκιμή όλων των παραμέτρων και εύρεση των βέλτιστων.

	ALL	ALL – AKT	ALL – CREB	ALL – ERK12	ALL – HSP27	ALL – Ikb	ALL – IRB	ALL – IRS1	ALL – JNK	ALL – MEK1	ALL – P38	ALL – P70S6	ALL – cmet	ALL – igfr
Suni_mhc														
Sora_mhc														
Lapa_mhc														
Gefi_mhc														
Erlo_mhc														
Dmso_mhc														
Borte_mhc														

Kernel	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Accuracy	0.714	0.714	0.714	0.714	1	0.714	0.714	0.714	0.571	0.714	0.714	0.714	0.714	0.714
Sensitivity	0.75	0.75	0.75	0.75	1	0.75	0.75	0.75	0.5	0.75	0.75	0.75	0.75	0.75
Specificity	0.667	0.667	0.667	0.667	1	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667	0.667

Πίνακας 4.11: Αποτελέσματα κατηγοριοποίησης χρησιμοποιώντας δεδομένα από την κυτταροσειρά mhc και τη μέθοδο one-out, μετά από δοκιμή όλων των παραμέτρων και εύρεση των βέλτιστων.

4.7. Πρόβλεψη για την αποτελεσματικότητα νέων φαρμάκων

Μετά την διεξαγωγή των παραπάνω πειραμάτων έγινε μια δοκιμή να κατηγοριοποιήσουμε 5 νέα φάρμακα κάνοντας μια πρόβλεψη για την αποτελεσματικότητα τους. Τα φάρμακα αυτά χορηγήθηκαν σε κυτταροσειρές ίδιου τύπου με αυτές για τα γνωστά φάρμακα.

Τα φάρμακα αυτά είναι τα εξής:

- PI103
- MEK32
- JNJ
- VANDE
- DASA

Για τη διαδικασία αυτή χρησιμοποιήθηκαν οι μετρήσεις από τα 7 φάρμακα που χρησιμοποιήσαμε στην προηγούμενη ενότητα για εκπαίδευση του αλγορίθμου (train dataset) ώστε να γίνει η πρόβλεψη για τα 5 νέα φάρμακα.

Επίσης σε όλες τις μετρήσεις χρησιμοποιήθηκε γραμμικός kernel, αφού σχεδόν σε όλες τις προηγούμενες δοκιμές έδειξε τα καλύτερα αποτελέσματα. Στον πίνακα 4.12 φαίνονται τα αποτελέσματα που λάβαμε από τον κώδικα.

A. INTEGRATED	
'PI103_HepG2'	**FAIL**
'PI103_Hep3B'	**FAIL**
'PI103_Huh7'	**FAIL**
'PI103_Mhc'	**FAIL**
'MEK32_HepG2'	**FAIL**
'MEK32_Hep3B'	**FAIL**
'MEK32_Huh7'	**PASS**
'MEK32_Mhc'	**FAIL**
'JNJ_HepG2'	**FAIL**
'JNJ_Hep3B'	**FAIL**
'JNJ_Huh7'	**FAIL**
'JNJ_Mhc'	**FAIL**
'VANDE_HepG2'	**FAIL**
'VANDE_Hep3B'	**PASS**
'VANDE_Huh7'	**PASS**
'VANDE_Mhc'	**FAIL**
'DASA_HepG2'	**PASS**
'DASA_Hep3B'	**FAIL**
'DASA_Huh7'	**FAIL**
'DASA_Mhc'	**PASS**

B. Hep3B	
'PI103_Hep3B'	**FAIL**
'MEK32_Hep3B'	**FAIL**
'JNJ_Hep3B'	**FAIL**
'VANDE_Hep3B'	**FAIL**
'DASA_Hep3B'	**FAIL**

C. HepG2	
PI103_HepG2'	**PASS**
'MEK32_HepG2'	**FAIL**
'JNJ_HepG2'	**FAIL**
'VANDE_HepG2'	**FAIL**
'DASA_HepG2'	**FAIL**

D. Huh7	
'PI103_Huh7'	**FAIL**
'MEK32_Huh7'	**PASS**
'JNJ_Huh7'	**FAIL**
'VANDE_Huh7'	**PASS**
'DASA_Huh7'	**FAIL**

E. Mhc	
'PI103_Mhc'	**FAIL**
'MEK32_Mhc'	**FAIL**
'JNJ_Mhc'	**FAIL**
'VANDE_Mhc'	**FAIL**
'DASA_Mhc'	**PASS**

Πίνακας 4.12: Πρόβλεψη για την αποτελεσματικότητα των νέων φαρμάκων, χρησιμοποιώντας γραμμικό kernel, σε integrated dataset χρησιμοποιώντας όλες της κυτταροσειρές (11A) και σε κάθε κυτταροσειρά ξεχωριστά (11B-E).

5. Επίλογος

Σε αυτή τη διπλωματική εργασία ερευνήσαμε τη χρήση των Support Vector Machines σε συνδυασμό με συγκεκριμένα σύνολα δεδομένων, με σκοπό την κατηγοριοποίηση στην εξέλιξη φαρμάκων.

Ασχοληθήκαμε τόσο με την επαλήθευση δεδομένων χρησιμοποιώντας φάρμακα το αποτέλεσμα των οποίων είναι γνωστό, όσο και με την προσπάθεια πρόβλεψης για την αποτελεσματικότητα ενός νέου φαρμάκου.

Οι ακρίβειες κατά την επαλήθευση κυμαίνονταν από 0 μέχρι και 71.4%, και στην κυτταροσειρά Mhc είδαμε μέχρι και 100%, αλλά με specificity ίσο με 1, το οποίο υποδηλώνει ότι το αποτέλεσμα μπορεί να είναι πλασματικό.

Οι κυτταροσειρές Hep3B και Huh7 ανταποκρίθηκαν καλύτερα με τη μέθοδο one-only, ενώ οι κυτταροσειρές HepG2 και mhc έδωσαν μεγαλύτερη ακρίβεια με τη μέθοδο one-out.

Οι κυτταροσειρές στις οποίες ο κατηγοριοποιητής φάνηκε να δίνει καλύτερα αποτελέσματα είναι οι HepG2 και Mhc, ειδικά με τη μέθοδο one-out. Επίσης όταν ο αλγόριθμος δούλευε με integrated datasets έδινε καλύτερη ακρίβεια για τα φάρμακα Lapatinib, Dmso και Bortezomib, τα οποία είναι όλα φάρμακα που έχουν αποτύχει στις κλινικές δοκιμές.

Από τα αποτελέσματα που λάβαμε είναι προφανές ότι δεν μπορούμε να εγγυηθούμε την αξιοπιστία των αποτελεσμάτων, και σίγουρα χρειάζεται βελτίωση στο κομμάτι της πρόβλεψης.

Μια διαφορετική υλοποίηση του κώδικα σε συνδυασμό με τη δοκιμασία διαφορετικών SVM toolboxes να βοηθήσει σημαντικά στη βελτίωση της ποιότητας των αποτελεσμάτων, ώστε το ποσοστό επιτυχίας μιας πρόβλεψης να είναι πιο αποτελεσματικό και αξιόπιστο, εμπνέοντας εμπιστοσύνη στα αποτελέσματα.

6. Βιβλιογραφία

[1] Pharmacokinetics. (2006). In *Mosby's Dictionary of Medicine, Nursing, & Health Professions*. Philadelphia, PA: Elsevier Health Sciences. Retrieved December 11, 2008.

[2] Lees P, Cunningham FM, Elliott J (2004). "Principles of pharmacodynamics and their applications in veterinary pharmacology". *J. Vet. Pharmacol. Ther.* **27** (6): 397–414.

[3] "In Vitro Biocompatibility Testing of Biomaterials and Medical Devices", U. Muller, Medical Device Technology, March 2008

[4] Vignais, Paulette M.; Pierre Vignais (2010). *Discovering Life, Manufacturing Life: How the experimental method shaped life sciences*. Berlin: Springer.

[5] Life Science Technologies Cell Signaling: In Vivo Veritas, Science Magazine, 2007

[6] "Use of Laboratory Animals in Biomedical and Behavioral Research", Institute for Laboratory Animal Research, The National Academies Press, 1988. Also see Cooper, Sylvia. "Pets crowd animal shelter", *The Augusta Chronicle*, August 1, 1999; and Gillham, Christina.

[7] Avorn J. (2004). *Powerful Medicines*, pp. 129-133. Alfred A. Knopf.

[8] Van Spall HG, Toren A, Kiss A, Fowler RA (March 2007). "Eligibility criteria of randomized controlled trials published in high-impact general medical journals: a systematic sampling review". *JAMA* **297** (11): 1233–40.

[9] The regulatory authority in the USA is the Food and Drug Administration; in Canada, Health Canada; in the European Union, the European Medicines Agency; and in Japan, the Ministry of Health, Labour and Welfare

[10] Crossley, MJ; Turner, P; Thordarson, P (2007). "Clinical Trials - What Your Need to Know". *American Cancer Society* **129** (22): 7155.

[11] DiMasi J (2002). "The value of improving the productivity of the drug development process: faster times and better decisions".

Pharmacoeconomics **20 Suppl 3**: 1–10.

[12] DiMasi J, Hansen R, Grabowski H (2003). "The price of innovation: new estimates of drug development costs". *J Health Econ* **22** (2): 151– 85.

[13] Adams C, Brantner V (2006). "Estimating the cost of new drug development: is it really 802 million dollars?". *Health Aff (Millwood)* **25** (2): 420–8.

[14]Christopher Paul Adams, Van Vu Brantner (February 2009) "Health Economics", Volume 19, Issue 2, pages 130-141,

[15] Stratmann, Dr. H.G. (September 2010). *Analog Science Fiction and Fact* **CXXX** (9): 20.

[16] "R&D costs are on the rise". *Medical Marketing and Media*. June 2003.

[17] Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Haussler D, editor. 5th Annual ACM Workshop on COLT. Pittsburgh (Pennsylvania): ACM Press. pp. 144–152.

[18] Schölkopf B, Smola A (2002) Learning with kernels. Cambridge (Massachusetts): MIT Press.

[19] Vapnik V (1999) The nature of statistical learning theory. 2nd edition. Springer.

[20] Müller KR, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Netw 12: 181–201.

[21] Schölkopf B, Tsuda K, Vert JP (2004) Kernel methods in computational biology. Cambridge (Massachusetts): MIT Press.

[22] Vert JP (2007) Kernel methods in genomics and computational biology. In: Camps-Valls G, Rojo-Alvarez JL, Martinez-Ramon M, editors. Kernel methods in bioengineering, signal and image processing. Idea Group. pp. 42–63.

[23] Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge (United Kingdom): Cambridge University Press.

[24] Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20: 273–297.

[25] Chang CC, Lin CJ (2001) LIBSVM: A library for support vector machines.

[26] Provost F (2000) Learning with imbalanced data sets 101.

[27] Pavlidis P, Weston J, Cai J, Noble W (2002) Learning gene functional classifications from multiple data types. J Comput Biol 9: 401–411.

[28] Lanckriet G, Bie TD, Cristianini N, Jordan M, Noble W (2004) A statistical framework for genomic data fusion. Bioinformatics 20: 2626–2635.

[29] Ben-Hur A, Noble WS (2005) Kernel methods for predicting protein–protein interactions. Bioinformatics 21: (Supplement 1)i38–i46.

[30] Ben-Hur A, Ong CS, Sonnenburg S, Schölkopf B, Rätsch G (2008) Support Vector Machines and Kernels for Computational Biology. PLoS Comput Biol 4(10): e1000173.

ΠΑΡΑΡΤΗΜΑ

1. Πρόγραμμα "**main_all.m**" με σκοπό τον χειρισμό των ξεχωριστών κυτταροσειρών χρησιμοποιώντας όλα τα σήματα.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   THIS IS THE MAIN PART OF THE ALGORITHM FOR EACH SEPARATE CELL LINE
%   IT SEPARATES THE DATASET IN FOLDS AND EACH FOLD REPRESENTS THE
%   OBSERVATIONS WE HAVE FROM EACH MEDICINE ON A PARTICULAR CELL LINE
%   EACH TIME ONE DRUG IS USED AS A TEST DATASET
%   AND THE REST ARE USED AS TRAIN DATASETS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
load bool_0_60_train.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   SELECT CELL LINE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k_folds = huh7_data.folded_data;
actual_labels=huh7_data.data(:,end);
L=7;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PARAMETERS FOR REPETITIVE RUNS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

kernel=[0 1 2];
gamma=[0.1 0.5 1 2 5 10 15 20 30];
degree=[2 3 4 5 6];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

final_labels2=[]; counter=0;
for count_kernel=1:size(kernel,2);
    final_labels=[];
    for count_gamma = 1:size(gamma,2)
        for count_degree = 1:size(degree,2)
            krnl = ['-t ', num2str(kernel(1,count_kernel)),'-
d',num2str(degree(1,count_degree)),'-g',
num2str(gamma(1,count_gamma))];
            assigned_labels=[];
            TP=0;TN=0;FP=0;FN=0;
            for i=1:L
```

```
                train_data=[];
                test_data=k_folds{i,1};
                for j=1:L
                    if j~=i
                        train_data=[train_data;k_folds{j,1}];
                    end
                end
                model = svmtrain(train_data(:,size(train_data,2)),
train_data(:,1:(size(train_data,2)-1)), krnl);
                [predict_label_L, accuracy_L, dec_values_L] =
svmpredict(test_data(:,size(test_data,2)),
test_data(:,1:(size(test_data,2)-1)), model);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       CALCULATE TP,TN,FP,FN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                for k=1:size(predict_label_L,1)
                    if predict_label_L(k,1)>0
                        if test_data(k,size(train_data,2))>0
                            TP=TP+1;
                        elseif test_data(k,size(train_data,2))<0
                            FP=FP+1;
                        end
                    elseif predict_label_L(k,1)<0
                        if test_data(k,size(train_data,2))<0
                            TN=TN+1;
                        elseif test_data(k,size(train_data,2))>0
                            FN=FN+1;
                        end
                    end
                end
                assigned_labels=[assigned_labels;predict_label_L];
                clear model
            end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     CALCULATION OF ACCURACY, SENSITIVITY,SPECIFICITY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            accuracy=(TN+TP)/7;
            sens=(TN/(TN+FP));
            spec=(TP/(TP+FN));

labels=[kernel(1,count_kernel)*ones(size(assigned_labels,1),1),degree(1
,count_degree)*ones(size(assigned_labels,1),1),gamma(1,count_gamma)*one
s(size(assigned_labels,1),1),assigned_labels,actual_labels,10*(assigned
_labels==actual_labels),accuracy*ones(size(assigned_labels,1),1),sens*o
nes(size(assigned_labels,1),1),spec*ones(size(assigned_labels,1),1)];
            counter=counter+1;
            final_labels=[final_labels;labels];
        end
    end
    final_labels2=[final_labels2;final_labels];
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   EXPORT SVM DATA TO XLS FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xlswrite('results_all.xls',final_labels2,'huh7')
```

2. Πρόγραμμα "**main_1_out.m**" για integrated datasets και ξεχωριστών κυτταροσειρών, το οποίο αφαιρεί μια στήλη για τη δημιουργία και των χειρισμό των dataset "**one_only**" και "**one_out**".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   THIS IS THE MAIN PART OF THE ALGORITHM FOR EACH SEPARATE CELL LINE
%   IT SEPARATES THE DATASET IN FOLDS AND EACH FOLD REPRESENTS THE
%   OBSERVATIONS WE HAVE FROM EACH MEDICINE ON A PARTICULAR CELL LINE
%   EACH TIME ONE DRUG IS USED AS A TEST DATASET
%   AND THE REST ARE USED AS TRAIN DATASETS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
load bool_0_60_train.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   SELECT CELL LINE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k_folds = huh7_data.folded_data;
actual_labels=huh7_data.data(:,end);
L=7;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PARAMETERS FOR REPETITIVE RUNS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

kernel=[0 1 2];
gamma=[0.1 0.5 1 2 5 10 15 20 30];
degree=[2 3 4 5 6];


final_labels2=[]; counter=0;
for count_kernel=1:size(kernel,2);
    final_labels=[];
    for count_gamma = 1:size(gamma,2)
        for count_degree = 1:size(degree,2)
            krnl = ['-t ', num2str(kernel(1,count_kernel)),'-
d',num2str(degree(1,count_degree)),'-g',
num2str(gamma(1,count_gamma))];
            assigned_labels=[];
            TP=0;TN=0;FP=0;FN=0;
            for i=1:L
                train_data=[];
                test_data=k_folds{i,1};
                for j=1:L
                    if j~=i
```

```
                        train_data=[train_data;k_folds{j,1}];
                    end
                end
                model = svmtrain(train_data(:,size(train_data,2)),
train_data(:,1:(size(train_data,2)-1)), krnl);
                [predict_label_L, accuracy_L, dec_values_L] =
svmpredict(test_data(:,size(test_data,2)),
test_data(:,1:(size(test_data,2)-1)), model);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     CALCULATE TP,TN,FP,FN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                for k=1:size(predict_label_L,1)
                    if predict_label_L(k,1)>0
                        if test_data(k,size(train_data,2))>0
                            TP=TP+1;
                        elseif test_data(k,size(train_data,2))<0
                            FP=FP+1;
                        end
                    elseif predict_label_L(k,1)<0
                        if test_data(k,size(train_data,2))<0
                            TN=TN+1;
                        elseif test_data(k,size(train_data,2))>0
                            FN=FN+1;
                        end
                    end
                end
                assigned_labels=[assigned_labels;predict_label_L];
                clear model
            end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     CALCULATION OF ACCURACY, SENSITIVITY,SPECIFICITY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            accuracy=(TN+TP)/7;
            sens=(TN/(TN+FP));
            spec=(TP/(TP+FN));

labels=[kernel(1,count_kernel)*ones(size(assigned_labels,1),1),degree(1
,count_degree)*ones(size(assigned_labels,1),1),gamma(1,count_gamma)*one
s(size(assigned_labels,1),1),assigned_labels,actual_labels,10*(assigned
_labels==actual_labels),accuracy*ones(size(assigned_labels,1),1),sens*o
nes(size(assigned_labels,1),1),spec*ones(size(assigned_labels,1),1)];
            counter=counter+1;
            final_labels=[final_labels;labels];
        end
    end
    final_labels2=[final_labels2;final_labels];
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   EXPORT SVM DATA TO XLS FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xlswrite('results_all.xls',final_labels2,'huh7')
```

3. Υποπρόγραμμα "**creating_k_folds.m**" το οποίο χωρίζει το dataset σε folds

```
function k_folds=creating_k_folds(data)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   CREATION OF FOLDS PER DRUG FOR THE PRESET CELL LINE
%   TO FACILITATE DATA USAGE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k_folds{1,1}=data(1,:);
k_folds{2,1}=data(2,:);
k_folds{3,1}=data(3,:);
k_folds{4,1}=data(4,:);
k_folds{5,1}=data(5,:);
k_folds{6,1}=data(6,:);
k_folds{7,1}=data(7,:);
end
```

4. Πρόγραμμα "**main_all_int.m**" με σκοπό τον χειρισμό των integrated dataset χρησιμοποιώντας όλα τα σήματα.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   THIS IS THE MAIN PART OF THE ALGORITHM FOR INTEGRATED DATASETS
%   IT SEPARATES THE DATASET IN FOLDS AND EACH FOLD REPRESENTS THE
%   OBSERVATIONS WE HAVE FROM EACH MEDICINE ON ALL CELL LINES
%   EACH TIME ONE DRUG IS USED AS A TEST DATASET
%   AND THE REST ARE USED AS TRAIN DATASETS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
load bool_0_60_train.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   SELECT INTEGRATED DATASET
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k_folds = integrated_data.folded_data;
actual_labels=integrated_data.data(:,end);
L=7;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PARAMETERS FOR REPETITIVE RUNS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

kernel=[0 1 2];
gamma=[0.1 0.5 1 2 5 10 15 20 30];
degree=[2 3 4 5 6];


final_labels2=[]; counter=0;
for count_kernel=1:size(kernel,2);
    final_labels=[];
    for count_gamma = 1:size(gamma,2)
        for count_degree = 1:size(degree,2)
            krnl = ['-t ', num2str(kernel(1,count_kernel)),'-
d',num2str(degree(1,count_degree)),'-g',
num2str(gamma(1,count_gamma))];
            assigned_labels=[];
            TP=0;TN=0;FP=0;FN=0;
            for i=1:L
                train_data=[];
                test_data=k_folds{i,1};
                for j=1:L
                    if j~=i
                        train_data=[train_data;k_folds{j,1}];
                    end
```

```
                end
                model = svmtrain(train_data(:,size(train_data,2)),
train_data(:,1:(size(train_data,2)-1)), krnl);
                [predict_label_L, accuracy_L, dec_values_L] =
svmpredict(test_data(:,size(test_data,2)),
test_data(:,1:(size(test_data,2)-1)), model);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     CALCULATE TP,TN,FP,FN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                for k=1:size(predict_label_L,1)
                    if predict_label_L(k,1)>0
                        if test_data(k,size(train_data,2))>0
                            TP=TP+1;
                        elseif test_data(k,size(train_data,2))<0
                            FP=FP+1;
                        end
                    elseif predict_label_L(k,1)<0
                        if test_data(k,size(train_data,2))<0
                            TN=TN+1;
                        elseif test_data(k,size(train_data,2))>0
                            FN=FN+1;
                        end
                    end
                end
                assigned_labels=[assigned_labels;predict_label_L];
                clear model
            end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     CALCULATE ACCURACY, SENSITIVITY,SPECIFICITY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            accuracy=(TN+TP)/size(assigned_labels,1);
            sens=(TN/(TN+FP));
            spec=(TP/(TP+FN));

labels=[kernel(1,count_kernel)*ones(size(assigned_labels,1),1),degree(1
,count_degree)*ones(size(assigned_labels,1),1),gamma(1,count_gamma)*one
s(size(assigned_labels,1),1),assigned_labels,actual_labels,10*(assigned
_labels==actual_labels),accuracy*ones(size(assigned_labels,1),1),sens*o
nes(size(assigned_labels,1),1),spec*ones(size(assigned_labels,1),1)];
            counter=counter+1;
            final_labels=[final_labels;labels];
        end
    end
    final_labels2=[final_labels2;final_labels];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   EXPORT SVM DATA TO XLS FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
xlswrite('results_all.xls',final_labels2)
```

5. Πρόγραμμα "**main_1_out_int.m**" για integrated datasets, το οποίο αφαιρεί μια στήλη για τη δημιουργία και των χειρισμό των dataset "**one_only**" και "**one_out**".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   THIS IS THE MAIN PART OF THE ALGORITHM FOR EACH SEPARATE CELL LINE
%   IT SEPARATES THE DATASET IN FOLDS AND EACH FOLD REPRESENTS THE
%   OBSERVATIONS WE HAVE FROM EACH MEDICINE ON A PARTICULAR CELL LINE
%   IT FUNCTIONS BY LEAVING OUT ONE OF THE 14 COLUMNS TO CREATE TWO
%   DATASETS, ONE DATASET CONTAINING ALL COLUMNS MINUS ONE COLUMN AND
%   ONE DATASET CONTAING ONLY THE LEFT OUT COLUMN
%   EACH TIME ONE DRUG IS USED AS A TEST DATASET
%   AND THE REST ARE USED AS TRAIN DATASETS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   DATASET SELECTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load bool_0_60_train.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   ITNEGRATED DATASET SELECTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

train_data = integrated_data.data;
actual_labels=integrated_data.data(:,end);
L=7;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PARAMETERS FOR REPETITIVE RUNS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

kernel=[0 1 2];
gamma=[0.1 0.5 1 2 5 10 15 20 30];
degree=[2 3 4 5 6];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   SELECTION OF COLUMN TO BE LEFT OUT AND CREATION OF THE TWO
%   DATASETS DATA1 CONTAINS ALL MINUS ONE COLUMN AND DATA2
%   CONTAINS ONLY ONE COLUMN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for col=1:13
    if col==1
```

```
        data1=train_data(:,8:end);
    elseif col==13
        data1=[train_data(:,1:end-8),train_data(:,end)];
    else
        data1=[train_data(:,1:((col-
1)*7)),train_data(:,(col*7+1):end)];
    end
    data2=[train_data(:,((col-1)*7+1):(col*7)),train_data(:,end)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   k_folds_1: CREATES FOLDS FROM DATA1
%   k_folds_2: CREATES FOLDS FROM DATA2
%   FINAL LABELS: CONTAINS ALL THE LABELS FROM TESTING EACH DRUG
%                 ON ALL CELL LINES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    k_folds_1=creating_k_folds_int(data1);
    k_folds_2=creating_k_folds_int(data2);
    final_labels2_1=[]; final_labels2_2=[];
    counter=0;
    for count_kernel=1:size(kernel,2);
        final_labels_1=[];final_labels_2=[];
        for count_gamma = 1:size(gamma,2)
            for count_degree = 1:size(degree,2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   SVM PARAMETERS AR GIVEN FOR EACH RUN
%   AND STARTING VALUES SET FOR THE CONFUSION MATRIX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                krnl = ['-t ', num2str(kernel(1,count_kernel)),'-
d',num2str(degree(1,count_degree)),'-g',
num2str(gamma(1,count_gamma))];
                assigned_labels_1=[];assigned_labels_2=[];
                TP1=0;TN1=0;FP1=0;FN1=0;
                TP2=0;TN2=0;FP2=0;FN2=0;
                for i=1:L
                    train_data_1=[];train_data_2=[];

test_data_1=k_folds_1{i,1};test_data_2=k_folds_2{i,1};
                    for j=1:L
                        if j~=i
                            train_data_1=[train_data_1;k_folds_1{j,1}];
                            train_data_2=[train_data_2;k_folds_2{j,1}];
                        end
                    end
                    model_1 =
svmtrain(train_data_1(:,size(train_data_1,2)),
train_data_1(:,1:(size(train_data_1,2)-1)), krnl);
                    model_2 =
svmtrain(train_data_2(:,size(train_data_2,2)),
train_data_2(:,1:(size(train_data_2,2)-1)), krnl);
```

```
                    [predict_label_L_1, accuracy_L_1, dec_values_L_1] =
svmpredict(test_data_1(:,size(test_data_1,2)),
test_data_1(:,1:(size(test_data_1,2)-1)), model_1);
                    [predict_label_L_2, accuracy_L_2, dec_values_L_2] =
svmpredict(test_data_2(:,size(test_data_2,2)),
test_data_2(:,1:(size(test_data_2,2)-1)), model_2);
                    [TN1 TP1 FN1
FP1]=confusion_matrix(predict_label_L_1,test_data_1,train_data_1,TN1,TP
1,FN1,FP1);
                    [TN2 TP2 FN2
FP2]=confusion_matrix(predict_label_L_2,test_data_2,train_data_2,TN2,TP
2,FN2,FP2);

assigned_labels_1=[assigned_labels_1;predict_label_L_1];

assigned_labels_2=[assigned_labels_2;predict_label_L_2];
                    clear model_1 model_2
                end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     CALCULATION OF ACCURACY, ENSITIVITY,SPECIFICITY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


accuracy_1=(TN1+TP1)/size(assigned_labels_1,1);accuracy_2=(TN2+TP2)/siz
e(assigned_labels_2,1);
                sens_1=(TN1/(TN1+FP1));sens_2=(TN2/(TN2+FP2));
                spec_1=(TP1/(TP1+FN1));spec_2=(TP2/(TP2+FN2));

labels_1=[kernel(1,count_kernel)*ones(size(assigned_labels_1,1),1),degr
ee(1,count_degree)*ones(size(assigned_labels_1,1),1),gamma(1,count_gamm
a)*ones(size(assigned_labels_1,1),1),assigned_labels_1,actual_labels,10
*(assigned_labels_1==actual_labels),accuracy_1*ones(size(assigned_label
s_1,1),1),sens_1*ones(size(assigned_labels_1,1),1),spec_1*ones(size(ass
igned_labels_1,1),1)];

labels_2=[kernel(1,count_kernel)*ones(size(assigned_labels_2,1),1),degr
ee(1,count_degree)*ones(size(assigned_labels_2,1),1),gamma(1,count_gamm
a)*ones(size(assigned_labels_2,1),1),assigned_labels_2,actual_labels,10
*(assigned_labels_2==actual_labels),accuracy_2*ones(size(assigned_label
s_2,1),1),sens_2*ones(size(assigned_labels_2,1),1),spec_2*ones(size(ass
igned_labels_2,1),1)];
                counter=counter+1;
                final_labels_1=[final_labels_1;labels_1];
                final_labels_2=[final_labels_2;labels_2];
            end
        end
        final_labels2_1=[final_labels2_1;final_labels_1];
        final_labels2_2=[final_labels2_2;final_labels_2];
    end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   EXPORT SVM DATA TO XLS FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    sheet_1=sheet_name(col);

    sheet_2=['all - ',sheet_1];
    xlswrite('results_1_out.xls',final_labels2_1,sheet_2);
    xlswrite('results_1_only.xls',final_labels2_2,sheet_1);
end
```

6. Υποπρόγραμμα "**creating_k_folds_int.m**" το οποίο χωρίζει το integrated dataset σε folds.

```
function k_folds=creating_k_folds_int(data)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   CREATION OF FOLDS PER DRUG FOR THE INTEGRATED DATASETS
%   TO FACILITATE DATA USAGE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k_folds{1,1}=data(1:4,:);
k_folds{2,1}=data(5:8,:);
k_folds{3,1}=data(9:12,:);
k_folds{4,1}=data(13:16,:);
k_folds{5,1}=data(17:20,:);
k_folds{6,1}=data(21:24,:);
k_folds{7,1}=data(25:28,:);
end
```

7. Υποπρόγραμμα "**confusion_matrix.m**" στο οποίο υπολογίζονται τα true positives, true negative, false positives, false negatives για περαιτέρω χρήση.

```
function [TN TP FN FP] = confusion_matrix(predict_label_L, ...
                         test_data,train_data,TN,TP,FN,FP)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   COMPILATION OF A CONFUSION MATRIX THAT CONTAINS
%   TRUE/FALSE POSITIVES/NEGATIVES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   TP: TRUE POSITIVES (POSITIVES PREDICTED AS POSITIVES)
%   FP: FALSE POSITIVES (NEGATIVES PREDICTED AS POSITIVES)
%   TN: TRUE NEGATIVES (NEGATIVES PREDICTED AS NEGATIVES)
%   FN: FALSE NEGATIVES (POSITIVES PREDICTED AS NEGATIVES)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k=1:size(predict_label_L,1)
    if predict_label_L(k,1)>0
        if test_data(k,size(train_data,2))>0
            TP=TP+1;
        elseif test_data(k,size(train_data,2))<0
            FP=FP+1;
        end
    elseif predict_label_L(k,1)<0
        if test_data(k,size(train_data,2))<0
            TN=TN+1;
        elseif test_data(k,size(train_data,2))>0
            FN=FN+1;
        end
    end
end
end
```

8. Υποπρόγραμμα "**sheet_name.m**" το οποίο αποθηκεύει τα ονόματα των σημάτων ώστε να βάλει τις κατάλληλες ονομασίες στα αρχεία excel που εξάγει ο κώδικας.

```
function sheet=sheet_name(col)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   ASSIGNING COLUMN NAMES FOR THE XLS FILE
%   THAT CONTAINS CLASSIFICATION RESULTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if col==1
    sheet='AKT';
elseif col==2
    sheet='CREB';
elseif col==3
    sheet='ERK12';
elseif col==4
    sheet='HSP27';
elseif col==5
    sheet='IKB';
elseif col==6
    sheet='IRB';
elseif col==7
    sheet='IRS1';
elseif col==8
    sheet='JNK';
elseif col==9
    sheet='MEK1';
elseif col==10
    sheet='P38';
elseif col==11
    sheet='P70S6';
elseif col==12
    sheet='cmet';
elseif col==13
    sheet='igfr';
end
```

9. Πρόγραμμα **"main_predict.m"** με σκοπό την πρόβλεψη του label των καινούριων φαρμάκων

```
clear all
close all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     LOAD TRAIN AND TEST DATASETS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

load integrated_unk_tst_data.mat;
test_data=integrated_unk_tst_data.data;

load integrated_vertex_data.mat;
train_data=integrated_data.data

assign_label=test_data(:,size(test_data,2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     SET SVM PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

input('INPUT KERNEL TYPE (0-LINEAR,1-POLYNOMIAL,2-RBF,3-SIGMOID): ');
prm=ans;
input('INPUT DEGREE: ');
dg=ans;
input('INPUT GAMMA VALUE: ');
gm=ans;
krnl = ['-t ', num2str(prm),'-d',num2str(dg),'-g', num2str(gm)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     EXECUTION OF THE MODEL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

model = svmtrain(train_data(:,size(train_data,2)),
train_data(:,1:size(train_data,2)-1), krnl);
[predict_label_L, accuracy_L, dec_values_L] =
svmpredict(assign_label(:,1), test_data(:,1:size(test_data,2)-1),
model);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     EXPORT SVM DATA TO XLS FILES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xlswrite('results.xls',predict_label_L,'integrated','D');
```