

Ατομική Διπλωματική Εργασία

**ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ ΑΠΕΙΚΟΝΙΣΗΣ ΚΥΤΤΑΡΙΚΟΥ ΔΙΚΤΥΟΥ
ΩΣ ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΒΟΗΘΟΣ ΜΟΡΙΑΚΗΣ ΒΙΟΛΟΓΙΑΣ**

Έμιλη Ν. Ταλιαδώρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2015

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ ΑΠΕΙΚΟΝΙΣΗΣ ΚΥΤΤΑΡΙΚΟΥ ΔΙΚΤΥΟΥ ΩΣ
ΕΠΙΣΤΗΜΟΝΙΚΟΣ ΒΟΗΘΟΣ ΜΟΡΙΑΚΗΣ ΒΙΟΛΟΓΙΑΣ**

Έμιλη Ν. Ταλιαδώρου

Επιβλέπων Καθηγητής

Αντώνης Κάκας

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2015

Ευχαριστίες

Φτάνοντας στο τελικό στάδιο των προπτυχιακών μου σπουδών, στο τμήμα Πληροφορικής του Πανεπιστημίου Κύπρου θα ήθελα να ευχαριστήσω όλους όσους συνέβαλαν στην περάτωση της διπλωματικής μου εργασίας.

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον Δρ. Αντώνη Κάκα για την αμέριστη βοήθεια και καθοδήγηση που μου παρείχε στο ερευνητικό στάδιο όσο και στην μετέπειτα μελέτη, εξέλιξη και ολοκλήρωση της διπλωματικής εργασίας που έπεται. Η βοήθεια και οι συμβουλές του διαδραμάτισαν σπουδαίο ρόλο για να φέρω εις πέρας το παρόν έργο.

Ακόμα θα ήθελα να ευχαριστήσω την υποψήφια διδάκτωρ κα. Χριστιάννα Νεοφύτου ειδική στο τομέα της Βιολογίας, που με τις γνώσεις της και την βοήθεια της με καθοδήγησε για την όσο καλύτερη υλοποίηση του κατάλληλου συστήματος, ως η κύρια χρήστης του.

Επίσης ένα μεγάλο ευχαριστήσω οφείλω στους γονείς μου και στα αδέρφια μου, Βαγγέλη και Κυριάκο, για την αγάπη, την αμέριστη στήριξη και βοήθειά που μου παρείχαν τα τελευταία χρόνια στην φοιτητική μου πορεία, τόσο ηθική όσο και οικονομική.

Τέλος δεν μπορώ να παραλείψω να ευχαριστήσω τους φίλους μου, που με ανέχτηκαν, από τησχόλη και εκτός σχολής. Καθώς συνέβαλαν και αυτοί με τον τρόπο τους στην μέχρι τώρα πορεία μου.

Περίληψη

Το παρόν πόνημα πραγματεύεται στον τομέα της Συστημικής Βιολογία, που αποτελεί επιστημονικό κλάδο της Βιοπληροφορικής και ασχολείται με την απεικόνιση και επεξεργασία βιολογικών δεδομένων, με απώτερο στόχο τη διάγνωση και θεραπεία διαφόρων ασθενειών, όπως ο καρκίνος.

Πιο συγκεκριμένα, το υπό εξέταση αντικείμενο είναι η δημιουργία μιας γραφικής διεπαφής χρήστη η οποία θα είναι συνυφασμένη για τις ανάγκες βιολόγων, οι οποίοι ασχολούνται με τη μελέτη καρκινικών κυττάρων. Θα αυτοματοποιεί μέρος από την διαδικασία των πειράματος που εκτελούν καθημερινά.

Το γραφικό αυτό περιβάλλον θα απεικονίζει βιολογικά δίκτυα (μονοπάτια) σε μορφή γράφου τα οποία σχετίζονται με τις κυτταρικές διεργασίες της απόπτωσης και του πολλαπλασιασμού του κυττάρου. Επιπλέον, χρησιμοποιώντας ως είσοδο πειραματικά δεδομένα από το χρήστη, προβάλλει την διαμόρφωση της απεικόνισης του δικτύου, όπως αυτή θα προέκυπτε μετά από υπολογισμούς που θα πραγματοποιούνται στον επιστημονικό βοηθό, αναφορικά με υποθέσεις που γίνονται για την κατάσταση των γονιδίων μετά από το πείραμα, αλλά και μετά τη διάδοση των διαφόρων σημάτων ανάμεσά τους.

Στην εργασία περιγράφονται οι έρευνες που απαιτήθηκαν για τη δημιουργία της γραφικής διεπαφής και σχετίζονται με το βιολογικό υπόβαθρο του θέματος, αλλά και με τις ήδη υπάρχουσες υλοποιήσεις που επιτελούν τον ίδιο σκοπό. Στη συνέχεια, καταγράφονται και αναλύονται οι απαιτήσεις του συστήματος, που αποκτήθηκαν μέσω ερωτηματολογίου και συναντήσεων με τη βασική χρήστη του συστήματος.

Ακολουθεί μια περιγραφή των διαφόρων τεχνολογιών και εργαλείων που μελετήθηκαν μέχρι να βρεθούν τα κατάλληλα για την ικανοποίηση των αναγκών και απαιτήσεων του χρήστη, ενώ αναλύεται η διαδικασία σχεδιασμού και υλοποίησης του συστήματος.

Τέλος, γίνεται αναφορά στις διαδικασίες ελέγχου και αξιολόγησης του συστήματος, με τα αποτελέσματά τους, ενώ η εργασία ολοκληρώνεται με την εξαγωγή χρήσιμων συμπερασμάτων και την καταγραφή προτάσεων για περαιτέρω έρευνα και ανάπτυξη.

Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή.....	1
1.1 Γενική εισαγωγή.....	1
1.2 Στόχος διπλωματικής εργασίας	2
1.3 Δομή διπλωματικής εργασίας.....	3
Κεφάλαιο 2: Βιολογικό Υπόβαθρο του Προβλήματος	5
2.1 Εισαγωγή στο Πρόβλημα	5
2.2 Βασικές Έννοιες Βιολογίας.....	5
2.2.1 Κύτταρο, Κυτταρικός Κύκλος και Απόπτωση	6
2.2.2 Καρκινογένεση	8
2.2.3 Βιολογικά Δίκτυα.....	9
2.3 Κυτταρικά Δίκτυα στην Πληροφορική	10
2.3.1 Επεξήγηση Απεικόνισης Κυτταρικού Δικτύου.....	10
Κεφάλαιο 3: Μελέτη υπαρχόντων συστημάτων	12
3.1 Σύστημα ApoCelSys.....	12
3.2 Σύστημα BioCT	15
3.3 Η Ιστοσελίδα Cell Signaling	18
3.4 Προσέγγιση Παρούσας Εργασίας	19
Κεφάλαιο 4: Γνώση & Τεχνολογίες που Μελετήθηκαν	22
4.1 Εισαγωγή.....	22
4.2 Σύνολο Τεχνολογιών 1 : GUI σε Java, Graphviz.....	23
4.2.1 Η Γλώσσα Προγραμματισμού Java.....	23
4.2.2 GUI σε Java : το λογισμικό NetBeans IDE	23
4.2.3 Graphviz (Graph Visualization Software).....	24
4.2.4 Graphviz Java API.....	25
4.2.5 Συνδυασμός GUI σε Java και Graphviz.....	26
4.3 Σύνολο Τεχνολογιών 2 : GUI σε Java, GraphStream	26
4.4 Σύνολο Τεχνολογιών 3 : GUI σε Java, Cytoscape, βιβλιοθήκη KEGG	28
4.4.1 GUI σε Java	28
4.4.2 Cytoscape	28
4.4.3 Βάση Δεδομένων KEGG (Kyoto Encyclopedia of Genes and Genomes)	29
4.4.4 Συνδυασμός GUI σε Java και Cytoscape.....	29

4.5 Σύνολο Τεχνολογιών 4 : GUI σε Java, Graphviz, ZGRViewer	30
4.5.1 ZVTM.....	30
4.5.2 ZGRViewer	31
4.5.3 Συνδυασμός GUI σε Java, Graphviz και ZGRViewer	32
Κεφάλαιο 5: Το Σύστημα BioCLC – Biological Cell Cycle	33
5.1 Εισαγωγή.....	33
5.2 Ανάλυση Συστήματος - Απαιτήσεις και προδιαγραφές.....	33
5.3 Μοντέλο Κύκλου Ζωής Λογισμικού Συστήματος.....	37
5.4 Σχεδιασμός Συστήματος	43
5.5 Υλοποίηση Συστήματος.....	44
Κεφάλαιο 6: Έλεγχος – Αξιολόγηση Συστήματος.....	68
6.1 Εισαγωγή.....	68
6.2 Έλεγχος Συστήματος	68
6.3 Αξιολόγηση - Αποτελέσματα.....	69
Κεφάλαιο 7: Συμπεράσματα – Μελλοντική Εργασία.....	70
7.1 Συμπεράσματα.....	70
7.2 Μελλοντική Εργασία	71
Βιβλιογραφία	73
Παράρτημα Β	85
Παράρτημα Γ	86

Κεφάλαιο 1: Εισαγωγή

1.1 Γενική εισαγωγή

1.2 Στόχος διπλωματικής εργασίας

1.3 Δομή διπλωματικής εργασίας

1.1 Γενική εισαγωγή

Η Βιοπληροφορική είναι ένας σύγχρονος επιστημονικός χώρος ο οποίος τα τελευταία χρόνια αναπτύσσεται ραγδαία. Δύσκολα κανείς μπορεί να δώσει ένα ολοκληρωμένο και σαφή ορισμό και αυτό συμβαίνει γιατί απασχολεί πολλούς επιστημονικούς κλάδους, Για παράδειγμα συνενώνει τον κλάδο της Βιολογία, της Χημεία, της Φυσική, της Πληροφορική, των Μαθηματικά και το τομέας της Ιατρική.

Ο Δρ. Temple Smith (καθηγητής του Πανεπιστημίου της Βοστόνης) έχει διατυπώσει ένα πιο ολοκληρωμένο ορισμό ο οποίος αναγράφεται πιο κάτω

« Η επιστήμη της Βιοπληροφορικής είναι συνένωση των επιστημών της Μοριακής Βιολογίας και της Πληροφορικής. Έχει ως αντικείμενο της την βέλτιστη συνεργασία τους για το καλό (ελπίζουμε) της ανθρωπότητας, πρακτικά στην αποθήκευση, επεξεργασία και ανάλυση των βιολογικών δεδομένων ».

Στη Βιοπληροφορική συλλέγονται και οργανώνονται δεδομένα από όλους τους πιο πάνω τομείς. Η ανάπτυξη και η χρήση υπολογιστικών εργαλείων βοηθάει στη ανάλυση και ερμηνεία των βιολογικών και άλλων δεδομένων. Αυτό είναι εφικτό με την συλλογή πειραματικών δεδομένων, την δημιουργία μοντέλων όπου γίνεται

μια εναλλαγή μεταξύ θεωρίας, μοντελοποίησης, προσομοίωσης και πειραμάτων. Όλα τα προηγούμενα έχουν σαν στόχο την πρόγνωση διαφόρων παθήσεων, ασθενειών καθώς και πολλών άλλων ανωμαλιών του ανθρώπινου οργανισμού καθώς και την βελτίωση πολλών μηχανισμών υποστήριξης και βοήθειας όπου υπάρχει ήδη πρόβλημα.[15]

Για χάρη της παρούσας διπλωματικής εργασίας θα χρησιμοποιήσουμε την απαραίτητη γνώση που έχουμε στη διάθεση μας από τα πιο πάνω και θα ασχοληθούμε εκτενέστερα με μια γραφική διεπαφή απεικόνισης και επεξεργασίας κυτταρικών δικτύων/μονοπατιών.

1.2 Στόχος διπλωματικής εργασίας

Στα πλαίσια της παρούσας Διπλωματικής εργασίας, θα γίνει προσπάθεια δημιουργίας εφαρμογής γραφικής διεπαφής απεικόνισης κυτταρικού δικτύου που θα λειτουργεί ως αυτόματος επιστημονικός βοηθός, για πειραματικές πρακτικές στο πεδίο της Βιολογίας των συστημάτων. Ανώτερος στόχος είναι να καταφάουμε να παρατηρήσουμε πως οι πληροφορίες προχωρούν στα μονοπάτια των κυτταρικών δικτύων. Η γραφική διεπαφή, θα πρέπει να διαθέτει τα κατάλληλα χαρακτηριστικά ώστε να είναι εύχρηστη, φιλική προς το χρήστη, εύκολη στην εκμάθηση και στην κατανόηση του περιεχομένου και τελικά να διευκολύνει το Βιολόγο στις έρευνες και στα πειράματα που πραγματοποιεί.

Ταυτόχρονα με τον προαναφερθέντα βασικό στόχο, επιδιώκεται η απόκτηση γνώσης σχετικά με την επιστήμη της Βιοπληροφορικής και κατ' επέκταση στον τομέα της Συστημικής Βιολογίας, που θα συμβάλλει στην κατανόηση των απαιτήσεων του χρήστη – Βιολόγου για τη διεπαφή. Η συνεχής συνεργασία με το χρήστη και η τακτική επικοινωνία θα βοηθήσει σημαντικά στην σωστή εξαγωγή των απαιτήσεων και προδιαγραφών του συστήματος αρχικά, αλλά και στον έλεγχο και αξιολόγησή του στο τέλος.

Επιπλέον στόχος, εξίσου σημαντικός για την επίτευξη του απώτερου σκοπού, είναι η έρευνα και μελέτη διαφόρων τεχνολογιών, εργαλείων και μεθόδων σχεδιασμού, ανάπτυξης και υλοποίησης, για την εξεύρεση του καταλληλότερου συνόλου τεχνολογιών που θα βοηθήσουν στην δημιουργία της διεπαφής.

Εν κατακλείδι, μετά την ολοκλήρωση και αξιολόγηση της διεπαφής, θα πρέπει να μελετηθούν και έπειτα να είναι διαθέσιμες προτάσεις που θα προωθούν τη μελλοντική έρευνα και ανάπτυξης για την περαιτέρω επέκταση της διεπαφής.

1.3 Δομή διπλωματικής εργασίας

Η δομή που θα ακολουθήσει στη παρούσα διπλωματική εργασία, επί της ουσίας συμβαδίζει με την πορεία που ακολουθήθηκε για την ανάπτυξη της γραφικής διεπαφής, και ακολουθεί τον κύκλο ζωής ενός λογισμικού. Στα κεφάλαια που ακολουθούν γίνεται περιγραφή όλων των φάσεων ανάπτυξης.

Στο Κεφάλαιο 2 περιγράφεται η φάση της έρευνας σχετικά με το πρόβλημα. Γίνεται εισαγωγή στο πρόβλημα και σε βασικές έννοιες της Βιολογίας και της Βιοπληροφορικής, για την καλύτερη κατανόηση των απαιτήσεων του συστήματος.

Στο Κεφάλαιο 3 καταγράφονται τα προϋπάρχοντα συστήματα, τα οποία θα διαδεχθεί η νέα υλοποίηση, ενώ δίνεται έμφαση και στην προσέγγιση που θα ακολουθήσει η παρούσα εργασία.

Το Κεφάλαιο 4 αφιερώνεται στην έρευνα που έγινε σχετικά με τα διάφορα εργαλεία και τεχνολογίες, μέχρι την τελική επιλογή του κατάλληλου σύνολο τεχνολογιών που θα χρησιμοποιηθεί.

Το Κεφάλαιο 5 είναι το βασικότερο κεφάλαιο, στο οποίο παρουσιάζεται το μεγαλύτερο μέρος του κύκλου ζωής της ανάπτυξης του συστήματος και αφορά στην καταγραφή και ανάλυση απαιτήσεων (όπως δόθηκαν από τη βασική χρήστη του συστήματος, κα. Χριστιάνα Νεοφύτου), στο σχεδιασμό και στην υλοποίηση του συστήματος.

Το Κεφάλαιο 6 περιλαμβάνει τις τελευταίες φάσεις του κύκλου ζωής για την ολοκλήρωση της διεπαφής, δηλαδή τα στάδια του ελέγχου και της αξιολόγησης του συστήματος.

Τέλος, στο Κεφάλαιο 7 ολοκληρώνεται η εκπόνηση της διπλωματικής εργασίας, με την εξαγωγή συμπερασμάτων, καθώς και προτάσεις για μελλοντική εργασία.

Κεφάλαιο 2: Βιολογικό Υπόβαθρο του Προβλήματος

2.1 Εισαγωγή στο Πρόβλημα	4
2.2 Βασικές Έννοιες Βιολογίας	4
2.3 Κυτταρικά Δίκτυα στην Πληροφορική	9

2.1 Εισαγωγή στο Πρόβλημα

Στο κεφάλαιο αυτό, θα γίνει μια σύντομη παρουσίαση των βασικών εννοιών της Βιολογίας και της Βιοπληροφορικής που σχετίζονται άμεσα με το αντικείμενο της παρούσας διπλωματικής εργασίας, με σκοπό την καλύτερη κατανόηση του υπό εξέταση προβλήματος, δηλαδή της απεικόνισης και επεξεργασίας κυτταρικών δικτύων με σκοπό την ηλεκτρονική επεξεργασία βιολογικών πειραμάτων σε καρκινικά κύτταρα.

Πιο αναλυτικά, ο χρήστης του συστήματος, που εν προκειμένω είναι βιολόγος, πραγματοποιεί πειράματα στο εργαστήριο για τα οποία επιθυμεί να μπορεί να παρακολουθεί και να καταγράφει τα συστατικά και τις διεργασίες ενός κυττάρου, ώστε να μπορεί στη συνέχεια να μελετήσει πώς τα δεδομένα αλλάζουν εξαιτίας των δύο βασικών διεργασιών που σχετίζονται με την καρκινογένεση.

Στην συνέχεια του κεφαλαίου, λοιπόν, αναλύονται συνοπτικά οι όροι της Βιολογίας που θα φανούν χρήσιμοι κατά την ανάλυση των απαιτήσεων του συστήματος, και, μετέπειτα, κατά την υλοποίηση και αξιολόγησή του.

2.2 Βασικές Έννοιες Βιολογίας

Στην ενότητα αυτή, θα γίνει μια σύντομη παρουσίαση ορισμένων βασικών εννοιών της Βιολογίας και πιο συγκεκριμένα, της Συστημικής Βιολογίας, η οποία αποτελεί νέο και καινοτόμο επιστημονικό κλάδο που μελετά γενικότερα το βιολογικό σύστημα, σε αντίθεση με τους κλάδους της Βιοχημείας και της Μοριακής Βιολογίας που

ασχολούνται περισσότερο με χαρακτηριστικά των μερών ενός οργανισμού ή ενός κυττάρου. Εστιάζοντας στη Συστημική Βιολογία, λόγω της φύσης και του αντικειμένου της, απαιτεί την συνεργασία διαφόρων επιστημονικών κλάδων, από τη βιολογία και τη φυσική, ως τα μαθηματικά και την (βιο)πληροφορική.

2.2.1 Κύτταρο, Κυτταρικός Κύκλος και Απόπτωση

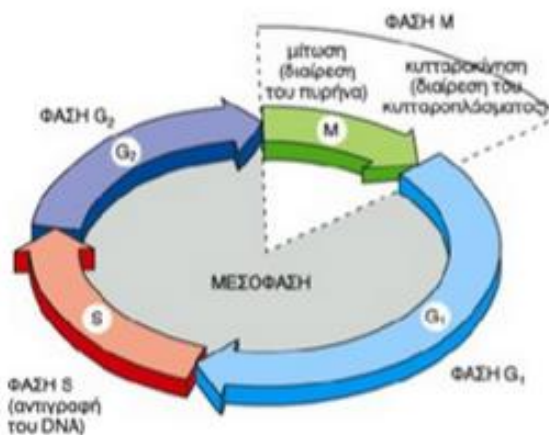
Το κύτταρο αποτελεί τη βάση της δομικής και λειτουργικής μονάδας της ζωής, μια δομή που έχουν όλα τα στοιχεία για να μπορούν να χαρακτηριστούν ζωντανά. Σε κάθε κύτταρο υπάρχει ο πυρήνας, με εξαίρεση τα ερυθρά αιμοσφαίρια που δεν υπάρχει. Εντός του πυρήνα, βρίσκεται το DNA, το οποίο μπορεί να το χαρακτηρίσει κανείς ως ένα τεράστιο μόριο. Το DNA αποτελείται από δομικές μονάδες, όπως τα σάκχαρα και οι αζωτούχες βάσεις, τα οποία ενώνονται το ένα μετά το άλλο και οργανώνονται σε αλληλουχίες. Στην αλληλουχία, κατασκευάζονται οι πρωτεΐνες του κυττάρου.

Το κάθε κύτταρο αποτελείται από διεργασίες, πρωτεΐνες και ουσίες. Δύο από τις σημαντικότερες διεργασίες που συμβαίνουν στο κύτταρο και θα ασχοληθούμε εκτενέστερα στην εργασία αυτή, είναι η διαδικασία της απόπτωσης και του κυτταρικού κύκλου (ή πολλαπλασιασμού).

Ο κυτταρικός κύκλος είναι μια αυστηρά ελεγχόμενη διεργασία που τελικά οδηγεί στη διαίρεση και τον πολλαπλασιασμό του περιεχομένου του κυττάρου. Πρόκειται για τη χρονική περίοδο μεταξύ του τέλους μιας κυτταρικής διαίρεσης και του τέλους της επόμενης και περιλαμβάνει τη μεσόφαση και τη μίτωση. Κατά την διάρκεια του κυτταρικού κύκλου από ένα κύτταρο έχουμε την δημιουργία δύο θυγατρικών κυττάρων.

Ο πολλαπλασιασμός του κυττάρου, αποτελεί μια από τις σημαντικότερες διεργασίες του κυττάρου κατά την οποία αναπαράγονται τα κύτταρα όλων των πολυκύτταρων οργανισμών. Για να διασφαλιστεί η σωστή εξέλιξη του κυτταρικού κύκλου, αναπτύσσονται στα κύτταρα κάποια **δίκτυα ρυθμιστικών πρωτεϊνών**, αποκαλούμενα και ως **συστήματα ελέγχου κυτταρικού κύκλου**. Για να γίνω πιο σαφής, ο κυτταρικός κύκλος αποτελείται από δύο βασικά γεγονότα, την διαίρεση του πυρήνα (μίτωση) και την διαίρεση του κυττάρου (κυτταροκίνηση). Η φάση που περιλαμβάνει τις δύο αυτές διαιρέσεις αποκαλείτε 'Φάση Μ'. Πριν και μετά από την Φάση Μ, έχουμε την

‘Μεσόφαση’, όπου και η Μεσόφαση διαιρείται σε άλλες 3 φάσεις, διαφορετικές φυσικά από τις προηγούμενες. Διαιρείται στη ‘Φάση S’, στη ‘Φάση G1’ και στη ‘Φάση G2’. Η Φάση G1(χάσμα 1^ο) υπάρχει για να δώσει ένα χρονικό περιθώριο στο κύτταρο με διάφορες πολύπλοκες διαδικασίες, να προετοιμασθεί για την έναρξη της επόμενης φάσης, της Φάσης S. Στη Φάση S, γίνεται η σύνθεση του DNA, δηλαδή αντιγράφει με εξαιρετική ακρίβεια τον εαυτό του και δημιουργεί ένα πανομοιότυπο αντίγραφο του πυρήνα. Έπεται η Φάση G2(Χάσμα 2^ο), η φάση αυτή και πάλι υπάρχει για να δώσει ένα χρονικό περιθώριο στο κύτταρο έτσι ώστε να προετοιμασθεί για την επόμενη φάση που είναι η Φάση M όπου το διπλασιασμένο DNA έχει χωρισθεί με ακρίβεια σε δύο ίσα μέρη ώστε να είναι πανομοιότυπο με το αρχικό. Εκτός από αυτές τις φάσεις, έχουμε και τη ‘Φάση G0’, είναι η φάση ηρεμίας όπου κάποιο κύτταρο μετά την ολοκλήρωση του μπορεί αντί να προχωρήσει στη φάση G1 να παραμείνει σε αυτή τη κατάσταση μέχρι να υπάρξουν νέες ανάγκες για τον οργανισμό. Περιληπτικά η σειρά η οποία ακολουθείτε είναι: Φάση G1 → Φάση S → Φάση G2 → Φάση M. [29]



Εικόνα 1 : Ο πολλαπλασιασμός του κυττάρου

Ένα είδος αυτού του τύπου δικτύου ρυθμιστικών πρωτεϊνών είναι οι κυκλινες και στο κέντρο του συστήματος ελέγχου το οποίο περιγράφηκε πιο πάνω βρίσκονται κάποιοι βιολογικοί διακόπτες, που ελέγχουν κύριες διεργασίες του κύκλου, όπως η αντιγραφή του DNA και ο διαχωρισμός χρωμοσωμάτων. Γενικά, το σύστημα ελέγχου κυτταρικού κύκλου ενεργοποιεί με αυστηρό τρόπο τις απαιτούμενες πρωτεΐνες που συμβάλλουν στην ενεργοποίηση κάθε διεργασίας και μόλις τελειώσει η αντίστοιχη διεργασία, πάλι με αυστηρό τρόπο, απενεργοποιεί αυτές τις πρωτεΐνες. [11] [12]

Η φάση του συστήματος ελέγχου, η οποία ελέγχει τις μεταβάσεις από μία φάση σε μια άλλη του κυτταρικού κύκλου χωρίζεται σε έξι φάσεις:

1. Πρόφαση,
2. Προμετάφαση,
3. Μετάφαση,
4. Ανάφαση,
5. Τελόφαση και
6. Κυτταροκίνηση, δηλαδή η διαίρεση του κυττάρου, αρχίζει στην τελευταία φάση της μίτωσης και τελειώνει με το διαχωρισμό των δύο κυττάρων. Είναι η τελευταία φάση του κυτταρικού κύκλου. [11] [12]

Η Απόπτωση είναι μια φυσιολογική διαδικασία που σχετίζεται με τον κυτταρικό θάνατο, όταν αυτός είναι επιθυμητό ή προγραμματισμένο γεγονός ακολουθώντας ένα συγκεκριμένο γονιδιακό πρόγραμμα μέσω σημάτων από το εσωτερικό ή εξωτερικό περιβάλλον του κυττάρου (σε αντίθεση με τη νέκρωση, που ορίζεται ως η μορφή κυτταρικού θανάτου που προκαλείται από απότομες κυτταρικές βλάβες). Είναι εξαιρετικά σημαντική διαδικασία που καθορίζει την ορθή ανάπτυξη των οργανισμών. Μπορεί να χαρακτηριστεί και ως ένα στερεοτυπικό πρόγραμμα κυτταρικής αυτοκαταστροφής («αυτοκτονίας»), με το οποίο οι πολυκύτταροι οργανισμοί απαλλάσσονται από γερασμένα, κατεστραμμένα μολυσμένα ή βλαβερά κύτταρα.

Ένα κύτταρο θεωρείται αποπτωτικό όταν σταματήσει η επικοινωνία του με τα γειτονικά του κύτταρα. Επίσης δεν είναι δυνατή η αντιγραφή, μεταγραφή και επανόρθωση του DNA. Ο μηχανισμός της απόπτωσης είναι πολύπλοκος και δεν έχει αποσαφηνιστεί πλήρως, ενώ χαρακτηρίζεται από βιοχημικές και μορφολογικές αλλαγές στο κύτταρο, όπως η συρρίκνωση της χρωματίνης και η συρρίκνωση του κυττάρου χωρίς διάσπαση της κυτταρικής μεμβράνης. [11] [13]

2.2.2 Καρκινογένεση

Η καρκινογένεση, ένα ζήτημα που απασχολεί τους κλάδους της Βιολογίας και της Ιατρικής, όσο και τομής που συνδέονται με αυτούς τους κλάδους. Επί της ουσίας καρκινογένεση είναι μια σειρά από γενετικές μεταλλάξεις και έπειτα κυτταρική

αναπαραγωγή. Αναφερόμαστε σε γονίδια τα οποία είναι κατασταλτικά στη διαδικασία της απόπτωσης.

Βασικό χαρακτηριστικό της καρκινογένεσης είναι ο πολλαπλασιασμός καρκινικών κυττάρων. Αυτό το είδος πολλαπλασιασμού αποτελείται από κύτταρα τα οποία υπό κανονικές συνθήκες θα έπρεπε να οδηγούνται (σε κάποια φάση) στην απόπτωση, θα έπρεπε να πεθαίνουν με αυστηρά ελεγχόμενο τρόπο όπως και τα υπόλοιπα κύτταρα, κάτι το οποίο δεν συμβαίνει ποτέ. Εν αντίθεση, μόνο πολλαπλασιάζονται και δημιουργούν με αυτό το τρόπο τους όγκους ή κοινός το καρκίνο. Δηλαδή, σωρεία από καρκινικά κύτταρα. Τα καρκινικά κύτταρα χαρακτηρίζονται ως ανθεκτικά στο μηχανισμό γήρανσης, στην απόπτωση όπως αναφέρθηκε πιο πάνω καθώς και στο να κάνουν μεταστάσεις. Επιπρόσθετα, δεν φτάνουν ποτέ σήματα τα οποία να διακόπτουν τουλάχιστον τον πολλαπλασιασμό τους. Αυτό συμβαίνει συνήθως από λάθη κατά την φάση της μίτωσης στο κύτταρο.

Ένα από τα αντικείμενα επισταμένης μελέτης της Βιολογίας είναι οι ιδιότητες των καρκινικών κυττάρων, για την διερεύνηση των οποίων οι Βιολόγοι πραγματοποιούν πολλά πειράματα, ακόμα και σε πραγματικά καρκινικά κύτταρα, στα οποία εισάγουν διάφορες ουσίες, με σκοπό να καταγράψουν την αντίδραση των κυττάρων σε αυτές και να εξετάσουν πιθανές λύσεις, ώστε τα καρκινικά κύτταρα να επανέλθουν σε φυσιολογική κατάσταση.

2.2.3 Βιολογικά Δίκτυα

Σημαντικά βιολογικά δίκτυα αναπαριστούν κυτταρικά συστήματα με κόμβους και ακμές. Οι κόμβοι σχετίζονται με μόρια όπως είναι οι πρωτεΐνες, το RNA και τα γονίδια, σχετίζονται επίσης και με τους μεταβολίτες. Οι ακμές ή σύνδεσμοι μεταξύ των κόμβων αντιπροσωπεύουν διάφορες αλληλεπιδράσεις μεταξύ των κόμβων. Τα δίκτυα γονιδιακής ρύθμισης και μετάδοσης σημάτων περιγράφουν πώς τα γονίδια ενεργοποιούνται ή εμποδίζονται και, επομένως, ποιες πρωτεΐνες παράγονται σε ένα κύτταρο σε μια συγκεκριμένη χρονική στιγμή. Τα δίκτυα αυτά, εάν τύχουν σωστής αξιοποιήσεις και γίνει η κατάλληλη και δομημένη αναπαράσταση των αλληλεπιδράσεων, είναι εφικτό τότε να χρησιμοποιηθούν ως μοντέλα πρόβλεψης μέσω

της θεωρίας των δικτύων. Τα βιολογικά δίκτυα μπορεί να είναι: Πρωτεϊνικά Δίκτυα(Protein – Protein interaction network), Μεταγραφικά δίκτυα(Protein – DNA Interaction Network), Μεταβολικά δίκτυα(Metabolomics Network), Νευρωνικά Δίκτυα(Neuron Synaptic Connection Network) είτε Δίκτυο Τροφικών Αλυσίδων Οικοσυστήματος(Ecological Food Web).

2.3 Κυτταρικά Δίκτυα στην Πληροφορική

Τα υπό εξέταση κυτταρικά δίκτυα ακολουθούν τα χαρακτηριστικά των Δικτύων Πρωτεΐνης – Πρωτεΐνης, η απεικόνιση των οποίων αποσκοπεί στην κατανόηση της συνολικής δομής των αλληλεπιδράσεων μεταξύ των πρωτεϊνών αλλά και ανάμεσα στις πρωτεΐνες και άλλες ουσίες.

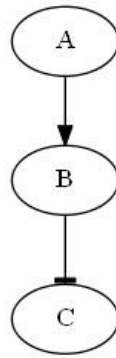
Τα δίκτυα αυτά είναι εγγενώς πολύπλοκα : μεγάλα, μη επίπεδα με πολλές διασταυρώσεις ακμών, πολλά ξεχωριστά συστατικά και κόμβους μεγάλου εύρους βαθμών. Κατά συνέπεια, οι κύριες απαιτήσεις απεικόνισης αφορούν τα κοινά αισθητικά κριτήρια για απεικονίσεις γράφων, όπως η ισοκατανομή κόμβων, η συμμετρία, τα ομοιόμορφα μήκη ακμών, ή οι Ευκλείδειες αποστάσεις που αντικατοπτρίζουν θεωρητικές αποστάσεις γράφων. [16]

2.3.1 Επεξήγηση Απεικόνισης Κυτταρικού Δικτύου

Ένα κυτταρικό δίκτυο αποτελείται από πρωτεΐνες, διεργασίες και διάφορες άλλες ουσίες, οι οποίες μπορούν να υπάρξουν σε τρεις καταστάσεις : ενεργή (active), απενεργοποιημένη (inactive), ουδέτερη (neutral).

Για την καλύτερη κατανόηση της απεικόνισης του κυττάρου, δίνεται το ακόλουθο παράδειγμα.

Έστω ότι υπάρχουν τρεις πρωτεΐνες, οι A, B και C.



Εικόνα 2 : Παράδειγμα απεικόνισης τριών πρωτεϊνών ενός κυτταρικού δικτύου

Για την απεικόνιση ενός κυτταρικού δικτύου, ορίζονται δύο είδη ακμών : η ενεργητική κατευθυνόμενη ακμή (activator), όπως φαίνεται από την πρωτεΐνη A στη B, και η ανασταλτική κατευθυνόμενη ακμή (inhibitor), όπως φαίνεται από τη B στη C.

Όταν δύο πρωτεΐνες συνδέονται με ενεργητική ακμή, τότε όποια κι αν είναι η κατάσταση της πρωτεΐνης από την οποία προέρχεται η ακμή, αυτή μεταφέρεται στην πρωτεΐνη στην οποία κατευθύνεται η ακμή. Στο παράδειγμα δηλαδή, αν η πρωτεΐνη A είναι ενεργή τότε θα είναι και η B και αν η A είναι απενεργοποιημένη, τότε θα είναι και η B.

Όταν δύο πρωτεΐνες συνδέονται με ανασταλτική ακμή και η πρωτεΐνη από την οποία προέρχεται η ακμή είναι ενεργή, τότε αυτόματα απενεργοποιείται η πρωτεΐνη στην οποία κατευθύνεται η ακμή. Στο παράδειγμα, αν η B είναι ενεργή τότε η C απενεργοποιείται. Αν η B είναι απενεργοποιημένη, τότε δε μπορεί να απενεργοποιήσει τη C κι έτσι μένει στην αρχική της κατάσταση, όποια κι αν ήταν αυτή, είτε παίρνει την κατάστασή της από άλλες ακμές που κατευθύνονται σε αυτή.

Στην περίπτωση που κατευθύνονται περισσότερες από μία ακμές σε μία πρωτεΐνη, τότε οι ακμές με το ανασταλτικό σήμα κυριαρχούν αυτών με το ενεργητικό.

Κεφάλαιο 3: Μελέτη υπαρχόντων συστημάτων

3.1 Σύστημα ApoCelSys	11
3.2 Σύστημα BioCT	14
3.3 Ιστοσελίδα Cell Signaling	18
3.4 Τίτλος τέταρτου υποκεφαλαίου (***)	19

3.1 Σύστημα ApoCelSys

Το σύστημα ApoCelSys δημιουργήθηκε στα πλαίσια της μεταπτυχιακής διατριβής του κύριου Σωτήρη Λαζάρου, το 2010 στο Πανεπιστήμιο Κύπρου. Είναι η πρώτη έκδοση του συστήματος το οποίο χρησιμοποιούσε ο χρήστης μας για ένα χρονικό διάστημα και χρειάζοταν αναβάθμιση. Η αναβάθμιση έγινε το 2014 και δημιουργήθηκε η δεύτερη έκδοση του συστήματος, το BioCT το οποίο θα αναλύσουμε εκτενέστερα στο αμέσως επόμενο υπο-κεφάλαιο (Κεφάλαιο 3.2).[3]

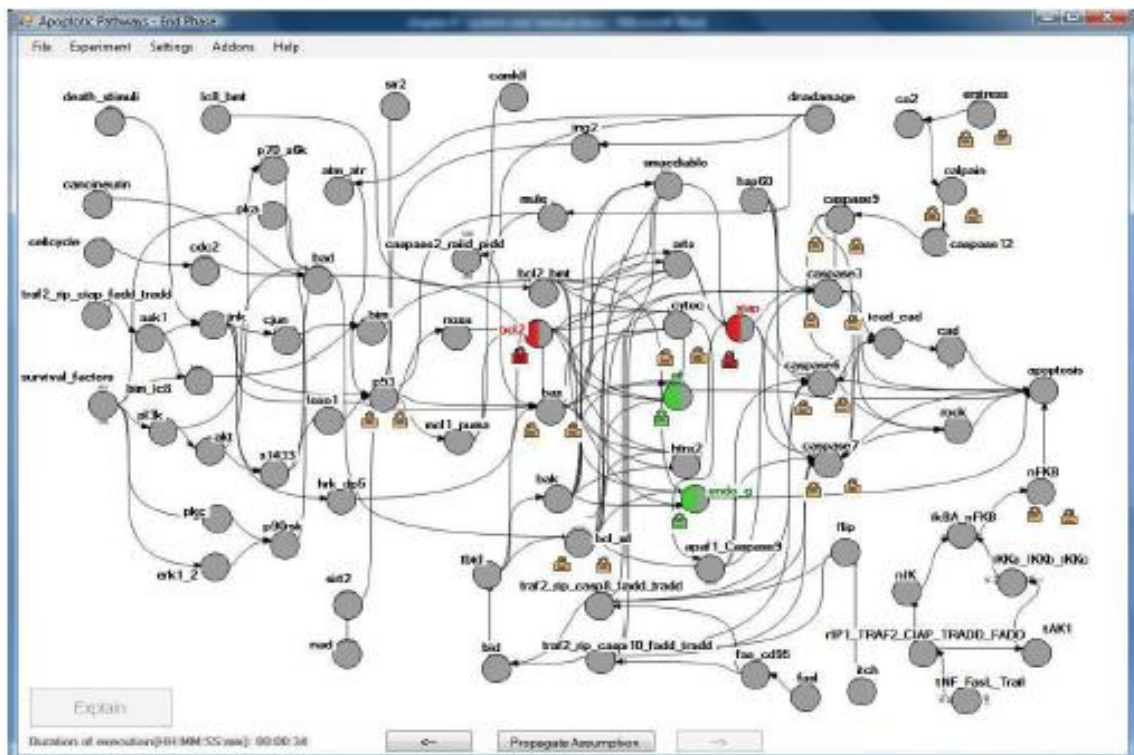
Το ApoCelSys, δημιουργεί βασικές υποθέσεις σε προσομοιωμένα κύτταρα, κάνει απεικόνιση μονοπατιών και ο βιολόγος μπορεί να παρατηρήσει την επίδραση του κάθε φαρμάκου στο κύτταρο. Η απεικόνιση των κυττάρων αυτών διαθέτει μόνο τη βασική λειτουργία της απόπτωσης. Έπειτα ο χρήστης έπαιρνε κάποιες παρατηρήσεις η οποίες ήταν αποτέλεσμα του Απαγωγικού Λογικού Προγραμματισμού (Abductive Logic Program - APL) και της εμπειρίας του βιολόγου. [3]

Στη φάση αυτή θα ήταν καλό να εξηγήσουμε την έννοια του Απαγωγικού Λογικού Προγραμματισμού - ΑΛΠ (Abductive Logic Program - APL). Για τον ΑΛΠ, δεν υπάρχει σαφής ορισμός, εντούτοις εξηγείται ως το πλαίσιο υψηλού επιπέδου αναπαράστασης γνώσης το οποίο μας επιτρέπει να λύσουμε δηλωτικά προβλήματα (declaratively) με απαγωγικό συλλογισμό. Θεωρείτε μια πιο εξειδικευμένη μορφή του Λογικού Προγραμματισμού που μας επιτρέπει να χρησιμοποιούμε κατηγορήματα τα οποία δεν είναι πλήρως ορισμένα – πιο αφαιρετικές υποθέσεις- τα κατηγορήματα αυτά ορίζονται ως *abdacible*. Προβλήματα τα οποία απασχολούν τον απαγωγικό λογικό προγραμματισμό είναι

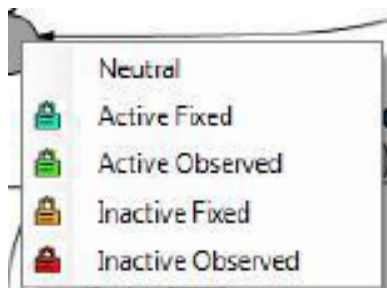
προβλήματα τα οποία τυγχάνουν παρατηρήσει και έπειτα επεξηγήσεις, ή προβλήματα στόχου. Τομείς που κατά κύριο λόγο χρησιμοποιούν τον Απαγωγικό Λογικό Προγραμματισμό είναι προβλήματα διάγνωσης, ή Φυσικής Γλώσσας, ή σχεδιασμού(Planning), ή Μηχανικής Μάθησης.

Οι λειτουργίες οι οποίες μας παρείχε το ApoCelSys ήταν οι εξής: Αρχικά, ο βιολόγος εισάγει τα δεδομένα που θέλει να μελετήσει στον κάθε κόμβο/συστατικό μέλος του κυττάρου, τον ενεργοποιεί ή τον απενεργοποιεί αναλόγως του πειράματος του. Αυτό το επιτυγχάνει κάνοντας δεξί κλικ στο κόμβο/συστατικό μέλος του κυττάρου ή διπλό αριστερό κλικ. Ακολούθως, στο σύστημα υπολογίζονται κάποια συμπεράσματα που είναι αποτέλεσμα υποθέσεων ή ρητών επεξηγήσεων με την εφαρμογή του Απαγωγικού λογικού προγραμματισμού. Τέλος, ο χρήστης βλέπει σε ποια στοιχεία έχει επιδράσει το φάρμακο καθώς και σε ποιους συγκεκριμένους κόμβους/ συστατικά μέλη.

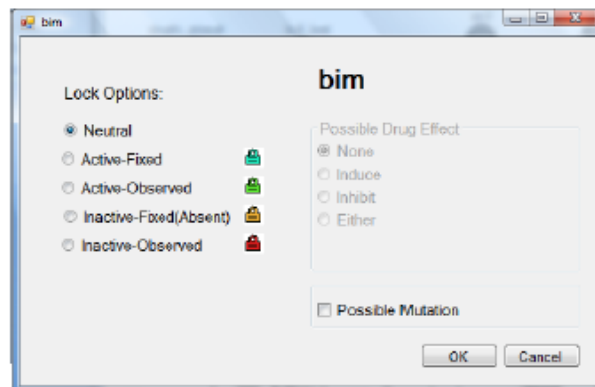
Πιο κάτω παραθέτοντας κάποια στιγμιότυπα(screenshots) του συστήματος για την πλήρη κατανόηση της διαδικασίας των πειραμάτων που εκτελεί ο Βιολόγος από την αρχή μέχρι το τελικό αποτέλεσμα που παίρνει μετά τη διάδοση σήματος(Propagation).



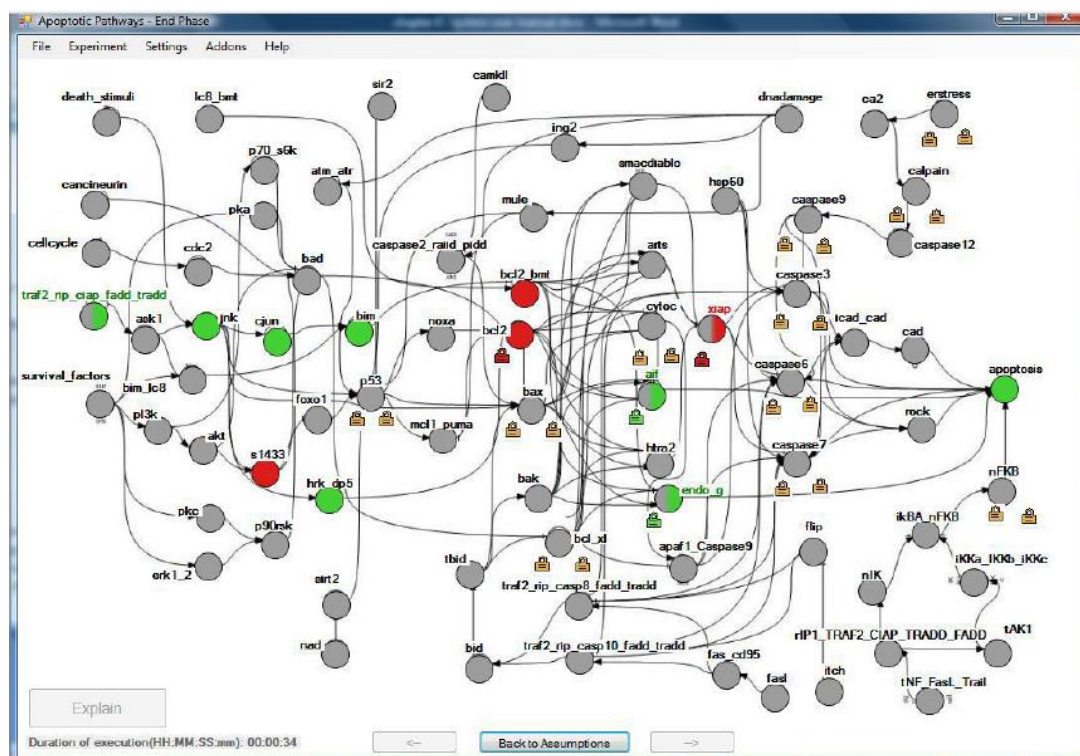
Εικόνα 3: Αρχική Αναπαράσταση Μονοπατιού και σχετικών μενού επιλογών του χρήστη



Εικόνα 4: Στιγμιότυπο σύντομου μενού επιλογών για την κατάσταση ενός γονιδίου (εμφανίζεται με δεξί κλικ στο γονίδιο)



Εικόνα 5: Στιγμιότυπο λεπτομερές μενού επιλογών για την κατάσταση ενός γονιδίου (εμφανίζεται με διπλό κλικ στο γονίδιο)



Εικόνα 6: Στιγμιότυπο του ApoCelSys μετά τις επιλογές που έχει δώσει ο χρήστης, δηλαδή μετά την διάδοση σήματος(Propagation)

3.2 Σύστημα BioCT

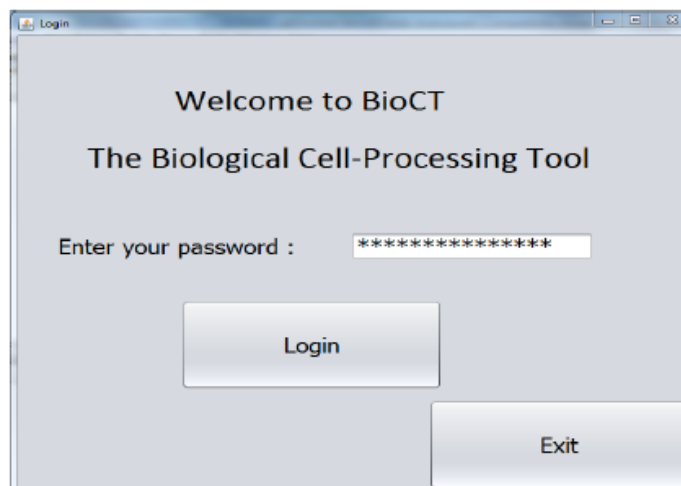
Το σύστημα BioCT δημιουργήθηκε στα πλαίσια της διπλωματικής εργασίας της απόφοιτης φοιτήτριας Ειρήνης Ανδρέου του τμήματος Πληροφορικής, Πανεπιστημίου Κύπρου ως αναβάθμιση του συστήματος ApoCelSys που περιγράφηκε πιο πάνω (Κεφάλαιο 3.1). Το BioCT είναι το σύστημα το οποίο χρησιμοποιούσε ο χρήστης μας, πριν αναβαθμιστεί στο σύστημα το οποίο θα παραδοθεί με την ολοκλήρωση της παρούσας διπλωματικής εργασίας.

Στο BioCT έχουν πραγματοποιηθεί αρκετές από τις απαιτήσεις του χρήστη όπως αναφέρονται στο κεφάλαιο 5.2 και η λειτουργικότητά του ήταν σε επαρκές επίπεδο.

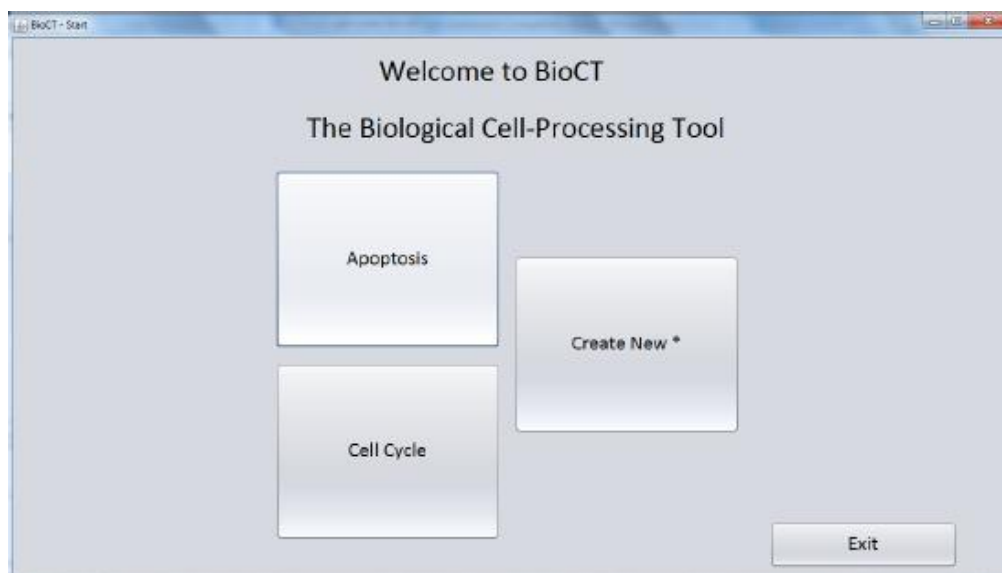
Ουσιαστικά, ο τρόπος αντιμετώπισης από μεριάς του δημιουργού του συστήματος, σε γενικές γραμμές ήταν να δημιουργήσει μια γραφική διεπαφή με δύο βασικές οθόνες, οι οποίες αναπαριστούν το βιολογικό δίκτυο της απόπτωσης και του πολλαπλασιασμού. Έπειτα γινόταν και η εισαγωγή των δεδομένων στις διάφορες ουσίες, πρωτεΐνες ή διεργασίες.

Οι λειτουργίες οι οποίες μας παρείχε το BioCT ήταν οι εξής: Αρχικά, ο βιολόγος κάνει είσοδο στο σύστημα με τον μυστικό κωδικό πρόσβασης. Έπειτα επιλέγει με ποίο από τα δύο δίκτυα θέλει να εκτελέσει πείραμα και εισάγει τα δεδομένα που θέλει να μελετήσει στον κάθε κόμβο/συστατικό μέλος του κυττάρου. Οι επιλογές που έχει είναι να ενεργοποιεί ή να απενεργοποιεί τα συστατικά μέλη αναλόγως του πειράματος του. Το σημείο αυτό πραγματοποιείτε κάνοντας δεξί κλικ στο κόμβο/συστατικό μέλος του κυττάρου ή διπλό αριστερό κλικ. Ακολούθως, στο σύστημα υπολογίζονται κάποια συμπεράσματα που είναι αποτέλεσμα ρητών δηλώσεων μέσα από τον κώδικά. Τέλος, ο χρήστης βλέπει σε ποια στοιχεία έχει επιδράσει το φάρμακο καθώς και σε ποιους συγκεκριμένους κόμβους/συστατικά μέλη.

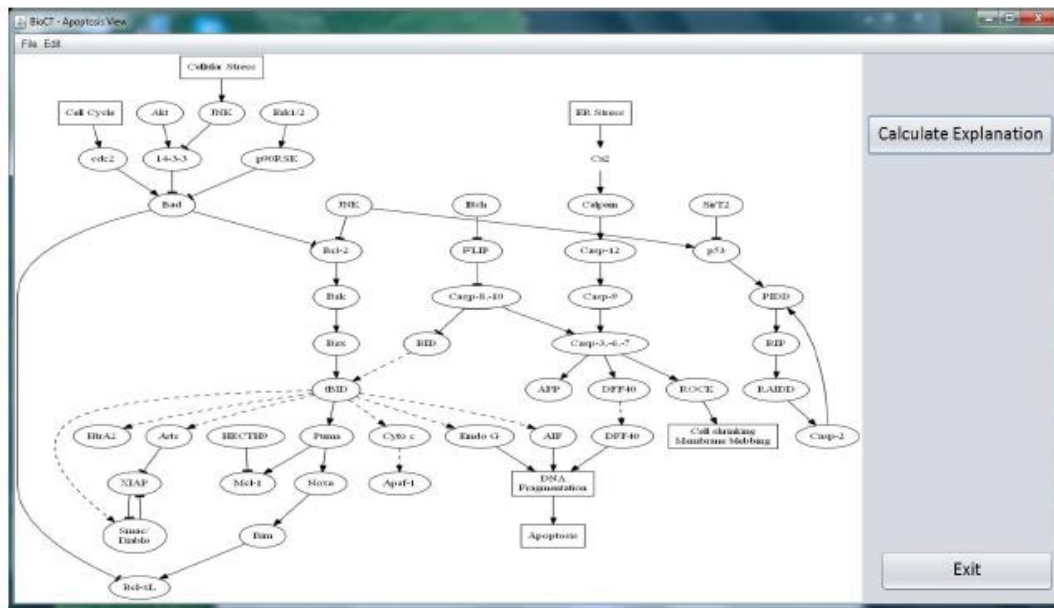
Πιο κάτω παραθέτοντας κάποια στιγμιότυπα (screenshots) του συστήματος για την πλήρη κατανόηση της διαδικασίας των πειραμάτων που εκτελεί ο Βιολόγος από την αρχή μέχρι το τελικό αποτέλεσμα που παίρνει μετά τη διάδοση σήματος (Propagation).



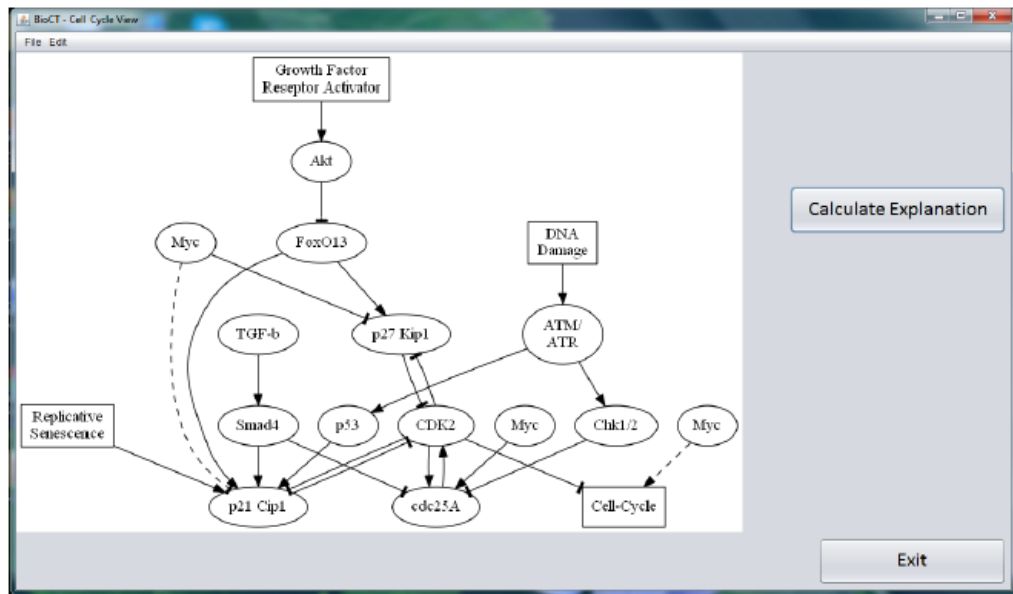
Εικόνα 7: Είσοδος στο σύστημα



Εικόνα 8: Αρχική Οθόνη Συστήματος



Εικόνα 9: Οθόνη Αναπαράστασης του Κυτταρικού Δικτύου Απόπτωσης



Εικόνα 10: Οθόνη Αναπαράστασης του Κυτταρικού Δικτύου Πολλαπλασιασμού

Ο τρόπος αντιμετώπισης του συστήματος οδήγησε την τελική του μορφή σε ένα στατικό περιβάλλον. Πιο κάτω θα μελετηθούν οι παράγοντες οι οποίοι συνέβαλλαν στην στατικότητα του κώδικα της υλοποίησης αυτής.

Αρχικά, αφού μελετήσαμε εις βάθος την όλη μελέτη και πορεία που ακολουθήθηκε για αυτή την έκδοση, παρατηρήσαμε σημεία που πρέπει να βελτιωθούν και να ολοκληρωθούν για το καλύτερο δυνατό αποτέλεσμα.

Παρατήρηση 1:

Στο πηγαίο κώδικα των δύο κλάσεων που ορίζουν τις διεργασίες τις Απόπτωσης και του Πολλαπλασιασμού (Apoptosis.java και CellCycle.java), όλες οι διαδρομές αποθήκευσης των resources, input και output είναι απόλυτες (Absolute Paths) και αναφέρονται σε συγκεκριμένο χρήστη(user) ενός H/Y.

Παρατήρηση 2:

Συνεχίζοντας πρέπει να αναφερθεί ότι ο χρήστης και η λειτουργικότητα του συστήματος περιορίζετε σε λειτουργικό Windows.

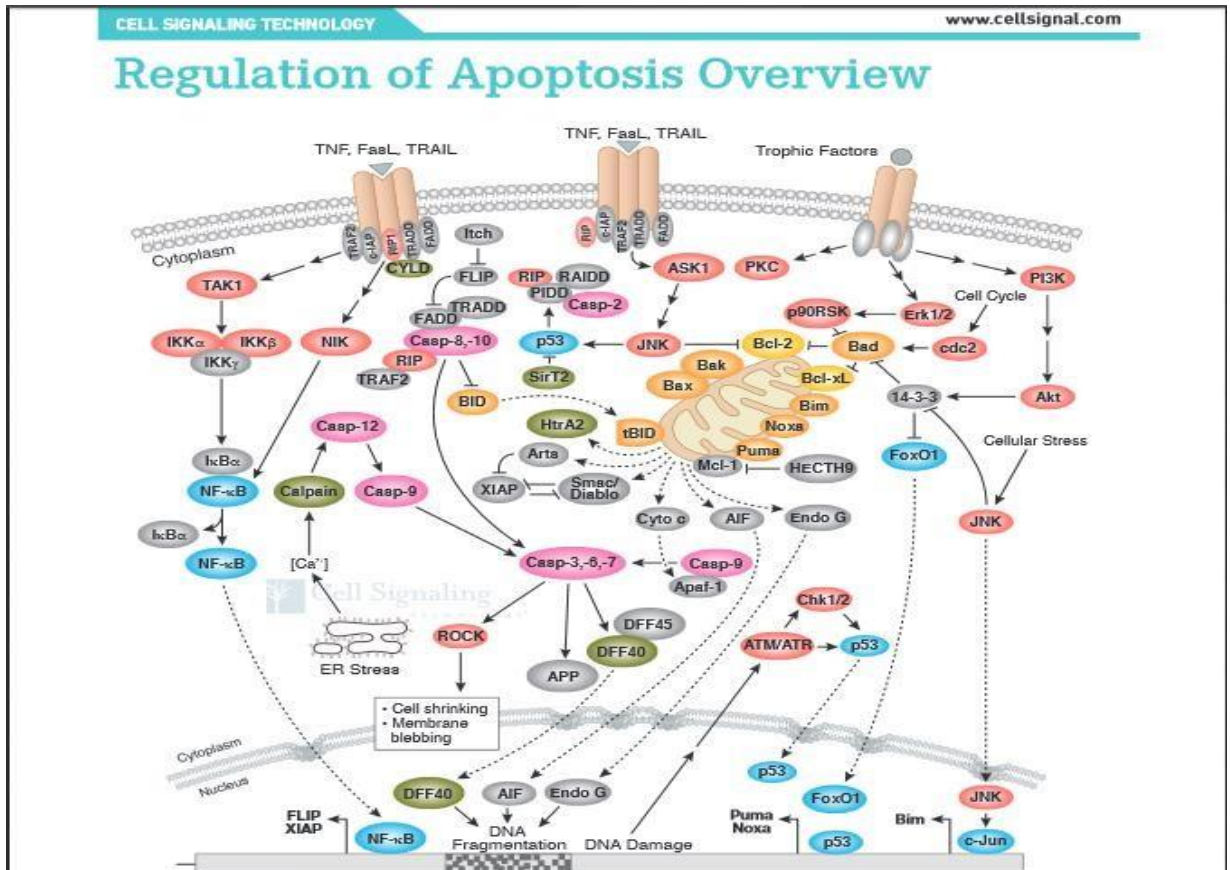
Κατά συνέπεια από την Παρατήρηση 1 & 2, είναι φανερό ότι περιορίζεται, η ανεξαρτησία της διεπαφής από τον H/Y στον οποίο λειτουργεί.

Παρατήρηση 3:

Για την δημιουργία του δικτύου αναπαράστασης χρησιμοποιήθηκε το εργαλείο GraphViz το οποίο αλλάζει δυναμικά τις συντεταγμένες των κόμβων με κάθε προσθήκη κόμβου και τις αποθηκεύει σε αρχείο κειμένου, επομένως μπορούν μεν να ανακτηθούν, αλλά λόγω της διαφοράς scaling των παραγόμενων εικόνων και του frame του συστήματος, οι συντεταγμένες αυτές δεν συμπίπτουν. Σε αυτή την υλοποίηση, δεν ήταν δυνατή η εύρεση κάποιου αλγορίθμου για την αποτελεσματική συσχέτιση των δύο ειδών συντεταγμένων και γι' αυτό το λόγο δεν υλοποιήθηκε ούτε η προσθαφαίρεση κόμβων ούτε το zoom.

3.3 Η Ιστοσελίδα Cell Signaling

Η Cell Signaling είναι μια ιστοσελίδα η οποία μεταξύ άλλων υπηρεσιών προς Βιολόγους, παρέχει και απεικονίσεις διαφόρων διεργασιών – μονοπατιών - των κυττάρων. Ανάμεσα σε αυτές τις απεικονίσεις είναι και οι διεργασίες που θα μελετήσουμε στην παρόν εργασία, της απόπτωσης και του πολλαπλασιασμού. Οι απεικονίσεις αυτές όμως, παρόλο που είναι πολύ χρήσιμες στους Βιολόγους, δεν παρέχουν καμία σημασιολογική έννοια και δυνατότητα επεξεργασίας των μονοπατιών μετά από πειράματα. [2]



Εικόνα 11:Απεικόνιση Δικτύου Απόπτωσης από τη σελίδα της Cell Signaling [2]

3.4 Προσέγγιση Παρούσας Εργασίας

Ουσιαστικά, η υλοποίηση που αποτελεί αντικείμενο της παρούσας εργασίας, αποτελεί απόγονο των δυο προαναφερθέντων συστημάτων, ApoCelSys και BioCT, τα οποία δημιουργήθηκαν από φοιτητές του Πανεπιστημίου Κύπρου, για τις ανάγκες των εργαστηρίων του Τμήματος Βιολογίας.

Πιο συγκεκριμένα, η τρέχουσα υλοποίηση, BioCLC, αναλυτική παρουσίαση της οποίας θα ακολουθήσει σε επόμενα κεφάλαια, βασίστηκε στο προηγούμενο σύστημα BioCT. Μελετήθηκαν αναλυτικά τα χαρακτηριστικά του, τόσο σε επίπεδο σχεδιασμού όσο και υλοποίησης, εντοπίστηκαν ορισμένες αδυναμίες και ελλείψεις και έγινε προσπάθεια βελτίωσης και αναβάθμισης του συστήματος, όπου κατέστη αυτό δυνατό, ενώ καταγράφηκαν και τα σημεία στα οποία απαιτείται περαιτέρω εξέλιξη.

Αναφορικά, με τις αδυναμίες που εντοπίστηκαν στο BioCT, αρχικά, στον πηγαίο κώδικα των δύο κλάσεων που ορίζουν τις διεργασίες τις Απόπτωσης και του Πολλαπλασιασμού (Apoptosis.java και CellCycle.java), όλες οι διαδρομές αποθήκευσης των resources, input και output είναι απόλυτες (Absolute Paths) και αναφέρονται σε συγκεκριμένο χρήστη (user) ενός H/Y, με αποτέλεσμα τον περιορισμό στα χαρακτηριστικά του συστήματος στο οποίο θα εκτελείται η εφαρμογή και, κατ'επέκταση, της ανεξαρτησία της διεπαφής από τον H/Y στον οποίο λειτουργεί. Μια απλή λύση για αυτό είναι να αντικατασταθεί το absolute path (π.χ. C:\\Users\\user\\BioCT\\) με το working directory το οποίο βρίσκεται με την ακόλουθη γραμμή κώδικα :

```
String workDir = System.getProperty("user.dir");
```

Με βάση την παραπάνω αλλαγή μπορούν να οριστούν τα διάφορα αρχεία εισόδου/εξόδου(io), όπως για παράδειγμα :

```
String out4File = workDir + "out4.txt";
```

```
...
```

```
FileReader f;
```

```
f = new FileReader(out4File);
```

ενώ αντίστοιχα μπορεί να οριστεί μια πιο συγκεκριμένη δομή στην οργάνωση των αρχείων, με υποφακέλους για την αποθήκευση των αρχείων io, images κλπ, όπως για παράδειγμα :

```
String outputDir = workDir + "Output\\";
```

```
String out4File = outputDir + "out4.txt";
```

Επιπλέον, για την δημιουργία του δικτύου αναπαράστασης χρησιμοποιήθηκε το εργαλείο GraphViz το οποίο αλλάζει δυναμικά τις συντεταγμένες των κόμβων με κάθε

προσθήκη κόμβου και τις αποθηκεύει σε αρχείο κειμένου, επομένως μπορούν μεν να ανακτηθούν, αλλά λόγω της διαφοράς scaling των παραγόμενων εικόνων και του frame του συστήματος, οι συντεταγμένες αυτές δεν συμπίπτουν. Σε αυτή την υλοποίηση, δεν ήταν δυνατή η εύρεση κάποιου αλγορίθμου για την αποτελεσματική συσχέτιση των δύο ειδών συντεταγμένων και γι' αυτό το λόγο δεν υλοποιήθηκε ούτε η προσθαφαίρεση κόμβων ούτε το zoom.

Συμπερασματικά, η έρευνα για αυτή την έκδοση του υπό μελέτη συστήματος για την επίλυση των προβλημάτων καταλήγει σε δυο προτεινόμενες λύσεις. Είτε πρέπει να γίνει περαιτέρω προσπάθεια εύρεσης κάποιου σχετικού αλγορίθμου που να λύνει το πρόβλημα αυτό, είτε θα απαιτηθεί η εύρεση εναλλακτικών βιβλιοθηκών (libraries) και APIs για τη δυναμική διαχείριση ή/και απεικόνιση γράφων στη Java.

Κεφάλαιο 4: Γνώση & Τεχνολογίες που Μελετήθηκαν

4.1 Εισαγωγή	22
4.2 Σύνολο Τεχνολογιών 1: GUI σε Java, Graphviz	23
4.3 Σύνολο Τεχνολογιών 2: GUI σε Java, GraphStream	26
4.4 Σύνολο Τεχνολογιών 3: GUI σε Java, Cytoscape, βιβλιοθήκη KEGG	28
4.4 Σύνολο Τεχνολογιών 4: GUI σε Java, Graphviz, ZGRViewer	29

4.1 Εισαγωγή

Κατά τη διάρκεια εκπόνησης της εργασίας, έγινε μια σημαντική προσπάθεια μελέτης ενός όσο το δυνατόν πιο διευρυμένου φάσματος εργαλείων, τεχνολογιών, γλωσσών προγραμματισμού και τεχνικών που αφορούν, αφενός σε περιβάλλοντα σχεδιασμού γραφικών διεπαφών (GUI) και αφετέρου σε συστήματα δημιουργίας, διαχείρισης και απεικόνισης γράφων, που χρησιμοποιούνται στην περίπτωση των υπό μελέτη κυτταρικών δικτύων.

Με δεδομένη την πρόθεση αναβάθμισης του ήδη υπάρχοντος συστήματος BioCT, το οποίο είχε υλοποιηθεί με χρήση Java και GraphViz, όπως περιγράφεται και στις ακόλουθες ενότητες, η μελέτη εστίασε περισσότερο σε αυτά τα εργαλεία. Σαν εναλλακτική λύση του GraphViz, μελετήθηκε και η βιβλιοθήκη GraphStream, η χρήση της οποίας όμως δεν κατέστη δυνατή λόγω περισσότερων απαιτήσεων σε τεχνολογίες με τις οποίες δεν υπήρχε ιδιαίτερη εξοικείωση. Ακόμη μελετήθηκε μια εκ νέου προσέγγιση με εντελώς διαφορετικά εργαλεία, η οποία αργότερα απορρίφθηκε και αυτή. Τελικά, για να ξεπεραστούν ορισμένες αδυναμίες του υπάρχοντος συστήματος, ειδικά αναφορικά με το zoom, βρέθηκε η λύση του ZGRViewer, που κάνει με επιτυχία scaling των γράφων, ενώ παρέχει και αρκετές ακόμα δυνατότητες, που καταγράφονται στη συνέχεια.

4.2 Σύνολο Τεχνολογιών 1 : GUI σε Java, Graphviz

Το σύνολο τεχνολογιών που θα περιγραφεί σε αυτό το υποκεφάλαιο, είναι το σύνολο που χρησιμοποιήθηκε για την ανάπτυξη του προηγούμενου συστήματος, BioCT, και το οποίο αποτέλεσε την αφετηρία και για την υλοποίηση της παρούσας εργασίας.

4.2.1 Η Γλώσσα Προγραμματισμού Java

Η Java, μια από τις πιο διαδεδομένες γλώσσες προγραμματισμού, σχεδιάστηκε στην εταιρία Sun Microsystems (η οποία συγχωνεύτηκε με την Oracle Corporation) και η πρώτη επίσημη παρουσίαση της έγινε αρχικά το 1996 με ευρεία αποδοχή. Η αρχική ιδέα της προήλθε από τον James Gosling.

Η Java έχει έντονο αντικειμενοστρεφή χαρακτήρα (object-oriented), βασισμένη σε κλάσεις (class-based), είναι σχετικά απλή, αφού η διαχείριση της μνήμης γίνεται από την ίδια την γλώσσα και απλούστερη αφού έχει επιτευχθεί η εξάλειψη των δεικτών(pointers) που ταλάνιζαν προηγούμενες γλώσσες προγραμματισμού. Επιπρόσθετα, προσφέρει φορητότητα, κάτι το οποίο την κάνει πιο ισχυρή, με τον όρο φορητότητα εννοούμε ότι, ένα πρόγραμμα μπορεί να γραφτεί και να μεταγλωττιστεί π.χ. σε λειτουργικό Windows και στη συνέχεια το ίδιο αρχείο να μπορεί να τρέξει και σε λειτουργικό Unix, "Write Once Run Anywhere - WORA". Επίσης κάνει αυστηρό έλεγχο τύπων (strongly typed), είναι σχεδιασμένη για να δημιουργεί όσο το δυνατό λιγότερες εξαρτήσεις υλοποίησης, είναι μεταγλωττιζόμενη(compiled) και ερμηνευόμενη (interpreted). Αυτά είναι μερικά από τα κύρια χαρακτηριστικά της Java τα οποία χαρίζουν μεγάλη ευελιξία στο προγραμματιστή [30]

4.2.2 GUI σε Java : το λογισμικό NetBeans IDE

Η Java υποστηρίζει πολυάριθμες βιβλιοθήκες, δύο εκ των οποίων είναι οι Swing και AWT που υποστηρίζουν σχεδιασμό και υλοποίηση γραφικών διεπαφών χρήστη. [19]

Όσον αφορά τα περιβάλλοντα ανάπτυξης Java εφαρμογών, είναι διαθέσιμο ένα μεγάλο εύρος εργαλείων, όπως τα NetBeans IDE, Eclipse IDE, IntelliJ IDEA κλπ. Το λογισμικό NetBeans IDE(- integrated development environment) είναι ένα εργαλείο για προγραμματιστές, στο οποίο μπορούν να γράψουν, να κάνουν compile, debug και να

αναπτύξουν ένα πρόγραμμα. Είναι γραμμένο σε γλώσσα προγραμματισμού Java αλλά έχει την δυνατότητα υποστήριξης όλων των γλωσσών προγραμματισμού. Επιπλέον, κάτι πολύ χρήσιμο θεωρείτε το γεγονός πως έχει ένα μεγάλο αριθμός υπομονάδων (modules) που βοηθάνε στην επέκταση της λειτουργικότητας του NetBeans IDE.

Μέχρι στιγμής θεωρείτε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για εφαρμογές σε Java. Πολύ εύκολα μπορούν να σχεδιαστούν και να υλοποιηθούν γραφικές διεπαφές χρήστη (Graphical User Interfaces, GUIs), και αυτό γιατί έχει ενσωματωμένες τις βιβλιοθήκες Swing και AWT. Το NetBeans μας δίνει την δυνατότητα μέσα από την παλέτα που περιέχετε να κάνουμε χρήση όλων των συστατικών στοιχείων (components) των βιβλιοθηκών που αναφέρθηκαν. Το βασικό συστατικό στοιχείο είναι το JFrame, που μπορεί να περιέχεται σε αυτό άλλα συστατικά στοιχεία όπως JMenus, JButtons, JTextFields κ.α. [20]

Ακόμα όσο αφορά το σχεδιασμό του GUI γραφικά, κάνοντας οι προγραμματιστές χρήση του Design View, στο οποίο όλα λειτουργούν πολύ απλά, μπορούμε με ένα απλό drag-and-drop από την παλέτα σε ένα καμβά, να σχεδιάσει τη διεπαφή και ο κώδικας να παραχθεί αυτόματα και να υπάρχει αργότερα η ευκαιρία επεξεργασίας τους μέσω του Source View. Εννοείται φυσικά, πως εάν επιθυμεί ο προγραμματιστής μπορεί να σύνταξης απευθείας τον κώδικα για τον σχεδιασμό των αντικειμένων της διεπαφής, όπου κάποιες φορές μπορεί να διευκολύνει την πιο αποτελεσματική διαχείρισή τους. Επίσης, μπορούν να προστεθούν event handlers στο κάθε συστατικό στοιχείο, όπως για παράδειγμα είναι το Mouse Clicked, Action Performed κ.α.

4.2.3 Graphviz (Graph Visualization Software)

Το Graphviz είναι λογισμικό ανοικτού πηγαίου κώδικα (open-source) κατάλληλο για την απεικόνιση γράφων (graph visualization), εξειδικεύετε στην αναπαράσταση δομημένων πληροφοριών, όπως διαγράμματα δικτύων. Έχει εφαρμογές στα Δίκτυα Υπολογιστών, την Βιοπληροφορική, την Τεχνολογία Λογισμικού, το Σχεδιασμό Βάσεων Δεδομένων και Διαδικτυακών Εφαρμογών, στη μηχανική μάθηση και σε γραφικές διεπαφές πολλών άλλων τεχνικών κλάδων. [22]

Τα προγράμματα απεικόνισης που γράφονται σε Graphviz, χρησιμοποιούν ως input την περιγραφή των γράφων σε απλή γλώσσα κειμένου και δημιουργούν διαγράμματα σε

αρχεία με ποικίλα χρήσιμα formats, είτε εικόνες png και svg, έγγραφα pdf ή Postscript, ή προβολή σε διαδραστικούς graph browsers. [22]

Το Graphviz διαθέτει πολλά χρήσιμα χαρακτηριστικά για το σχεδιασμό διαγραμμάτων και γράφων, με πολλές επιλογές για χρώματα, γραμματοσειρές, στυλ γραμμών, υπερσυνδέσμους, custom σχήματα και διατάξεις κόμβων σε πίνακα. [22]

Οι τύποι αρχείων που χρησιμοποιούνται ως input είναι οι ακόλουθοι : [22]

- dot : για ιεραρχικά ή πολυεπίπεδα σχέδια κατευθυνόμενων γράφων. Αυτό το είδος είναι και το προεπιλεγμένο εργαλείο στην περίπτωση κατευθυνόμενων ακμών. Είναι πολύ απλό στη συγγραφή και σύνταξη του, τόσο που είναι εφικτό να το γράψει και χρήστης ο οποίος δεν έχει γνώσεις προγραμματισμού.
- neato : ειδικό για «spring model» διατάξεις. Αυτό το είδος εισόδου είναι το προεπιλεγμένο εργαλείο όπου ο γράφος δεν ξεπερνά τους 100 κόμβους, και περιέχονται άγνωστα χαρακτηριστικά.
- fdp : και το fdp ειδικεύεται για «spring model» διατάξεις όπως και το neato.
- twopi : είναι κατάλληλο για ακτινικές διατάξεις, δηλαδή οι κόμβοι τοποθετούνται σε ομόκεντρους κύκλους, ανάλογα με την απόστασή τους με δεδομένο έναν κεντρικό κόμβο.
- circo : κατάλληλο για κυκλικές διατάξεις. Εξειδικευμένο για συγκεκριμένα διαγράμματα με πολλαπλές κυκλικές δομές, π.χ. τηλεπικοινωνιακά δίκτυα.

4.2.4 Graphviz Java API

Το Graphviz Java API είναι μια κλάση/βιβλιοθήκη (Graphviz.java) γραμμένη σε Java, με την οποία γίνεται κλήση προγραμμάτων που έχουν δημιουργηθεί σε γλώσσα DOT, μέσω κώδικα στη Java. Δημιουργήθηκε το 2003 από τον Laszlo Szathmary, για να κάνει πιο εύκολη τη χρήση του εργαλείου Graphviz, μέσα σε προγράμματα της Java. [21]

Η κλάση επισυνάπτετε ολοκληρωμένη, στο Παράρτημα Γ.

4.2.5 Συνδυασμός GUI σε Java και Graphviz

Έγινε μια προσπάθεια σύνδεσης τον ανωτέρω, δηλαδή του NetBeans IDE και της βιβλιοθήκης Graphviz.java. Αρχικά, με το NetBeans IDE το οποίο μπορέσαμε να δημιουργήσουμε μια πολύ εύχρηστη γραφική διεπαφή, στην οποία μπορέσαμε να προσθέσουμε και τη βιβλιοθήκη Graphviz.java. Έτσι παράχθηκαν οι κατάλληλες απεικονίσεις κυτταρικών δικτύων. Αυτές οι απεικονίσεις, εντάχθηκαν στη διεπαφή με τη μορφή .png εικόνων. Ακολούθως, εισήχθηκαν τα πειραματικά δεδομένα χρησιμοποιώντας jPopupMenu και για κάθε αλλαγή, παραγόταν και εμφανιζόταν μια νέα .png εικόνα περιέχοντας την κάθε αλλαγή.

Μ' αυτό το συνδυασμό εργαλείων, μπόρεσαν στην προηγούμενη υλοποίηση, BioCT, να καλυφθούν όλες οι απαιτήσεις του χρήστη σχετικά με τη γραφική διεπαφή, καθώς και σχετικά με την απεικόνιση των δύο κυτταρικών δικτύων. Ωστόσο, υπήρξαν αρκετά προβλήματα στατικότητας, όπως ήδη αναφέρθηκε στην ενότητα 3.4. Γι' αυτό το λόγο, διερευνήθηκαν εναλλακτικές.

4.3 Σύνολο Τεχνολογιών 2 : GUI σε Java, GraphStream

Το Graphstream αποτελεί μια Java βιβλιοθήκη για την μοντελοποίηση και ανάλυση δυναμικών γράφων, σχεδιασμένη από ερευνητικές ομάδες του πανεπιστημίου University of LeHavre. [23]

Διαχειρίζεται τα δυναμικά χαρακτηριστικά των γράφων, εστιάζοντας στην μοντελοποίηση δικτύων δυναμικής αλληλεπίδρασης διαφόρων ειδών και μεγεθών. Στόχος της βιβλιοθήκης αυτής είναι να παρέχει μια μέθοδο αναπαράστασης και επεξεργασίας των γράφων, προτείνοντας έναν αριθμό classes που επιτρέπουν τον σχεδιασμό κατευθυνόμενων και μη-κατευθυνόμενων γράφων (l-graphs ή p-graphs). Ταυτόχρονα, επιτρέπει την αποθήκευση χαρακτηριστικών (attributes) στα δομικά στοιχεία των γράφων (elements), όπως αριθμούς, strings ή αντικείμενα. [23]

Επιπροσθέτως, το GraphStream παρέχει τον τρόπο διαχείρισης της χρονικής εξέλιξης του γράφου, δηλαδή τον τρόπο με τον οποίο προστίθεται ή αφαιρούνται οι κόμβοι, καθώς και τον τρόπο με τον οποίο εμφανίζονται, εξαφανίζονται ή μεταλλάσσονται τα διάφορα χαρακτηριστικά των στοιχείων του γράφου. [23]

Για την διαχείριση δυναμικών γράφων, η βιβλιοθήκη δεν ορίζει μόνο τις δομές των γράφων, αλλά και την έννοια της «ροής των συμβάντων του γράφου» («stream of graph events»), τα οποία αφορούν σε προσθήκη, αφαίρεση κόμβου ή ακμής, καθώς και σε προσθήκη, αφαίρεση ή τροποποίηση ενός attribute του γράφου/κόμβου/ακμής. [23]

Αξίζει να αναφερθεί σε αυτό το σημείο, ότι μια λειτουργία του GraphStream είναι να διαβάζει αρχεία δεδομένων, όπως είναι αποθηκευμένα στο σύνολο δεδομένων(dataset) και να τα μετατρέπει σε ένα αντικείμενο τύπου Graph. Δέχεται πληθώρα τύπων αρχείων για είσοδο, μερικά από αυτά είναι: DGS (GraphStream), DOT (GraphViz), TLP (Tulip), κ.α.

Το GraphStream αντιμετωπίζει τη δομή του γράφου σαν pipe, το οποίο μπορεί να δέχεται συμβάντα λειτουργώντας ως sink (δεξαμενή) ή να τα παράγει, λειτουργώντας ως source (πηγή). Δέχεται το input από έναν file reader, ο οποίος διαβάζει τα events που περιγράφονται σε ένα αρχείο. Κάθε φορά που ο γράφος λαμβάνει ένα γεγονός μεταβάλλει τη δομή του για να αντικατοπτρίσει τα αποτελέσματα αυτού του γεγονότος, λειτουργώντας έτσι ως sink. Αποτελεί όμως και πηγή, διοχετεύοντας το output σε έναν viewer, ο οποίος αναλαμβάνει τη γραφική απεικόνιση του γράφου. [23]

Όσον αφορά στην γραφική απεικόνιση, το GraphStream δίνει τη δυνατότητα αυτόματης οπτικοποίησης του γράφου σε τυχαία διάταξη, ή τον καθορισμό των χαρακτηριστικών που θα εφαρμόσει ο viewer στην απεικόνιση των στοιχείων του γράφου, μέσω της χρήσης style sheets, τα οποία προσομοιώνουν τον τρόπο λειτουργίας του CSS στη χρήση HTML. Δίνεται δυνατότητα επιλογής ανάμεσα στον προεπιλεγμένο viewer του GraphStream που μπορεί να διαχειριστεί μόνο το χρώμα και μέγεθος των στοιχείων του γράφου και στο gs-ui module(-υπομονάδων) που παρέχει πλήρη υποστήριξη CSS για τον καθορισμό περαιτέρω χαρακτηριστικών. [23]

Η διαχείριση της γραφικής απεικόνισης των γράφων γίνεται κατά συνέπεια πιο ευέλικτη, σε σχέση με τον στατικό χαρακτήρα του GraphViz (με αποτέλεσμα το πρόβλημα του zoom ίσως να αντιμετωπιστεί και με εξ' ολοκλήρου διαφορετική προσέγγιση, αναλόγως τα αποτελέσματα της περαιτέρω μελέτης), ενώ και η δυναμική προσθαφαίρεση κόμβων και ακμών που αποτελεί βασική απαίτηση του υπό ανάπτυξη συστήματος, φαίνεται να διευκολύνεται σημαντικά με την βιβλιοθήκη GraphStream. [23]

Έγινε μελέτη του τρόπου αξιοποίησης του GraphStream και ένταξής του στην παλαιότερη υλοποίηση του συστήματος και πιο συγκεκριμένα του τρόπου με τον οποίο θα «συνεργάζονται» οι διάφορες λειτουργικές μονάδες της εφαρμογής. Ωστόσο, η διαδικασία ένταξής του αποδείχθηκε αρκετά πολύπλοκη, ενώ και για την γραφική απεικόνιση απαιτείτο εξοικείωση σε ένα πιο διευρυμένο γνωστικό φάσμα. [23]

4.4 Σύνολο Τεχνολογιών 3 : GUI σε Java, Cytoscape, βιβλιοθήκη KEGG

Το σύνολο τεχνολογιών που θα περιγραφεί σε αυτό το υποκεφάλαιο, είναι μια εκ νέου προσέγγιση του θέματος, η οποία και δεν ολοκληρώθηκε.

4.4.1 GUI σε Java

Τα χαρακτηριστικά έχουν αναλυθεί στο Κεφάλαιο 4.2.2

4.4.2 Cytoscape

Το Cytoscape είναι ένα λογισμικό ειδικό για το τομέα της βιοπληροφορικής, ανοικτού κώδικα για οπτικοποίηση των δικτύων μοριακών αλληλεπιδράσεων και όχι μόνον.

Μπορούν να προστεθούν πολλές λειτουργίες, προσθέτοντας plugins όπως και διάφορες εξωτερικές Βάσεις Δεδομένων. Επίσης μπορεί να γίνει και ανάπτυξη νέων plugin μέσω προγραμματιστικού περιβάλλοντος που παρέχει το ίδιο το Cytoscape σε γλώσσα προγραμματισμού Java.

Μερικά από τα πλεονεκτήματα του είναι ότι αρχικά σχεδιάστηκε για οπτικοποίηση βιολογικών γράφων, αλλά δεν περιορίζεται μόνο σε αυτούς. Μπορεί να γίνει οπτικοποίηση και χειρισμός γράφων οποιασδήποτε επιστημονικής περιοχής. Παράλληλα με αυτά, μας χαρίζει την δυνατότητα να σχεδιάζουμε εκ νέου γράφους, να τους εισάγουμε (import) από εξωτερικές Βάσεις Δεδομένων, είτε με μορφή πίνακα, είτε με μορφή εικόνας όπως ακριβώς παράγονται από τις εξωτερικές βάσεις δεδομένων. Επίσης, μετά το σχεδιασμό ή της εισαγωγή των δεδομένων, έχουμε τη δυνατότητα αποθήκευσης του γράφου ή εξαγωγής του σε άλλη μορφή. Οι μορφές που

μπορεί να εξάγει (export) το Cytoscape είναι διάφορες, όπως για παράδειγμα, απλό Network δηλαδή σε μορφή XGMML το οποίο είναι XML αρχείο , ή Network ως γραφικά δηλαδή σε .png, .pdf , .svg scalable Vector Graphics, .ps Post Script File. Όλες αυτές οι μορφές τυγχάνουν διαφορετικής επεξεργασίας και τρόπου αντιμετώπισης, ανάλογων των εργασιών που επιθυμούμε να κάνουμε.

4.4.3 Βάση Δεδομένων KEGG (Kyoto Encyclopedia of Genes and Genomes)

Η βάση Δεδομένων KEGG, είναι μια από τους πιο σημαντικούς και βασικούς διεθνείς οργανισμούς που παρέχει πληροφορίες σχετικά και Βιολογικά Δίκτυα. Αποτελεί πόρο για την κατανόηση υψηλού επιπέδου λειτουργιών και υπηρεσιών, ικανές να βοηθήσουν σε γενικές γραμμές το Βιολογικό Σύστημα. Εξυπηρετούν την κατανόηση των λειτουργιών του κυττάρου, του οργανισμού, του οικοσυστήματος όπως επίσης και πολλές άλλες μονάδες οι οποίες μπορούν να πάρουμε πληροφορίες και κατάλληλες απεικονίσεις από μοριακό επίπεδο μέχρι γενικής κλίμακας. Επίσης η KEGG παρέχει πληροφορίες για γονίδια, πρωτεΐνες, χημικές ενώσεις και ακολουθίες νουκλεοτιδίων

Κατ' επέκταση από αυτή τη βιβλιοθήκη μπορούμε να πάρουμε αρχεία XML, γνωστά και ως KGML (KEGG Markup Language).

4.4.4 Συνδυασμός GUI σε Java και Cytoscape

Στο σημείο αυτό, έγινε προσπάθεια να συνενωθούν τα παραπάνω. Έγινε μελέτη του τρόπου αξιοποίησης του Cytoscape και η ένταξη των πληροφοριών που μπορούμε να πάρουμε από το KEGG. Ο τρόπος με τον οποίο θα «συνεργάζονταν», κατάληγε σε χαοτικό αποτέλεσμα. Για να γίνω σαφής, το αποτέλεσμα που θα δημιουργείτο μετά από αρκετή μελέτη, θα ήταν μια γραφική διεπαφή στο GUI σε Java για επεξεργασία πειραματικών δεδομένων και ανεξάρτητα με αυτό, για την μορφοποίηση και επεξεργασία των γράφων θα έπρεπε να γίνετε στο περιβάλλον του Cytoscape. Συμπερασματικά, η προσέγγιση αυτή απαιτούσε μεγάλο χρονικό διάστημα και πολλές γνώσεις καθώς και εξοικείωση με το περιβάλλον του Cytoscape.

4.5 Σύνολο Τεχνολογιών 4 : GUI σε Java, Graphviz, ZGRViewer

Στο Σύνολο Τεχνολογιών 1 της ενότητας 4.2, μετά από αρκετή έρευνα και μελέτη, εντάχθηκε και ο ZGRViewer, που αποτελεί εργαλείο απεικόνισης γράφων (graph visualizer) υλοποιημένο σε Java και βασισμένο στο ZVTM (Zoomable Visual Transformation Machine) toolkit.

4.5.1 ZVTM

Τα εργαλεία Γραφικής Διεπαφής Χρήστη (GUI toolkits) όπως το Swing είναι ισχυρά, γενικά και φορητά, ωστόσο δεν μπορούν να χρησιμοποιηθούν σε κλάσεις ορισμένων γενικών εφαρμογών όπως οι επεξεργαστές γράφων, αναγκάζοντας τους προγραμματιστές να χρησιμοποιούν low-level APIs, όπως το Java2D, που έχουν μεγαλύτερο κόστος και δυσκολία στη χρήση. [25]

Το ZVTM είναι εργαλείο Διεπαφής Χρήστη με δυνατότητα μεγέθυνσης (Zoomable User Interface, ZUI) υλοποιημένο σε Java και σχεδιασμένο να διευκολύνει τη δημιουργία πολύπλοκων γραφικών επεξεργαστών με μεγάλο αριθμό αντικειμένων προς προβολή ή με πολύπλοκα γεωμετρικά σχήματα προς αναπαραγωγή. [25]

Βασίζεται στη μεταφορά των universes που μπορούν να παρατηρηθούν μέσω ευφών «movable/zoomable cameras» και προσφέρει δυνατότητες και χαρακτηριστικά που κάνουν την συνολική εμπειρία του τελικού χρήστη πιο ικανοποιητική. [25]

Το ZVTM περιλαμβάνει ένα μοντέλο γραφικών αντικειμένων που διευκολύνει την δημιουργία, τροποποίηση και αναπαραγωγή γραφικών οντοτήτων και επιτρέπει τον ορισμό custom σχημάτων, μέσω ενός απλού API, ενώ διαθέτει δυνατότητες ομαλής μεγέθυνσης (smooth zooming), πολλαπλών ανεξάρτητων επιπέδων (layers) σε ένα μοναδικό παράθυρο (viewport), προβολής vector graphics, κειμένου, bitmap εικόνων, pdf εγγράφων και υποστήριξης ενός υποσυνόλου του SVG. [25]

Αποτελεί συνέχεια ενός project που ξεκίνησε από το Xerox Research Centre Europe και συνεχίστηκε στο MIT και στο W3C, ως XVTM. Τώρα, αναπτύσσεται από το INRIA (Institut de Recherche Dédie au Numerique). [25]

4.5.2 ZGRViewer

Όπως προαναφέρθηκε, ο ZGRViewer αποτελεί εργαλείο απεικόνισης γράφων (graph visualizer) υλοποιημένο σε Java και βασισμένο στο ZVTM toolkit. Αποσκοπεί κυρίως στην προβολή γράφων που περιγράφονται με χρήση της γλώσσας DOT από το AT&T GraphViz και επεξεργάζονται από τα προγράμματα dot, neato, twopi κλπ. [24]

Σχεδιάστηκε για να διαχειρίζεται πολύ μεγάλους γράφους, και προσφέρει ένα ZUI που επιτρέπει smooth zooming και εύκολη περιήγηση στην απεικονισμένη δομή. Ορισμένα από τα χαρακτηριστικά που βοηθούν στην περιήγηση πολύ μεγάλων γράφων είναι : [24]

- Overview & Detail (Επισκόπηση και Λεπτομερής Προβολή)
- Focus & Context Magnification with Sigma Lenses (Εστίαση & Μεγέθυνση Πλαισίου μέσω Sigma Lenses)
- Graphical Fisheye Focus & Context Distortion (Γραφική Εστίαση Fisheye & Παραμόρφωση Πλαισίου)
- Navigation Along Graph Edges with Link Sliding (Περιήγηση κατά μήκος των Ακμών του Γράφου μέσω Link Sliding)
- Navigation from Node to Node with Bring & Co (Περιήγηση από Κόμβο σε Κόμβο με το Bring & Co)
- Running on a Cluster-Driven Wall-Sized Display

Μπορεί να φορτώσει οποιοδήποτε αρχείο χρησιμοποιεί την γλώσσα DOT για περιγραφή του γράφου. [24]

Βασίζεται στο GraphViz για να υπολογίσει και να δημιουργήσει ένα αρχείο SVG (Scalable Vector Graphics) που περιέχει την γραφική αναπαράσταση του γράφου από την αφηρημένη DOT αναπαράστασή του. Αυτό συνεπάγεται ότι το GraphViz πρέπει να είναι ήδη εγκατεστημένο στον υπολογιστή όπου θα εκτελεστεί ο ZGRViewer. Το SVG αρχείο που παράγεται από τα προγράμματα του GraphViz (dot, neato κλπ) αναλύεται από την υπομονάδα(module) εισαγωγής SVG του ZVTM και προβάλλεται στο χρήστη. [24]

Υπάρχει διαθέσιμη και έκδοση applet του ZGRViewer. [24]

4.5.3 Συνδυασμός GUI σε Java, Graphviz και ZGRViewer

Είναι προφανές και από τις ανωτέρω περιγραφές των εργαλείων, ότι είναι εξ' ορισμού κατασκευασμένα για να «συνεργάζονται» κάτω από το κοινό πλαίσιο εργασίας της Java. Επομένως, ο συνδυασμός αυτός των τεχνολογιών έχει τις δυνατότητες και προϋποθέσεις να αξιοποιηθεί κατάλληλα για την δημιουργία εφαρμογής που να ικανοποιεί τις απαιτήσεις που περιγράφονται στο επόμενο κεφάλαιο.

Η πρόκληση αφορά στην ορθή χρήση των βιβλιοθηκών (libraries) που παρέχονται από τα εργαλεία αυτά και στην αποτελεσματική ένταξή τους στην υπάρχουσα υλοποίηση, για την αναβάθμισή της.

Κεφάλαιο 5: Το Σύστημα BioCLC – Biological Cell Cycle

5.1 Εισαγωγή	32
5.2 Ανάλυση Συστήματος - Απαιτήσεις και προδιαγραφές	32
5.3 Μοντέλο Κύκλου Ζωής Λογισμικού Συστήματος	36
5.4 Σχεδιασμός Συστήματος	42
5.5 Υλοποίηση Συστήματος	43

5.1 Εισαγωγή

Το σύστημα BioCLC (Biological Cell Cycle) αποτελεί την τελική Γραφική Διεπαφή Χρήστη που δημιουργήθηκε στηριζόμενη στην αμέσως προηγούμενη υλοποίηση, το σύστημα BioCT. Επιχειρήθηκε η βελτίωση, αναβάθμιση και επέκταση των χαρακτηριστικών που παρείχε το BioCT, αλλά και ο ταυτόχρονος περιορισμός των αδυναμιών που είχε εμφανίσει.

Ικανοποιεί σημαντικό μέρος των απαιτήσεων και προδιαγραφών που καταγράφονται αναλυτικά στην επόμενη ενότητα, πάντα στα πλαίσια και τους στόχους της εκπονούμενης διπλωματικής εργασίας. Στις ενότητες που θα ακολουθήσουν την ανάλυση απαιτήσεων, θα γίνει μια σύντομη αναφορά στον κύκλο ζωής του λογισμικού που ακολουθήθηκε για την δημιουργία του τελικού συστήματος, ενώ θα γίνει αναλυτική περιγραφή όλων των λειτουργιών που υλοποιήθηκαν, καθώς και εκείνων που για συγκεκριμένους λόγους δεν κατέστη εφικτό να ολοκληρωθούν.

5.2 Ανάλυση Συστήματος - Απαιτήσεις και προδιαγραφές

Οι απαιτήσεις του συστήματος κατά κύριο λόγο ακολουθούν εκείνες της προηγούμενης υλοποίησης [8], με βασική μας πρόκληση την επίλυση των σημείων που δεν κατέστη δυνατόν να ολοκληρωθούν. Παράλληλα, στηρίζονται στα αποτελέσματα του

ερωτηματολογίου που δημιουργήσαμε σε συνεργασία με την απόφοιτη φοιτήτρια και δημιουργό της προηγούμενης έκδοσης του συστήματος, Ειρήνη Ανδρέου, το οποίο παρατίθεται στο Παραρτήματος Α. Το ερωτηματολόγιο απαντήθηκε από τον βασικό χρήστη της εφαρμογής, τη βιολόγο κα Χριστιάνα Νεοφύτου, κατόπιν συναντήσεων που πραγματοποιήθηκαν έτσι ώστε να συγκεντρώσουμε και καταγράψουμε τις προδιαγραφές της διεπαφής.

Πιο κάτω παρουσιάζονται οι Απαιτήσεις Προδιαγραφών του συστήματος:

5.2.1 Ασφαλής Είσοδος στο Σύστημα

Για λόγους ασφαλείας, το σύστημα θα πρέπει να χρησιμοποιείται μόνο από επικυρωμένους χρήστες. Μέσα από το σύστημα θα διαβάζονται πειραματικά δεδομένα, προσωπικά για τον κάθε χρήστη. Επομένως, σε κάθε χρήστη που θα εγκαθίσταται το σύστημα θα παρέχεται ένας μοναδικός και προκαθορισμένος προσωπικός κωδικός, ο οποίος θα ζητείται κάθε φορά κατά την εκκίνηση του συστήματος.

5.2.2 Αρχική Οθόνη

Όταν ο χρήστης κάνει με επιτυχία ασφαλή είσοδο στο σύστημα, τότε θα εμφανίζεται η αρχική οθόνη του συστήματος. Στην οθόνη αυτή, θα περιέχονται 3 βασικά κουμπιά για εκτέλεση πειραμάτων και η επιλογή της εξόδου από το σύστημα. Συγκεκριμένα, ο χρήστης, θα έχει την δυνατότητα να επιλέξει ένα από τα πιο κάτω:

Μονοπάτι Απόπτωσης

Στο μονοπάτι αυτό θα εμφανίζονται οι πρωτεΐνες, διεργασίες και ουσίες, και οι ακμές που υπάρχουν ανάμεσά τους, όπως φαίνονται στην ιστοσελίδα της Cell Signaling, συγκεκριμένα στο μονοπάτι που απεικονίζει τη ρύθμιση της απόπτωσης.

Μονοπάτι Πολλαπλασιασμού

Στο μονοπάτι αυτό θα εμφανίζονται οι πρωτεΐνες, διεργασίες και ουσίες, και οι ακμές που υπάρχουν ανάμεσά τους, όπως φαίνονται στην ιστοσελίδα της Cell Signaling, συγκεκριμένα στο μονοπάτι που απεικονίζει τη ρύθμιση του σημείου ελέγχου του κυτταρικού πολλαπλασιασμού, όπως αυτό παρουσιάζεται και πάλι στην ιστοσελίδα της Cell Signaling.

Νέο Κυτταρικό Δίκτυο / Μονοπάτι

Αυτή η οθόνη θα είναι κενή και θα παρέχει τη δυνατότητα στο χρήστη να ανοίξει ένα υπάρχον κυτταρικό δίκτυο και να το επεξεργαστεί.

Έξοδος

Με την επιλογή αυτή, θα πραγματοποιείται έξοδος από το σύστημα.

5.2.3 Αντιστοιχία Μεταξύ των Διαφόρων Οθονών

Καθώς ο χρήστης βρίσκεται στην οθόνη της απόπτωσης και μελετά μια ουσία, πρωτεΐνη ή διεργασία, θα πρέπει να υπάρχει η δυνατότητα να μεταβεί στην οθόνη του πολλαπλασιασμού στην ίδια ουσία, πρωτεΐνη ή διεργασία και η διαδικασία αυτή να έχει ένα διευκρινιστικό σήμα για εύκολο εντοπισμό της από τις υπόλοιπες. Επεξηγηματικά, με την μετάβαση αυτή ο χρήστης θα βλέπει τονισμένη με διαφορετικό χρώμα την επιθυμητή ουσία, πρωτεΐνη ή διεργασία. Στην περίπτωση που δεν υπάρχει αυτό το συστατικό μέλος, θα πρέπει με το ανάλογο μήνυμα να ενημερώνετε ο χρήστης. Η αντιστοιχία αυτή, γίνεται με τον ακριβώς ίδιο τρόπο και από την οθόνη του πολλαπλασιασμού, στην οθόνη της απόπτωσης.

5.2.4 Επιλογές Open και Save

Ο χρήστης, από το βασικού μενού επιλογών, θα πρέπει να έχει τη δυνατότητα να ανοίξει ένα παλαιότερο πείραμά του, ή να αποθηκεύσει οποιοδήποτε άλλο πείραμα για μελλοντική επεξεργασία του.

5.2.5 Έξοδος από μια Οθόνη

Σε κάθε οθόνη θα πρέπει να υπάρχει ένα κουμπί για έξοδο για να διευκολύνεται η περιήγηση του χρήστη στο σύστημα. Θα ήταν καλό να υπάρχει και μια εναλλακτική επιλογή για έξοδο, ίσως στο βασικό μενού επιλογών.

5.2.6 Επιλογή Αναζήτησης

Σε κάθε οθόνη θα πρέπει ο χρήστης να μπορεί να αναζητήσει μια πρωτεΐνη, ουσία ή διεργασία στο εκάστοτε μονοπάτι που βρίσκεται. Η αναζήτηση αυτή θα πραγματοποιείτε κάνοντας εισαγωγής του ονόματός της σε ένα νεό παράθυρο. Αν υπάρχει στο μονοπάτι, το σύστημα θα πρέπει να έχει ένα διευκρινιστικό σήμα όπως η αλλαγή χρώματος του κόμβου . Αν δεν υπάρχει, θα πρέπει με το ανάλογο μήνυμα να ενημερώνετε ο χρήστης.

5.2.7 Zoom

Σε κάθε οθόνη αναπαράστασης μονοπατιών, θα πρέπει να υπάρχει η επιλογή για zoom με σημεία ένδειξης (+) για μεγέθυνση (zoom-in) και (-) για σμίκρυνση (zoom-out) για διευκόλυνση του χρήστη, και γενικότερη ή ειδικότερη προβολή αναλόγως.

5.2.8 Απεικόνιση Μονοπατιών

Η απεικόνιση των μονοπατιών, όσον αφορά την απεικόνιση των κόμβων στα μονοπάτια, μας έχει ζητηθεί από τον χρήστη συγκεκριμένος τρόπος εμφάνισεις τους. Δηλαδή, οι πρωτεΐνες θα πρέπει να εμφανίζονται σε οβάλ σχήμα, οι διεργασίες σε τετράγωνο ή ορθογώνιο σχήμα και οι άλλες ουσίες χωρίς περίγραμμα, απλά με το όνομά τους.

5.2.9 Εισαγωγή Πειραματικών Δεδομένων

Ένας απλός τρόπος εισαγωγής πειραματικών δεδομένων είναι να εκτελείται από το χρήστη, δεξί κλικ πάνω σε μια πρωτεΐνη, ουσία ή διεργασία. Κάνοντας δεξί κλικ, ο

χρήστης θα μπορεί να επιλέξει εάν αυτή είναι active fixed (ενεργή και το ξέρουμε εκ των προτέρων), inactive fixed (απενεργοποιημένη και το ξέρουμε εκ των προτέρων, δηλαδή σαν να απουσιάζει από το κύτταρο), active observed (ενεργή μετά από το πείραμα) ή inactive observed (απενεργοποιημένη μετά από το πείραμα). Αν έχει επιλέξει το active fixed, το περίγραμμα της πρωτεΐνη, ουσία ή διεργασία θα γίνεται γαλάζιο, αν έχει επιλέξει το inactive fixed το περίγραμμα της θα γίνεται πορτοκαλί, έχει επιλέξει το active observed το περίγραμμα της θα γίνεται πράσινο και αν έχει επιλέξει το inactive observed, το περίγραμμα της θα γίνεται κόκκινο.

5.2.10 Απεικόνιση Διάδοσης Σήματος

Στην φάση που θα γίνεται η διάδοση σήματος, οι κόμβοι από τους οποίους έχει περάσει σήμα ενεργοποίησης, θα γεμίζουν με πράσινο χρώμα ενώ οι κόμβοι από τους οποίους έχει περάσει σήμα απενεργοποίησης, θα γεμίζουν με κόκκινο χρώμα.

5.2.11 Δυνατότητα Πρόσθεσης Νέων Κόμβων και Συνδέσεων

Ο χρήστης επιθυμεί να έχει την επιλογή «add». Όταν επιλεγθεί, θα εμφανίζεται ένα παράθυρο όπου ο χρήστης θα γράφει το όνομα του νέου κόμβου, θα επιλέγει αν είναι πρωτεΐνη, διεργασία ή άλλη ουσία και θα προσθέτει τις ανάλογες συνδέσεις με άλλους κόμβους καθώς και τον τύπο των συνδέσεων (ενεργοποιητικό ή ανασταλτικό). Επίσης, θα υπάρχει η επιλογή για προσθήκη νέας ακμής-σύνδεσης ανάμεσα σε υπάρχοντες κόμβους ή κόμβους οι οποίοι έχουν δημιουργηθεί με την επιλογή «add» νωρίτερα.

5.3 Μοντέλο Κύκλου Ζωής Λογισμικού Συστήματος

Ο κύκλος ζωής ανάπτυξης συστημάτων αποτελεί συστηματική μεθοδολογία για την ανάπτυξη λογισμικού με τις παρακάτω βασικές αρχές [26] :

- Η διαδικασία ανάπτυξης διαχωρίζεται σε φάσεις
- Κάθε φάση είναι η συνέχεια της προηγούμενης

- Κάθε φάση αποτελείτε από συγκεκριμένες εργασίες
- Οι εργασίες που προγραμματίζονται για τη κάθε φάσης, πρέπει να ολοκληρωθούν πριν την έναρξη της επόμενης
- Όταν ολοκληρωθεί η κάθε φάσης, γίνεται καταγραφή των εργασιών που περατώθηκαν και των αποτελεσμάτων, αυτό αποσκοπεί στον έγκαιρο εντοπισμό και την αποτελεσματική επίλυση τυχόν προβλημάτων.

Ακολουθώντας τις φάσεις αυτές και στη παρούσα ερευνά, είχαμε απώτερο σκοπό, την πλέον κατάλληλη γραφική διεπαφή για τον τελικό χρήστη, με γνώμονα πάντοτε την αποτροπή σφαλμάτων κατά την ανάπτυξη του Πληροφοριακού Συστήματος, όπως επίσης και στόχος μας ήταν να τηρήσουμε μια δεοντολογικά ορθή ροή των πραγμάτων.

Αφού λάβαμε υπόψη τις πιο πάνω βασικές αρχές, έπειτα, ακολουθήσαμε τα στάδια κύκλου ζωής Λογισμικού Συστήματος. Τα στάδια παρατίθενται πιο κάτω (όπου με έντονα γράμματα σημειώνονται όσα αφορούν στις φάσεις ανάπτυξης του λογισμικού) [26] :

- Προκαταρκτική έρευνα
- Μελέτη σκοπιμότητας
- **Ανάλυση απαιτήσεων**
- **Σχεδιασμός συστήματος**
- **Προγραμματισμός (Υλοποίηση)**
- **Έλεγχος συστήματος**
- Εγκατάσταση και εκπαίδευση
- Συντήρηση και βελτιώσεις

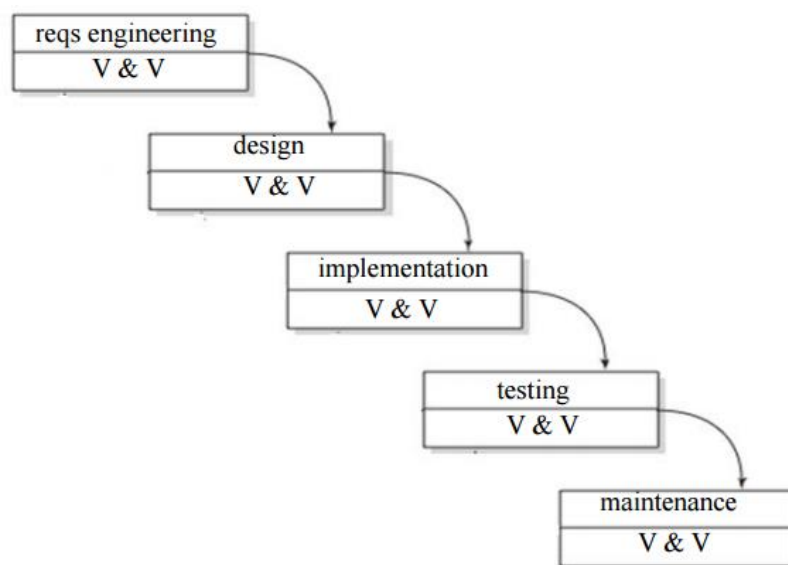
Ένα μοντέλο κύκλου ζωής λογισμικού είναι μια περιγραφή των δραστηριοτήτων και των επιμέρους φάσεων από τις οποίες περνάει μια εφαρμογή λογισμικού από τη σύλληψη της ιδέας μέχρι την απόσυρσή της, καθώς και των εργασιών που περιλαμβάνουν αυτές. Στοχεύει στην καθοδήγηση του κατασκευαστή, ώστε να επιτευχθεί η βέλτιστη υλοποίηση. [27]

Τα πιο διαδεδομένα μοντέλα κύκλου ζωής είναι [27] :

- Μοντέλο του Καταρράκτη (Waterfall) : Το σύστημα λογισμικού αναπτύσσεται μέσα από διαδοχικές επιμέρους φάσεις (σειριακά βήματα), καθεμιά από τις οποίες θεωρείται ολοκληρωμένη όταν έχουν παραχθεί συγκεκριμένα συστατικά του λογισμικού. Παρ' όλα αυτά, μπορούν να γίνουν μετατροπές ανάμεσα σε δύο βήματα. Η ανάπτυξη χαρακτηρίζεται ακολουθιακή (sequential) και διευκολύνει τη συντήρηση του λογισμικού.

Δεν θα μπορέσουμε να χρησιμοποιήσουμε αυτό το μοντέλο στη δική μας εργασία, γιατί προϋποθέτει την συνεχή ενημέρωση και εμπλοκή του χρήστη σε όλη την διαδικασία ανάπτυξης του συστήματος. Στη δικιά μας περίπτωση ο χρήστης θα εμπλακεί μόνο στην αρχή και θα δει το λογισμικό όταν ολοκληρωθούν πλήρως οι απαιτήσεις του.

Ακολουθεί σχεδιάγραμμα για την πλήρη κατανόηση του μοντέλου.



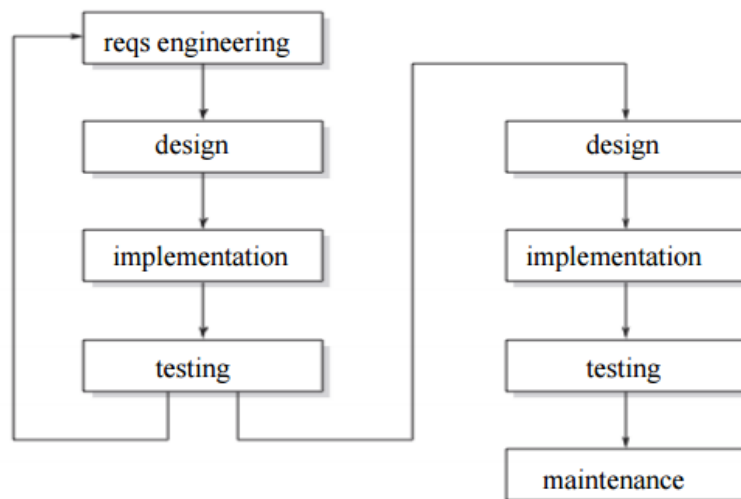
Εικόνα 12 : Σχεδιάγραμμα Μοντέλο του Καταρράκτη

- Μοντέλο Πρωτυποποίησης (Prototyping) : Αρχικά γίνεται ανάπτυξη του συστήματος σε τμήματα (πρωτότυπα), με διαδικασίες που επαναλαμβάνονται για ένα τμήμα του συστήματος κάθε φορά (επαναληπτική). Κάθε πρωτότυπο

περιλαμβάνει τις λειτουργίες που πρέπει να εκτελεί το λογισμικό, χωρίς όμως να έχει πλήρη λειτουργικότητα. Τίθεται σε δοκιμασία από τον πελάτη, συλλέγονται οι παρατηρήσεις, και δημιουργείται εκ νέου πρωτότυπο, ωστόσο κάποιο να ικανοποιεί τις απαιτήσεις του πελάτη. Επομένως, κατανοούνται πλήρως οι απαιτήσεις και ανάγκες του χρήστη και έτσι διαχειρίζονται νωρίτερα τα προβλήματα.

Δεν θα χρησιμοποιήσουμε αυτό το μοντέλο στη δική μας εργασία, ο λόγος που απορρίφθηκε είναι το γεγονός ότι δεν είχαμε τον απαιτούμενο χρόνο ώστε να κάνουμε πρωτότυπα, να τα παρουσιάσουμε στο χρήστη και έπειτα να αρχίσουμε την τελική υλοποίηση.

Ακολουθεί σχεδιάγραμμα για την πλήρη κατανόηση του μοντέλου.

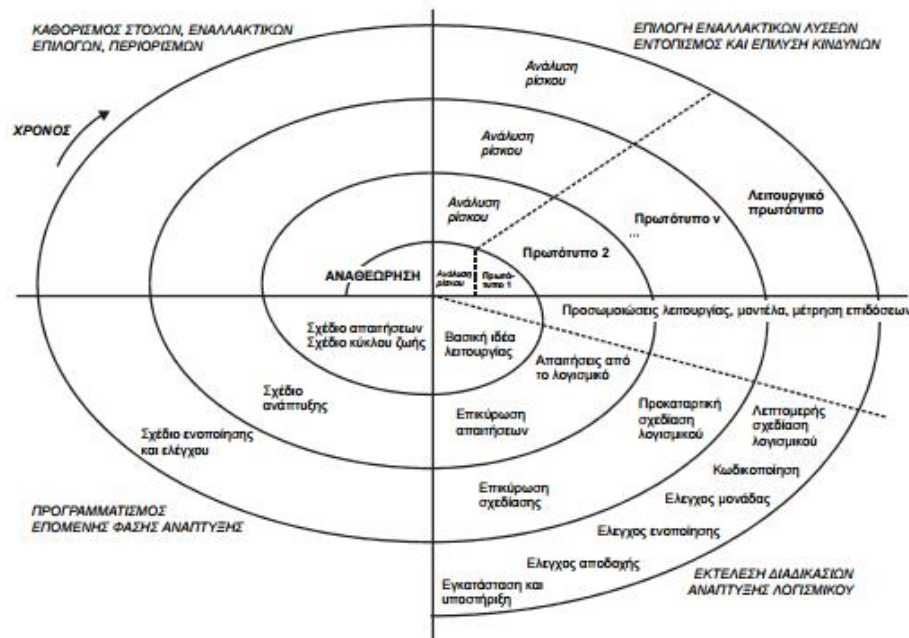


Εικόνα 13: Σχεδιάγραμμα Μοντέλο του Πρωτυποποίησης

- Σπειροειδές Μοντέλο (Spiral) : Αποτελεί γενίκευση του μοντέλου προτυποποίησης, με προσθήκες, και τέσσερις κατηγορίες εργασιών : (1)προσδιορισμό στόχων, (2)εντοπισμό και επίλυση κινδύνων, (3)εκτέλεση διαδικασιών ανάπτυξης και επαλήθευση, (4)εργασίες προγραμματισμού. Σε οποιαδήποτε φάση κι αν είμαστε, αν ο εντοπισμός και επίλυση κινδύνων αποτύχει, το έργο θα διακοπεί ή αν είναι σε πολύ προχωρημένο επίπεδο θα ολοκληρωθεί.

Δεν θα χρησιμοποιήσουμε αυτό το μοντέλο στη δική μας εργασία, ο λόγος που απορρίφθηκε είναι το γεγονός ότι το μοντέλο αυτό δεν είναι ειδικό για μικρά έργα όπως το δικός μας, αλλά για μεγάλης εμβέλειας λογισμικά.

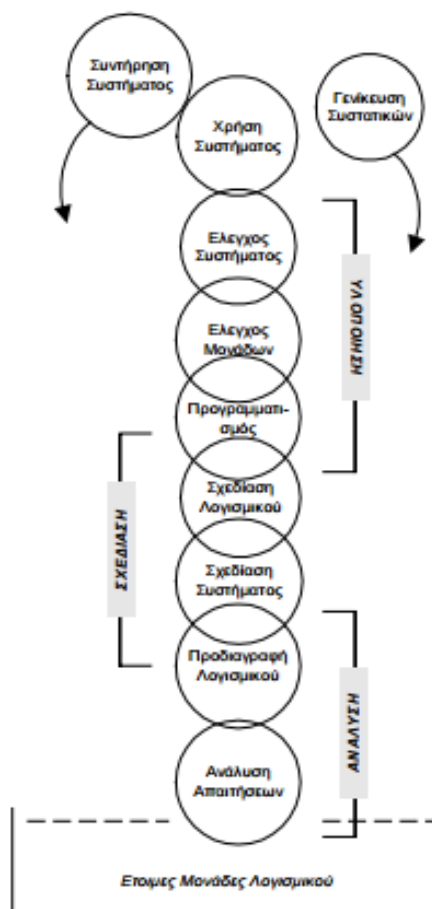
Ακολουθεί σχεδιάγραμμα για την πλήρη κατανόηση του μοντέλου.



Εικόνα 14: Σχεδιάγραμμα Μοντέλο του Καταρράκτη

- Μοντέλο του Πίδακα (Fountain) : Βασίζεται στην αντικειμενοστρεφή τεχνολογία ανάπτυξης, όπου οι έννοιες ανάλυση – σχεδίαση – κωδικοποίηση συγκλίνουν και το αποτέλεσμα κάθε διαδικασίας κατασκευής είναι όχι μόνο ένα σύστημα, αλλά και επαναχρησιμοποιήσιμες μονάδες για μελλοντικά συστήματα.

Ακολουθεί σχεδιάγραμμα για την πλήρη κατανόηση του μοντέλου.



Εικόνα 15: Σχεδιάγραμμα Μοντέλο του Πίδακα

Με δεδομένη την αντικειμενοστρεφή τεχνολογία των εργαλείων που θα χρησιμοποιηθούν αλλά και την αναμενόμενη συμπύκνωση των φάσεων ανάλυσης, σχεδίασης και κωδικοποίησης, υιοθετήθηκε το μοντέλο του πίδακα για την ανάπτυξη της εφαρμογής. Άλλωστε, η δυνατότητα επαναχρησιμοποίησης των παραγώγων της ανάπτυξης επιβεβαιώνεται προτού καν ξεκινήσει η δημιουργία του συγκεκριμένου λογισμικού, αφού γνωρίζουμε εξ' αρχής ότι θα αξιοποιηθούν ήδη υπάρχουσα εργαλεία, ενώ το προϊόν της παρούσας ανάπτυξης θα είναι στη διάθεση της μελλοντικής έρευνας.

5.4 Σχεδιασμός Συστήματος

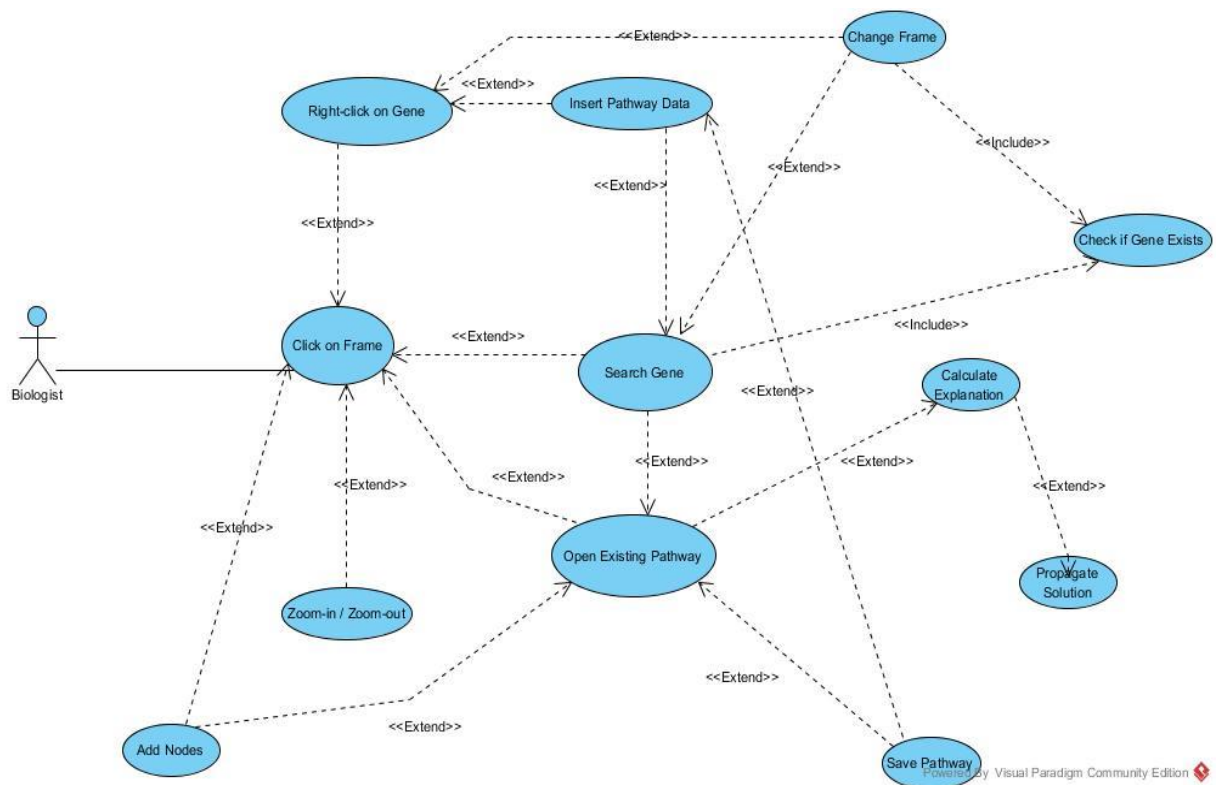
Στις προηγούμενες ενότητες και κεφάλαια, πραγματοποιήθηκαν οι αρχικές φάσεις του κύκλου ζωής του λογισμικού (προκαταρκτική έρευνα, μελέτη σκοπιμότητας, ανάλυση απαιτήσεων).

Αυτή η ενότητα είναι αφιερωμένη στη φάση του σχεδιασμού, ο οποίος συνήθως πραγματοποιείται με την δημιουργία διαγραμμάτων UML.

Η UML (Unified Modeling Language) είναι γλώσσα μοντελοποίησης γενικού σκοπού, ευρέως διαδεδομένη στον τομέα της τεχνολογίας λογισμικού, που σχεδιάστηκε για να παρέχει μια τυποποιημένη μέθοδο απεικόνισης του σχεδιασμού ενός συστήματος λογισμικού. Η μοντελοποίηση είναι απαραίτητη να χρησιμοποιείται και σε υπάρχοντα συστήματα και σε νέα συστήματα. Όσον αφορά τα υπάρχοντα συστήματα, η μοντελοποίηση τους θα βοηθήσει για την εξαγωγή των απαιτήσεων που υπάρχουν για το νέο σύστημα, επίσης είναι ένας εύκολος τρόπος για να εντοπισθούν θετικά και αρνητικά στοιχεία. Όσον αφορά τα νέα συστήματα, χρησιμοποιούνται για να συμφωνηθούν σχεδιαστικές προτάσεις και να εξηγηθούν με πιο τεχνικό τρόπο οι απαιτήσεις των ενδιαφερόμενων. Γίνεται δηλαδή μια τεκμηρίωση της υλοποίησης του συστήματος.[28] [32]

Τα διαγράμματα UML διακρίνονται σε διάφορα είδη. Διακρίνονται σε: Διαγράμματα χρήσης (use case diagrams), Διαγράμματα κλάσεων (class diagrams), Ακολουθιακά διαγράμματα (sequential diagrams), Διαγράμματα συστατικών (component diagrams) και Διαγράμματα δραστηριοτήτων (activity diagrams) [28]

Ένα ενδεικτικό διάγραμμα χρήσης του συστήματος παρουσιάζεται στη συνέχεια, στο οποίο ο ρόλος (actor) είναι ο χρήστης – βιολόγος που θα χρησιμοποιεί το σύστημα. Με τα διαγράμματα χρήσης, εξετάζονται διάφορες περιπτώσεις χρήσης, με διαφορετικά σενάρια, αναλόγως τις επιλογές του χρήστη. Οι σχέσεις επέκτασης («extend») από μια περίπτωση A σε μια άλλη B υποδηλώνουν ότι ένα στιγμιότυπο της περίπτωσης χρήσης B μπορεί να αυξηθεί από τη συμπεριφορά που καθορίζεται από την A, ενώ οι σχέσεις συμπερίληψης («include») υποδηλώνουν ότι ένα στιγμιότυπο της A θα περιέχει και τη συμπεριφορά που καθορίζει η B. Τέλος η σχέση συσχέτισης («association») αφορά τη συμμετοχή του actor στην περίπτωση χρήσης και είναι η μόνη σχέση ανάμεσα σε actor και περιπτώσεις χρήσης. [28]



Εικόνα 16:UML Διάγραμμα Χρήσης του Συστήματος με χρήση του εργαλείου
Visual Paradigm Community Edition

5.5 Υλοποίηση Συστήματος

Η υλοποίηση του συστήματος ξεκίνησε με την μελέτη και ανάλυση του πηγαίου κώδικα της προηγούμενης υλοποίησης, για τον εντοπισμό των αδυναμιών της. Όπως αναφέρθηκε και στην ενότητα 3.4, στον πηγαίο κώδικα των δύο κλάσεων που ορίζουν τις διεργασίες τις Απόπτωσης και του Πολλαπλασιασμού (Apoptosis.java και CellCycle.java), όλες οι διαδρομές αποθήκευσης των resources, input και output είναι απόλυτες (Absolute Paths) και αναφέρονται σε συγκεκριμένο χρήστη (user) ενός H/Y, με αποτέλεσμα τον περιορισμό στα χαρακτηριστικά του συστήματος στο οποίο θα εκτελείται η εφαρμογή και, κατ' επέκταση, της ανεξαρτησία της διεπαφής από τον H/Y στον οποίο λειτουργεί.

Για το λόγο αυτό, σε όλο τον κώδικα των δυο κλάσεων αντικαταστάθηκε το absolute path (π.χ. C:\\Users\\user\\BioCT\\) με το working directory, το οποίο βρίσκεται με την ακόλουθη γραμμή κώδικα :

```
String workDir = System.getProperty("user.dir");
```

Με βάση την παραπάνω αλλαγή μπόρεσαν να οριστούν τα διάφορα αρχεία εισόδου/εξόδου(io), όπως για παράδειγμα :

```
String out4File = workDir + "out4.txt";  
  
...  
FileReader f;  
f = new FileReader(out4File);
```

ενώ αντίστοιχα μπορεί να οριστεί μια πιο συγκεκριμένη δομή στην οργάνωση των αρχείων, με υποφακέλους για την αποθήκευση των αρχείων io, images κλπ, όπως για παράδειγμα :

```
String out4File = workDir + "\\Output\\out4.txt";
```

Επίσης, στην προηγούμενη υλοποίηση δημιουργούνταν από το GraphViz, αρχεία εικόνας png, τα οποία δεν ήταν επεξεργάσιμα. Σε αυτή την έκδοση, εκτός των png αρχείων, παράγονται και αρχεία svg, τα οποία προσφέρουν τη δυνατότητα επεξεργασίας μέσω κατάλληλου viewer, όπως ο ZGRViewer :

```
String type1 = "png";  
String type2 = "plain";  
String type3 = "svg";  
File out2 = new File(out2File + type2); // out.txt in this example  
File out1 = new File(outImageFile + type1); // out.png in this example  
File out3 = new File(outImageFile + type3); // out.svg in this example  
gv.writeGraphToFile(gv.getGraph(gv.getDotSource()), type1, out1);  
gv.writeGraphToFile(gv.getGraph(gv.getDotSource()), type2, out2);  
gv.writeGraphToFile(gv.getGraph(gv.getDotSource()), type3, out3);
```

Στο Παράρτημα Γ, παρατίθεται ο συνολικός κώδικας της κλάσης του GraphViz στην οποία έχει προστεθεί κώδικας (σημειωμένος με έντονα γράμματα) για τον έλεγχο της διαδρομής του προγράμματος dot.exe, από τον φάκελο εγκατάστασης του GraphViz. Πιο συγκεκριμένα, ελέγχονται τρεις πιθανές διαδρομές ανάλογα με το λειτουργικό

σύστημα και την έκδοσή του, και με την προϋπόθεση ότι η εγκατάσταση έχει γίνει στον προεπιλεγμένο φάκελο :

```
// Win 64bit
String DOT1 = "C:\\Program Files (x86)\\Graphviz2.30\\bin\\dot.exe";
// Win 32bit
String DOT2 = "C:\\Program Files\\Graphviz2.30\\bin\\dot.exe";
// Linux
String DOT3 = "/usr/bin/dot";
```

Επιπλέον, γίνεται παραμετρικά η εύρεση του φακέλου Temp στον οποίο γίνονται οι προσωρινές αποθηκεύσεις αρχείων από το GraphViz :

```
private static String workDir = System.getProperty("user.dir");
private static String TEMP_DIR = workDir+"\\Temp";
```

Αρκετός χρόνος και μεγάλη προσοχή αφιερώθηκε στην ενσωμάτωση όλων των απαραίτητων βιβλιοθηκών που αφορούν τον ZGRViewer μέσα στο project του BioCLC. Αφού ενσωματώθηκαν οι βιβλιοθήκες, ουσιαστικά δημιουργήθηκε ένα σύνολο κλάσεων με το πρόθεμα Bio στο package που περιέχει τις αρχικές κλάσεις του ZGRViewer, ώστε να υλοποιηθεί μια παραλλαγή του, ο BioViewer, χωρίς να αλλάξει τις αρχικές κλάσεις.

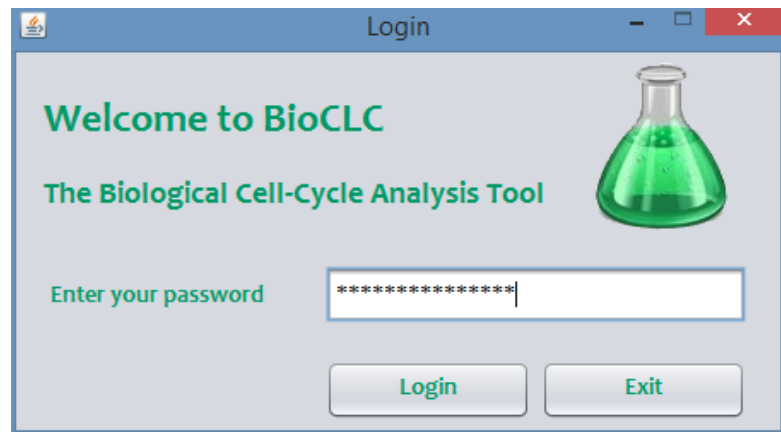
Στον BioViewer αφαιρέθηκαν κάποιες λειτουργίες που είτε δεν ήταν απαραίτητες είτε δεν συνεργάζονταν αποτελεσματικά με το υπάρχον περιβάλλον και τροποποιήθηκαν κάποιες άλλες (για παράδειγμα, όπως θα δούμε στη συνέχεια, το riemenu για το δεξί κλικ).

Το επόμενο σημαντικό μέρος της υλοποίησης – αναβάθμισης, αφορούσε στον επανασχεδιασμό των οθονών και την προσθήκη ορισμένων οθονών, για την προσθαφαίρεση κόμβων και ακμών. Στις ακόλουθες υποενότητες παρουσιάζεται το σύνολο των οθονών που σχεδιάστηκαν εξ'αρχής ή για πρώτη φορά σε αυτή την έκδοση.

5.5.1 User Interface Τελευταίας Έκδοσης

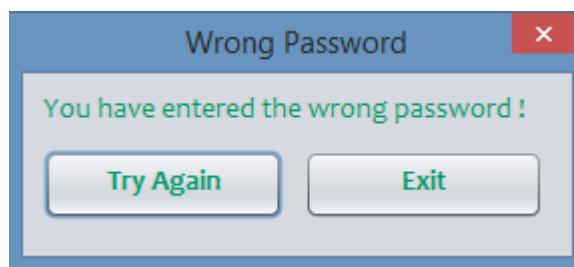
5.5.1.1 Ασφαλής Είσοδος στο Σύστημα

Η πρώτη οθόνη με την οποία ο χρήστης μας αλληλεπιδρά με το σύστημα, είναι ένα παράθυρο στο οποίο πρέπει να δώσει κωδικό πρόσβασης, ο οποίος του έχει δοθεί εκ των προτέρων (είναι διαθέσιμος στο README.TXT, στο Παράρτημα Β).



Εικόνα 17: Οθόνη Πρόσβασης στο Σύστημα

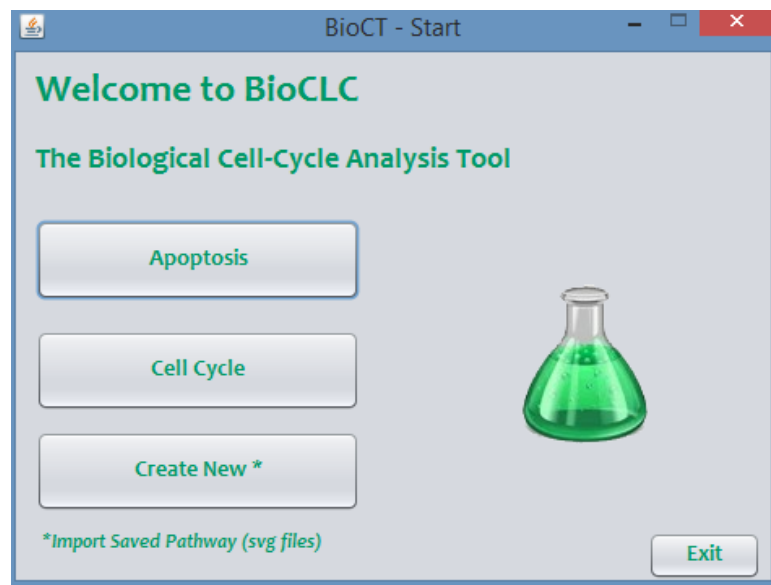
Αν ο κωδικός είναι σωστός, πραγματοποιείται είσοδος στο σύστημα αλλιώς έχει δώσει λάθος κωδικό και εμφανίζεται το ανάλογο μήνυμα λάθους



Εικόνα 18 : Οθόνη με μήνυμα λάθους για την εισαγωγή κωδικού πρόσβασης

5.5.1.2 Αρχική Οθόνη

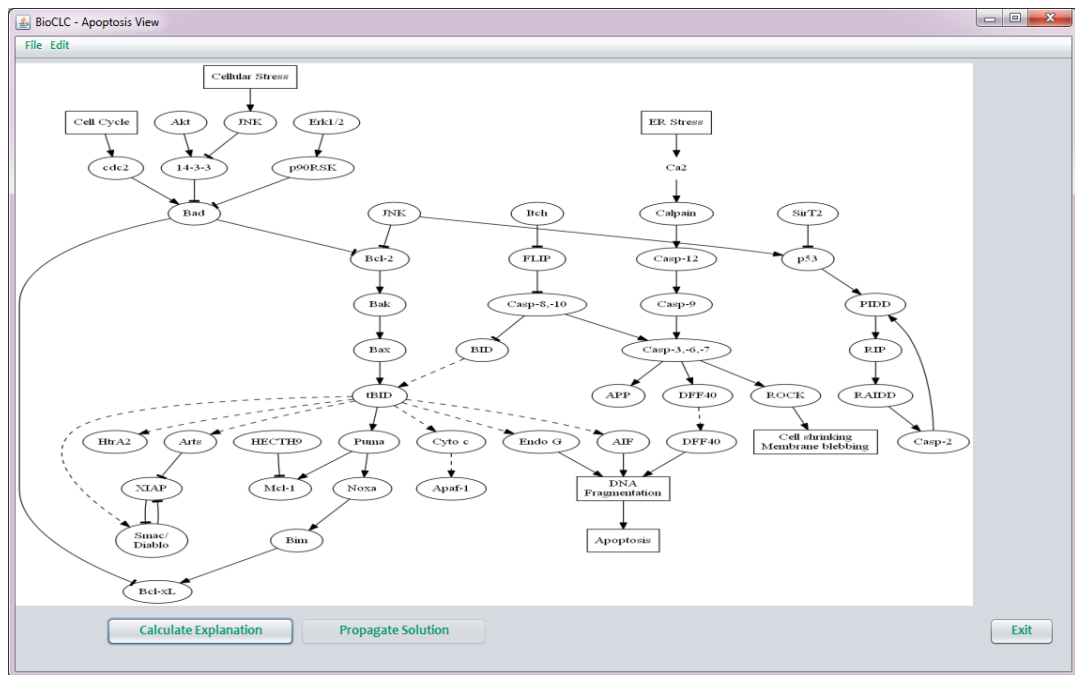
Στη συνέχεια, αφού ο χρήστης έχει κάνει είσοδο με επιτυχία, εμφανίζεται η αρχική οθόνη του συστήματος με τις τρεις βασικές επιλογές που περιγράφηκαν στις απαιτήσεις : την επιλογή για την εμφάνιση του μονοπατιού της απόπτωσης, την επιλογή για την εμφάνιση του μονοπατιού του πολλαπλασιασμού, την επιλογή για δημιουργία νέου μονοπατιού και τέλος η επιλογή για έξοδο.



Εικόνα 19 : Αρχική Οθόνη του Συστήματος

5.5.1.3 Οθόνη Απόπτωσης – Αρχική εικόνα

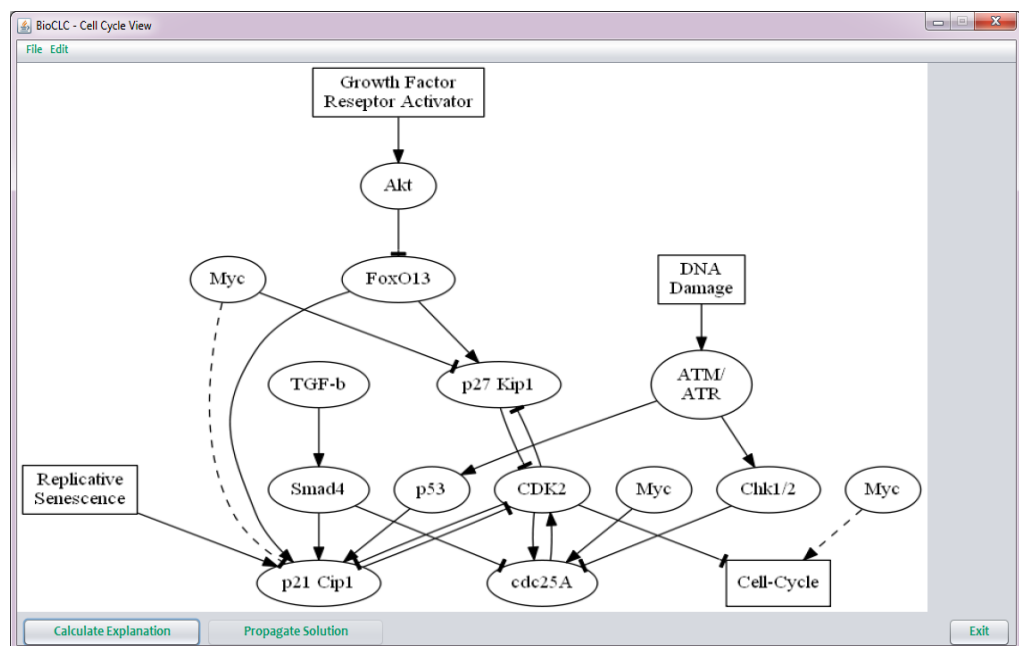
Στην οθόνη με το μονοπάτι για την απόπτωσης, απεικονίζεται το μονοπάτι με ξεκάθαρο τρόπο. Παρατηρούμε από την εικόνα που παρατίθεται πιο κάτω ότι οι διεργασίες απεικονίζονται με ορθογώνιο σχήμα, οι πρωτεΐνες με οβάλ σχήμα και οι άλλες ουσιών χωρίς περίγραμμα, όπως μας ζητήθηκε. Οι συνδέσεις των συστατικών μελών ξεχωρίζουν όταν είναι ενεργητικές κατευθυνόμενες ή ανασταλτικές κατευθυνόμενες. Επιπρόσθετα, στο πάνω μέρος της οθόνης υπάρχει το βασικό μενού επιλογών, στο κάτω μέρος της οθόνης βλέπουμε το κουμπί 'Calculate Explanation' και 'Propagate Solution' (οι λειτουργίες τους θα επεξηγηθούν σε μετέπειτα υποκεφάλαιο) και τέλος, κάτω δεξιά το κουμπί για έξοδο από την οθόνη.



Εικόνα 20: Οθόνη Απόπτωσης-Αρχική Οθόνη (πριν τα πειραματικά δεδομένα)

5.5.1.4 Οθόνη Πολλαπλασιασμού – Αρχική εικόνα

Στην οθόνη με το μονοπάτι για τον πολλαπλασιασμό οι επιλογές που έχει ο χρήστης και το μενού εμφανίζονται με τον ίδιο τρόπο όπως και στο μονοπάτι για την απόπτωση.



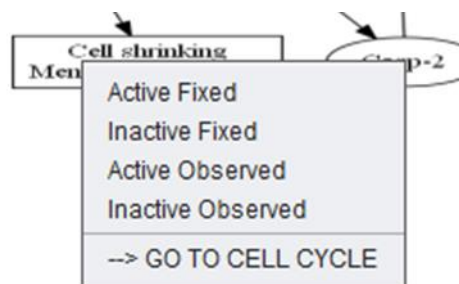
Εικόνα 21 : Οθόνη Πολλαπλασιασμού-Αρχική Οθόνη (πριν τα πειραματικά δεδομένα)

5.5.1.5 Εισαγωγή Πειραματικών δεδομένων και Αντιστοιχία Μεταξύ Οθονών

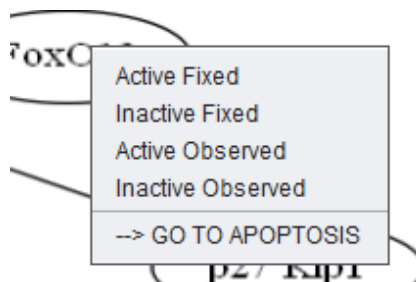
Η εισαγωγή των πειραματικών δεδομένων είναι εφικτή με δεξί κλικ πάνω στα γονίδια. Υπάρχουν οι τέσσερις επιλογές :

- active fixed (ενεργή και το ξέρουμε εκ των προτέρων),
- inactive fixed (απενεργοποιημένη και το ξέρουμε εκ των προτέρων, δηλαδή σαν να απουσιάζει από το κύτταρο),
- active observed (ενεργή μετά από το πείραμα),
- inactive observed (απενεργοποιημένη μετά από το πείραμα).

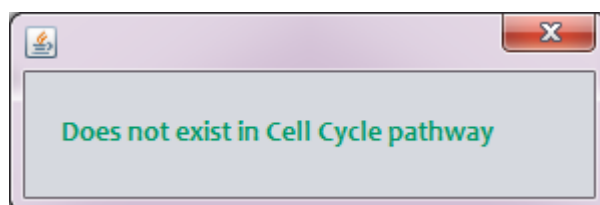
Ακόμα, βλέπουμε και μια τέταρτη επιλογή η οποία είναι ικανή να μας παραπέμψει στο ίδιο συστατικό μέλος της αντίστοιχη οθόνη. Όταν γίνει η παραπομπή, το συστατικό μέλος εμφανίζεται με διαφορετικό χρώμα (κίτρινο) για εύκολο εντοπισμό από τον χρήστη μας. Σε περίπτωση όπου το συστατικό μέλος δεν υπάρχει στην αντίστοιχη οθόνη, εμφανίζεται το ανάλογο προειδοποιητικό μήνυμα και παραμένουμε στην αρχική εικόνα που είμαστε, δηλαδή δεν γίνεται η παραπομπή.



Εικόνα 22: Μενού επιλογών για εισαγωγή πειραματικών δεδομένων στην οθόνη της απόπτωσης



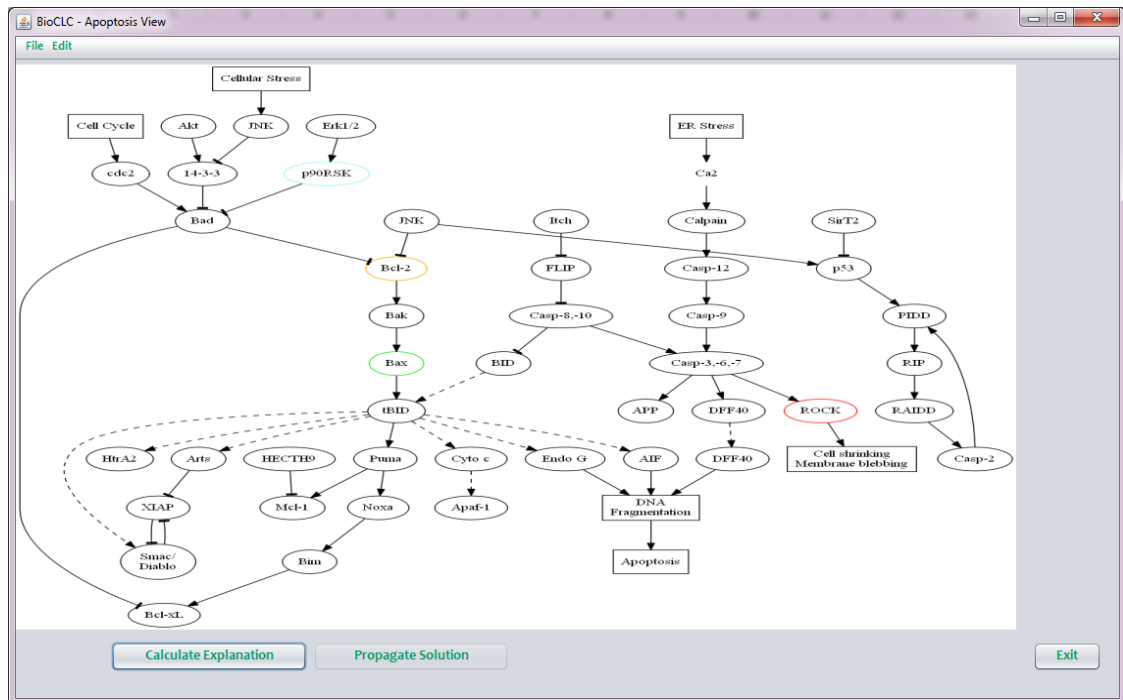
Εικόνα 23: Μενού επιλογών για εισαγωγή πειραματικών δεδομένων στην οθόνη της απόπτωσης



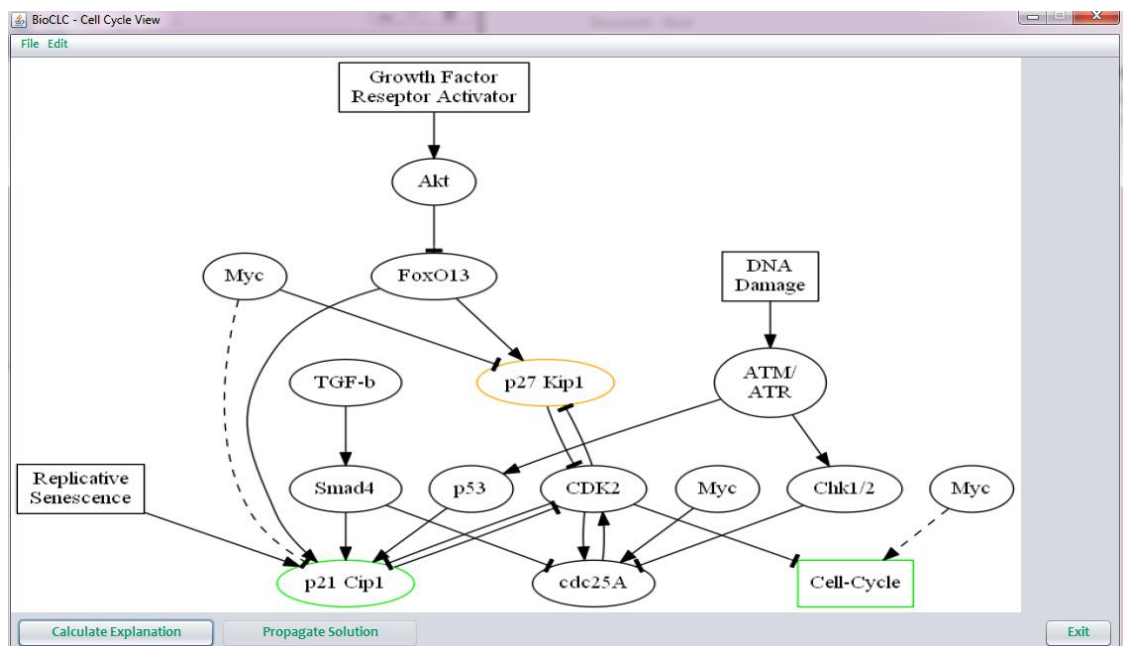
Εικόνα 24: Μήνυμα λάθους, θα εμφανίζεται εάν στην οθόνη του πολλαπλασιασμού δεν υπάρχει το ίδιο συστατικό μέλος που επιλέξαμε για παραπομπή στην οθόνη της απόπτωσης

5.5.1.6 Οθόνες Απόπτωσης και Πολλαπλασιασμού Μετά την Εισαγωγή των Πειραματικών Δεδομένων

Σύμφωνα με τις απαιτήσεις του χρήστη, μετά την εισαγωγή των πειραματικών δεδομένων, τα περιγράμματα των κόμβων μέσα στο κυτταρικό δίκτυο αλλάζουν χρώματα ανάλογα με τις επιλογές που έχει κάνει ο ίδιος χρήστης. Αν το γονίδιο είναι active fixed, το περίγραμμα του γίνεται γαλάζιο, αν είναι inactive fixed το περίγραμμα του γίνεται πορτοκαλί, αν είναι active observed το περίγραμμα του γίνεται πράσινο και αν είναι inactive observed, το περίγραμμα του γίνεται κόκκινο.



Εικόνα 25 : Οθόνη Απόπτωσης, μετά από εισαγωγή των πειραματικών δεδομένων (p90RSK : active fixed, Bcl-2 : inactive fixed, Bax :active observed, ROCK : inactive observed)



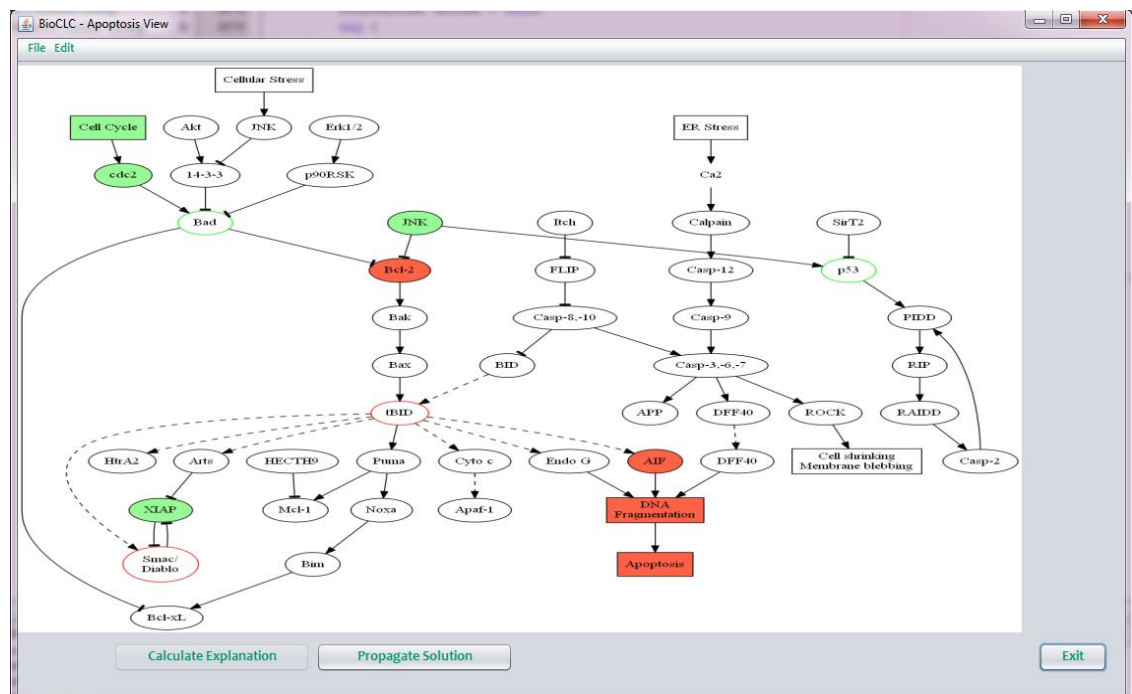
Εικόνα 26 : Οθόνη Πολλαπλασιασμού μετά από εισαγωγή των πειραματικών δεδομένων (p21 Cip1: active observed, p21Kip1 : active fixed, Cell-Cycle : active observed)

5.5.1.7 Οθόνες Απόπτωσης και Πολλαπλασιασμού Μετά τον Υπολογισμό Επεξήγησης (Calculate Explanation)

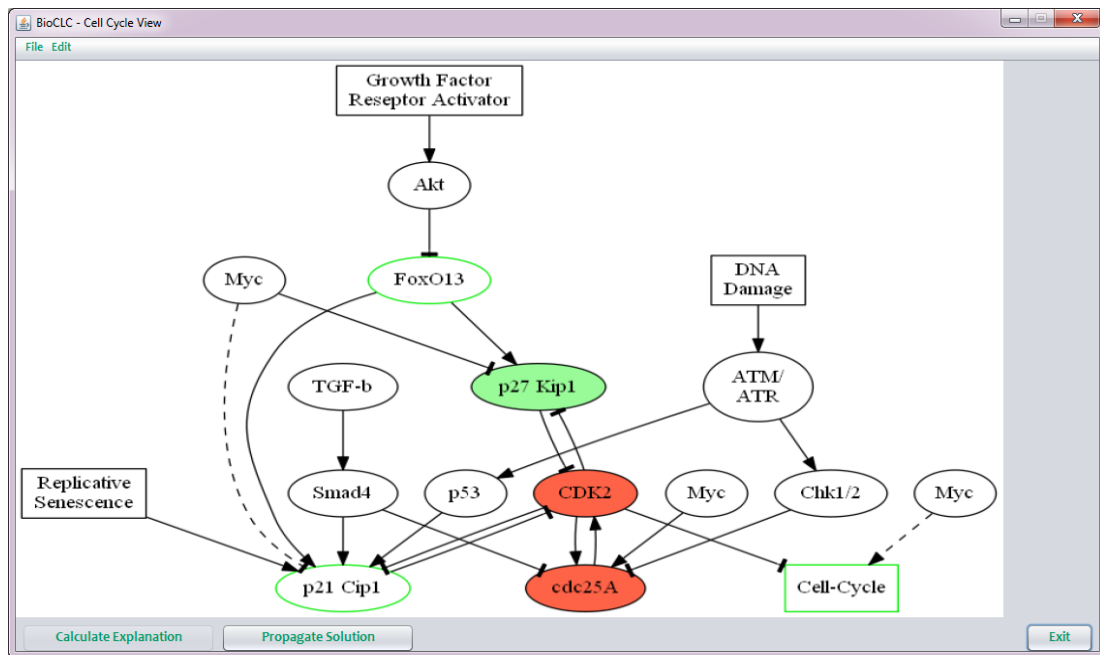
Στη φάση αυτή, θα γινόταν εισαγωγή του Απαγωγικού Λογικού Προγραμματισμού, εντούτοις αυτό το σημείο δε θα μας απασχολήσει σε αυτή τη διπλωματικής εργασίας. Με το πάτημα του κουμπιού ‘Calculate Explanation’ εμφανίζεται ένα ενδεικτικό αποτέλεσμα εμφάνισης της υπόθεσης. Θεωρήσαμε σημαντικό να εμφανίζεται αυτό το βήμα έτσι ώστε η υλοποίηση μας να είναι όσο το δυνατό πιο κοντά στην πλήρως ολοκληρωμένη διεπαφή.

Το σενάριο προβλέπει τον εξής συνδυασμό καταστάσεων γονιδίων : 53: active fixed, Bad: active observed, tBID: inactive observed, Smac/Diablo: inactive observed.

Όσα γονίδια επηρεάζονται από τα πειραματικά δεδομένα που έδωσε ο χρήστης (υποθετικά μέχρι στιγμής στην παρούσα εργασία), τότε θα εμφανίζονται με διαφορετικό χρώμα, δηλαδή θα εμφανίζονται γεμισμένα με πράσινο χρώμα εάν ενεργοποιηθούν και γεμισμένα με κόκκινο αν ήταν απενεργοποιημένα.



Εικόνα 27: Οθόνη Απόπτωσης, μετά από τον υπολογισμό της επεξήγησης



Εικόνα 28 : Οθόνη Πολλαπλασιασμού, μετά από τον υπολογισμό της επεξήγησης

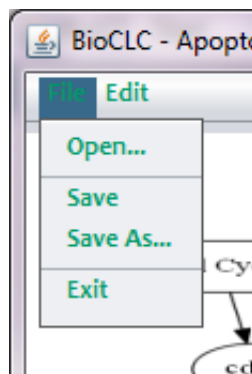
5.5.1.8 Οθόνες Απόπτωσης και Πολλαπλασιασμού Μετά τη Διάδοση του Σήματος (Propagate Solution)

Όπως και στο προηγούμενο σημείο, έτσι και σε αυτό το σημείο, θα έπρεπε να γίνει χρήση του Απαγωγικού Λογικού Προγραμματισμού, για τον υπολογισμό όλων των πιθανών λύσεων μετά από την διάδοση σήματος. Εφόσον δεν έχουμε τις κατάλληλες τεχνολογίες για τον υπολογισμό πραγματικών λύσεων, θα δείξουμε μόνο μια απεικόνιση η οποία θα υπάρχει στο σύστημα για μια πιθανή λύση.

Με αυτό τον τρόπο, επί της ουσίας, ο χρήστης θα μπορεί να βλέπει τη διάδοση του σήματος σε όλα τα πιθανά μονοπάτια και τέλος, να βλέπει όλες τις πιθανές λύσεις.

5.5.1.9 Επιλογές Open και Save

Στο βασικό μενού επιλογών, πάνω αριστερά, στην την επιλογή 'File', υπάρχουν οι επιλογές Open, Save και Save As κατά τις οποίες ο χρήστης μπορεί να ανοίξει και να αποθηκεύσει αντίστοιχα κάποιο πείραμα. Μπορεί να επιλέξει αρχεία από οποιοδήποτε φάκελο. Τα αρχεία που αναγνωρίζει το σύστημα είναι σε μορφή απλού κειμένου (text files) και με την εισαγωγή τους στο σύστημα γίνεται αυτόματη μετάφρασή τους σε .dot αρχεία, τα οποία αναγνωρίζει το εργαλείο Graphviz για να μπορέσουν στη συνέχεια να μεταφραστούν στις ανάλογες εικόνες. Ο χρήστης μπορεί να φυλάξει το πείραμά του με οποιοδήποτε όνομα και σε οποιαδήποτε διαδρομή, φτάνει να έχει extension .txt.



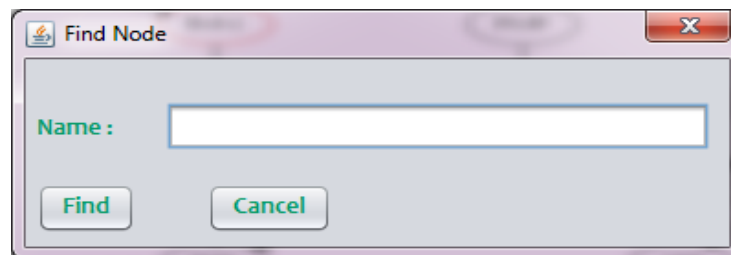
Εικόνα 31: Βασικό Μενού Επιλογών Αρχείων στις Οθόνες Απόπτωσης και Πολλαπλασιασμού

5.5.1.10 Επιλογή Αναζήτησης

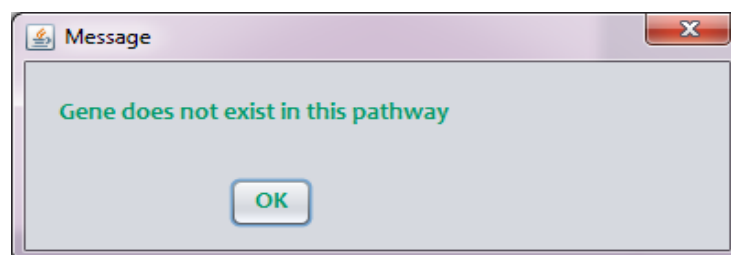
Στο βασικό μενού των οθονών Απόπτωσης και Πολλαπλασιασμού, κάτω από την επιλογή 'Edit' υπάρχει η επιλογή 'Find', η οποία εμφανίζει παράθυρο που ζητά από το χρήστη το όνομα του συστατικού μέλους που θέλει να αναζητήσει και πατώντας το κουμπί Find, αν υπάρχει το υπό αναζήτηση συστατικό μέλος στο μονοπάτι θα εμφανιστεί με κίτρινο χρώμα, ενώ δεν υπάρχει στο μονοπάτι, θα εμφανιστεί το ανάλογο μήνυμα.

Σε περίπτωση που ένα συστατικό μέλος μπορεί να εμφανίζεται περισσότερες από μία φορές, τότε πρέπει να κίτρινίζονται όλοι οι κόμβοι οι οποίοι εμφανίζεται το συστατικό μέλος. Ακόμη, η αναζήτηση δεν είναι ευαίσθητη σε πεζούς ή κεφαλαίους

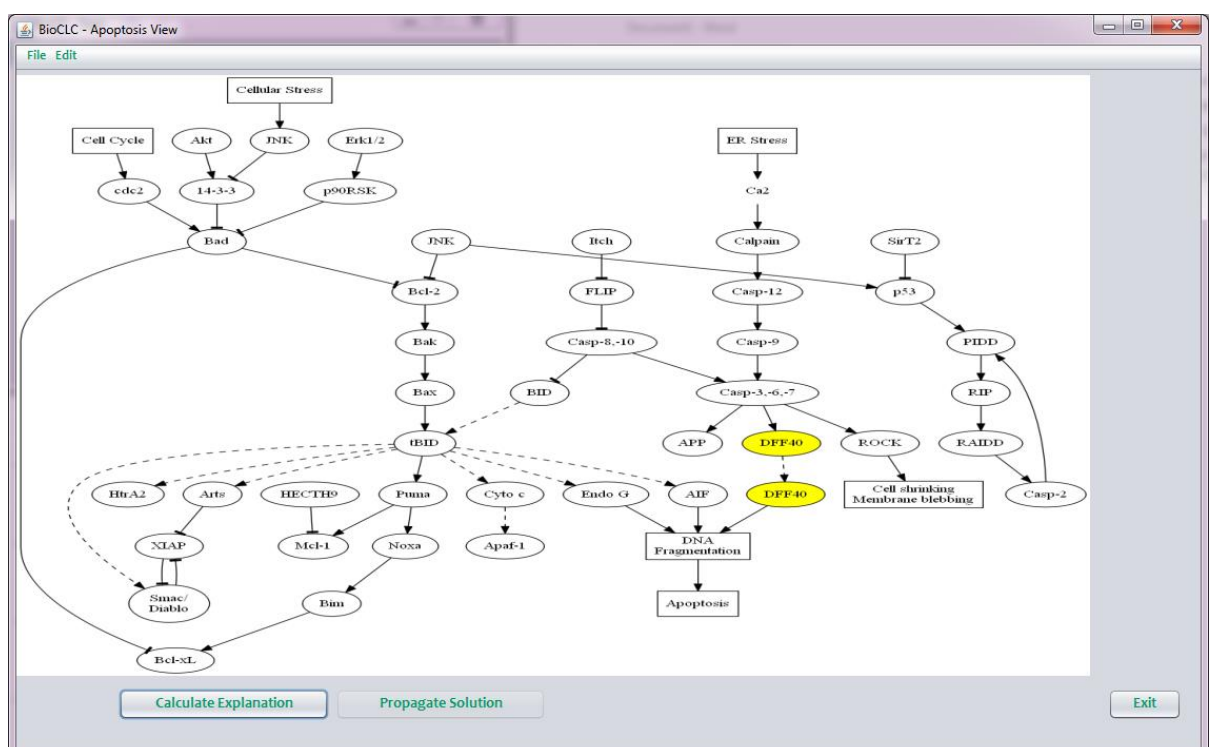
χαρακτήρες (case-insensitive), αλλά ο χρήστης θα πρέπει να δώσει την ακριβή ονομασία του συστατικού μέλους για γίνει δυνατή η εύρεσή του.



Εικόνα 32: Παράθυρο για αναζήτηση πρωτεΐνης, διεργασίας ή ουσίας



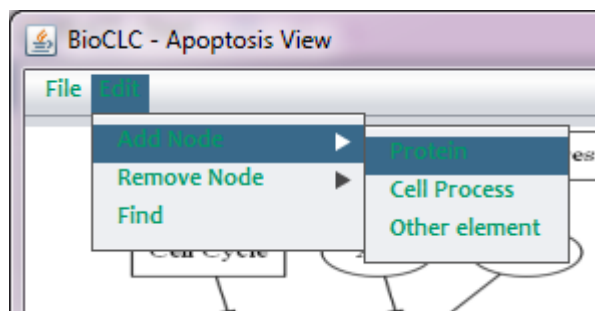
Εικόνα 33: Μήνυμα όταν το συστατικό μέλος που αναζητήθηκε δεν ανήκει στο μονοπάτι



Εικόνα 34: Οθόνη Απόπτωσης όταν αναζητήθηκε σε αυτήν το συστατικό μέλος «DFF40»

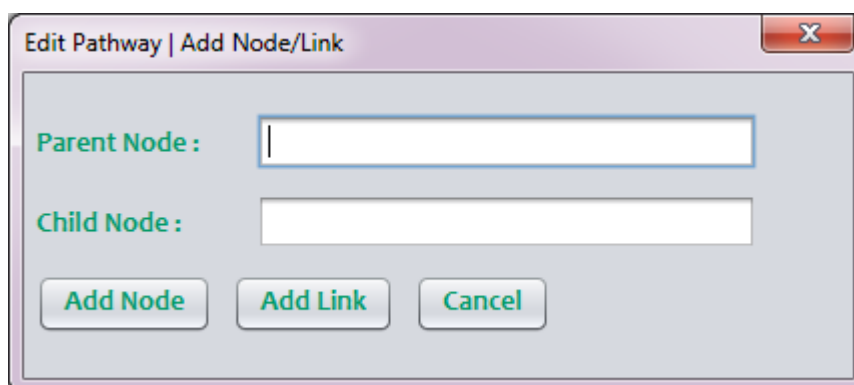
5.5.1.11 Προσθαφαίρεση Κόμβων / Ακμών

Στο βασικό μενού των οθονών Απόπτωσης και Πολλαπλασιασμού, κάτω από την επιλογή Edit, έχουν προστεθεί δύο επιπλέον υπομενού, τα «Add Node» και «Remove Node», με αντίστοιχες επιλογές για «Protein», «Cell Process» ή «Other Element» :

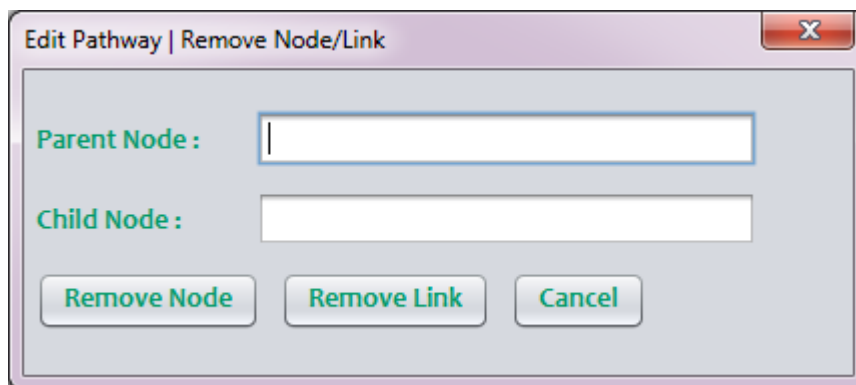


Εικόνα 35 : Μενού Προσθήκης Κόμβων/Ακμών

Με επιλογή είτε της Προσθήκης είτε της Αφαίρεσης εμφανίζεται η αντίστοιχη οθόνη. Στην περίπτωση της εισαγωγής κόμβου (Add Node) ο χρήστης θα πρέπει να εισάγει το Νέο Κόμβο στο πεδίο Child Node και τον υπάρχοντα κόμβο στον οποίο θα προστεθεί στο πεδίο Parent Node. Στην περίπτωση εισαγωγής ακμής (Add Link), συμπληρώνονται και τα δυο πεδία για να περιγράψουν την ακμή που τα συνδέει. Αν το Child Node δεν υπάρχει, θα δημιουργηθεί. Το Parent Node θα πρέπει να υπάρχει υποχρεωτικά. Με Cancel ο χρήστης εξέρχεται από τη συγκεκριμένη οθόνη.



Εικόνα 36: Οθόνη Προσθήκης Κόμβων/Ακμών



Εικόνα 37: Οθόνη Αφαίρεσης Κόμβων/Ακμών

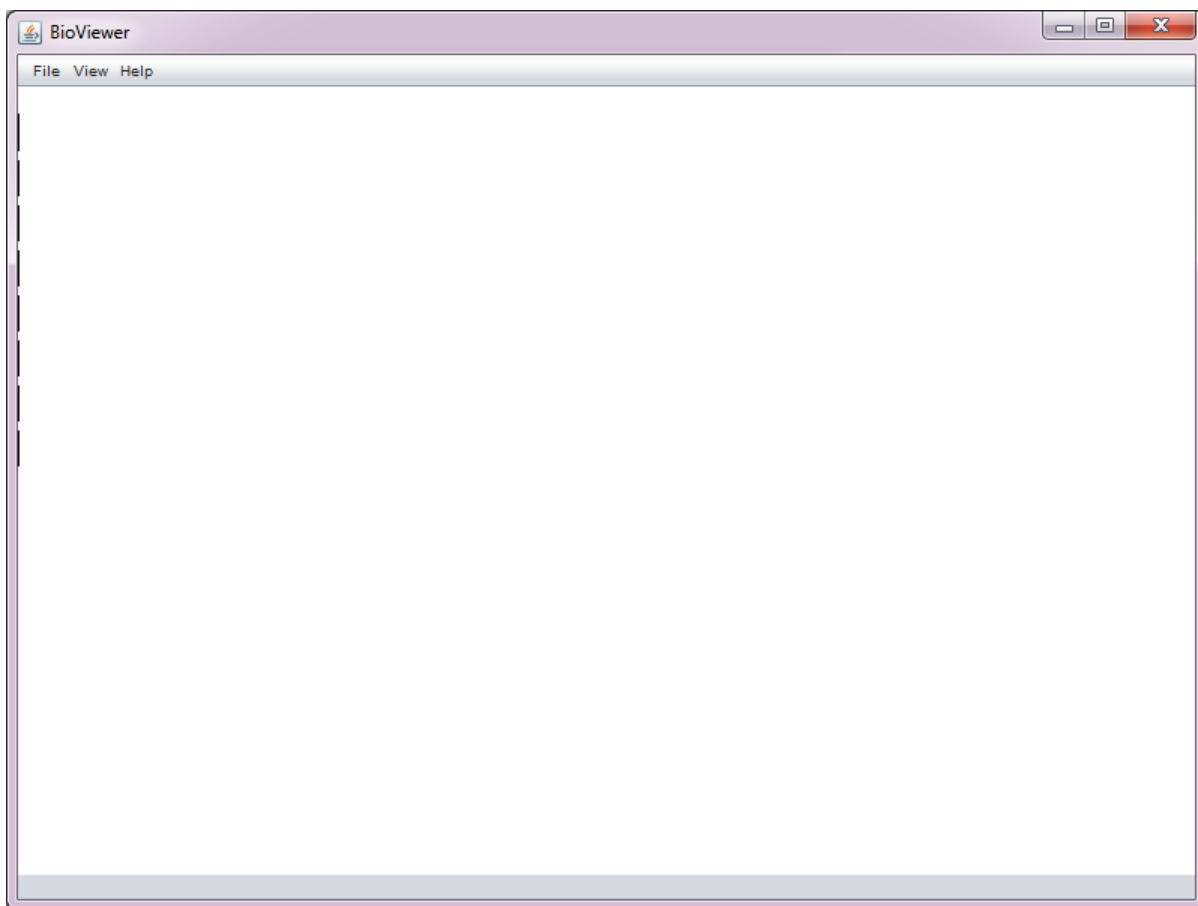
Για λόγους που θα επεξηγηθούν στη συνέχεια, η προαναφερθείσα λειτουργικότητα δεν υλοποιήθηκε πλήρως.

5.5.1.12 Δημιουργία Νέου Μονοπατιού / Επεξεργασία Υπάρχοντος

Αν στην κύρια Οθόνη ο Χρήστης επιλέξει το κουμπί «Create New*», τότε δεν θα προβληθεί οθόνη όπως οι προηγούμενες της Απόπτωσης και του Πολλαπλασιασμού, αλλά, θα ανοίξει ο BioViewer, ένα instance του ZGRViewer, τροποποιημένο για τις ανάγκες της συγκεκριμένης εφαρμογής (οι παρεμβάσεις στον κώδικα του ZGRViewer αναλύθηκαν νωρίτερα).

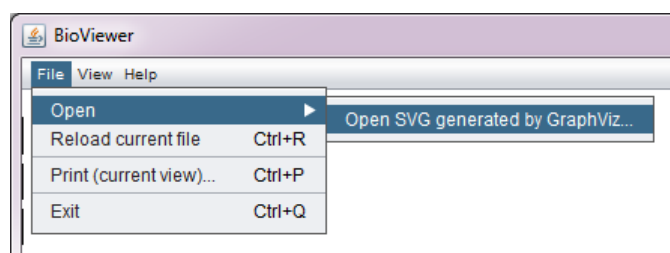
Στον συγκεκριμένο viewer, στην παρούσα φάση, υλοποιήθηκε μόνο η προβολή και επεξεργασία, αρχείων svg που έχουν παραχθεί από το GraphViz, μέσα από τις λειτουργίες της Απόπτωσης και του Πολλαπλασιασμού που αναφέρθηκαν προηγουμένως. Όπως έχει ήδη επεξηγηθεί, κάθε φορά που η εφαρμογή παράγει αρχεία εικόνων βάσει του αρχείου εισόδου .dot, τα παράγει πλέον σε δυο format, σε png για προβολή στις δυο προηγούμενες οθόνες και σε svg για προβολή μέσω του BioViewer.

Η οθόνη αρχικά είναι κενή, βάσει και των απαιτήσεων του χρήστη :

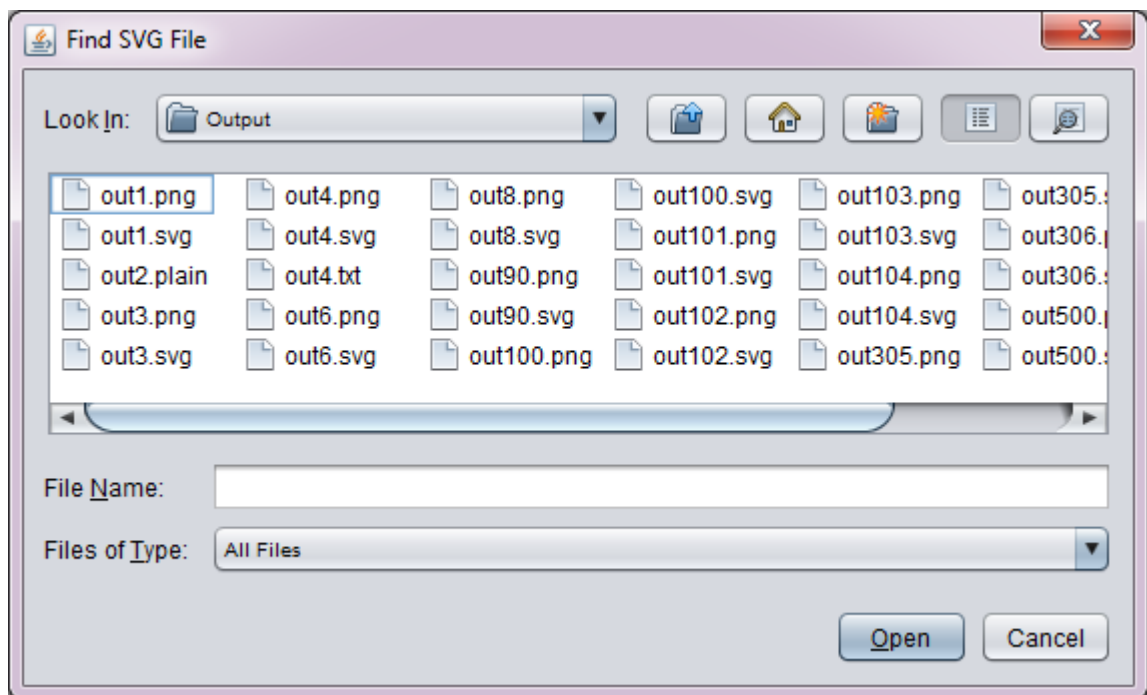


Εικόνα 38: Οθόνη (Κενή Αρχικά) για Εισαγωγή Υπαρχόντων Μονοπατιών

Επιλέγοντας από το Μενού File → Open → Open SVG generated by GraphViz, και αναζητώντας το αρχείο που θέλουμε στον φάκελο Output της εφαρμογής, ανοίγουμε όποιο ήδη υπάρχον μονοπάτι επιθυμούμε να επεξεργαστούμε :

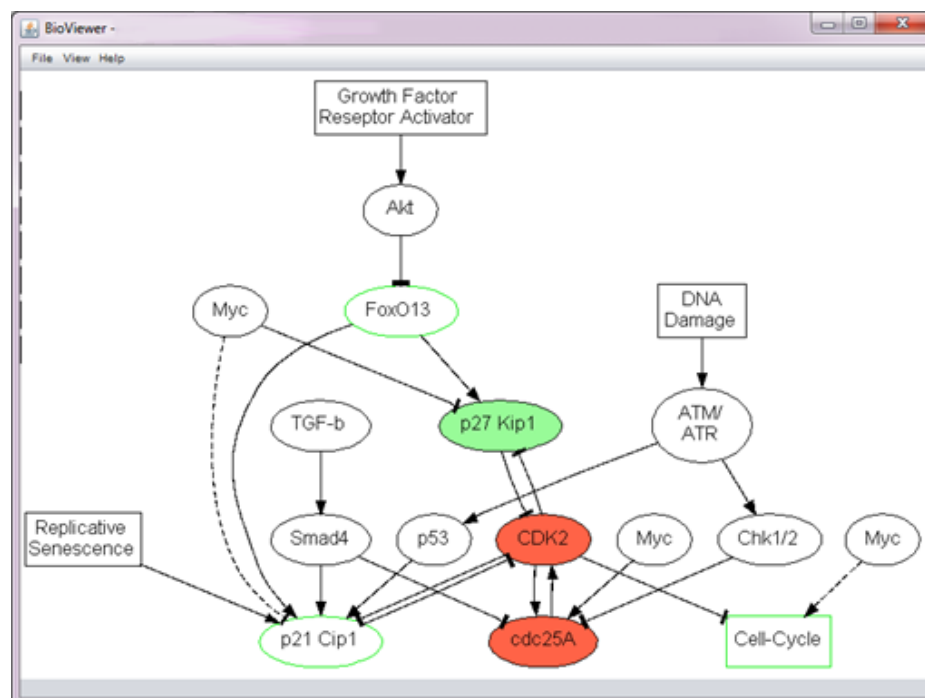


Εικόνα 39: Βήμα για Ανάκτηση Υπαρχόντων Μονοπατιών



Εικόνα 40: Επιλογή Αρχείων Μονοπατιών, τύπου SVG

Ανοίγοντας το αρχείο, προβάλλεται στον viewer και τότε είμαστε σε θέση να εκμεταλλευτούμε τα χαρακτηριστικά πλεονεκτήματά του :

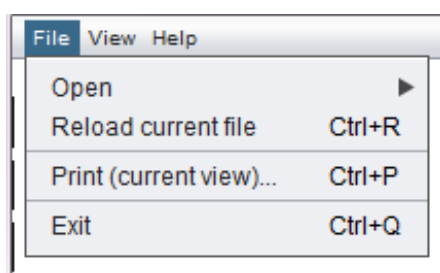


Εικόνα 41: Προβολή Κυτταρικού Δικτύου στο BioViewer

Στην επόμενη ενότητα, γίνεται λεπτομερής παρουσίαση του Bioviewer.

5.5.2 BioViewer

Προτού αναλυθούν τα χαρακτηριστικά και οι λειτουργίες του BioViewer είναι χρήσιμη μια περιγραφή των μενού :



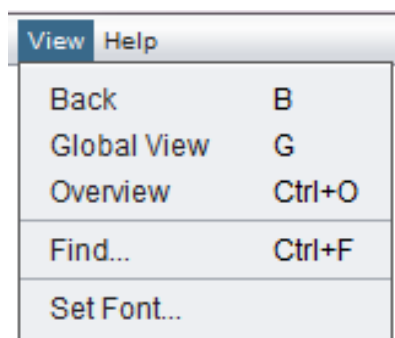
Εικόνα 42: Το Μενού 'File' στο BioViewer

Όπως προαναφέρθηκε η πρώτη επιλογή αφορά στο άνοιγμα (Open) ενός αρχείου. Στη συνέχεια δίνεται η δυνατότητα επαναφόρτωσης του αρχείου (Reload current file), σε περίπτωση που έχουμε πραγματοποιήσει αλλαγές τις οποίες δεν θέλουμε να διατηρήσουμε και θέλουμε να ξαναπροβάλλουμε το ίδιο αρχείο.

Δίνεται επίσης η δυνατότητα εκτύπωσης της τρέχουσας προβολής (Print current view), του γράφου δηλαδή όπως έχει εμφανιστεί και τροποποιηθεί μετά το άνοιγμα του αρχείου.

Τέλος υπάρχει η επιλογή της εξόδου, για επιστροφή στην κεντρική οθόνη του BioCLC.

Στο επόμενο μενού, στο View έχουμε επιλογές για την εμφάνιση και προβολή του γράφου :

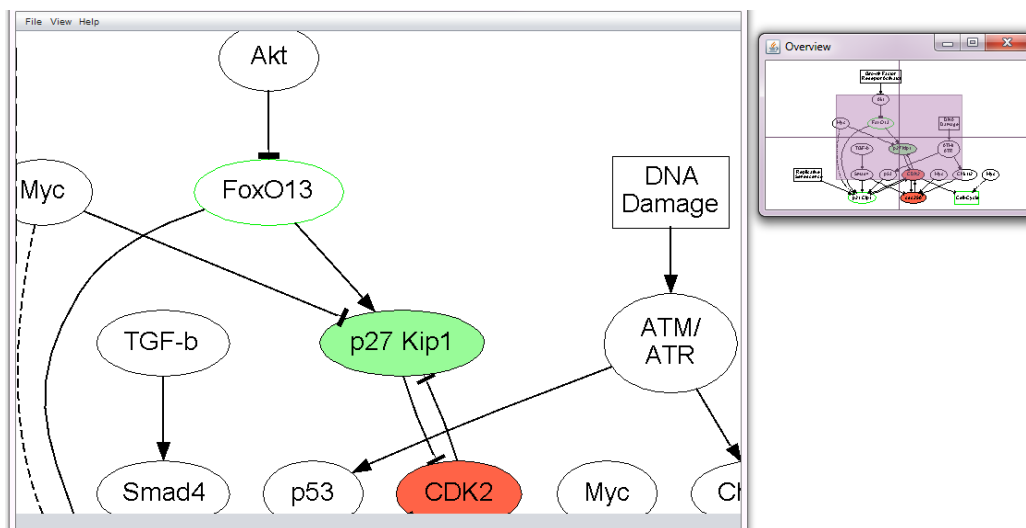


Εικόνα 43: Το Μενού 'View' στο BioViewer

Η επιλογή Back (keyboard shortcut B) αναφέρεται στην επιστροφή σε προηγούμενη προβολή. Αυτό είναι ιδιαίτερα χρήσιμο σε περιπτώσεις αναζήτησης γονιδίων, όπου σε πολλαπλές εμφανίσεις, ο viewer εστιάζει σε ένα κάθε φορά και δίνει δυνατότητα περιήγησης σε όλα τα γονίδια που έχουν βρεθεί με το συγκεκριμένο όνομα.

Η επιλογή Global View (keyboard shortcut G) επαναφέρει την πλήρη προβολή του γράφου και είναι χρήσιμη για εύκολη έξοδο από κατάσταση Zoom.

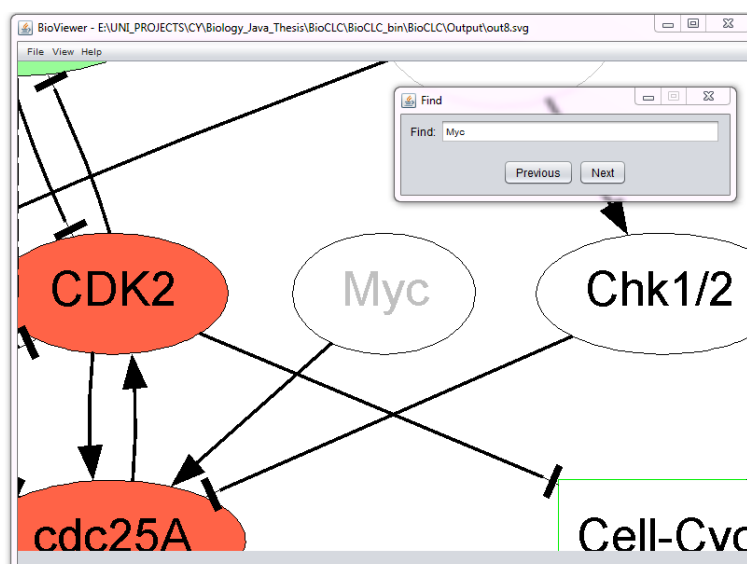
Η επιλογή Overview (keyboard shortcut Ctrl+O) ανοίγει ένα επιπλέον παράθυρο που ανά πάσα στιγμή, και ανεξαρτήτως της επεξεργασίας που κάνει ο χρήστης στο κυτταρικό δίκτυο και του του επιπέδου zoom στο οποίο δουλεύει, προβάλλει την πλήρη εικόνα του γράφου, δείχνοντας μάλιστα την περιοχή στην οποία είμαστε τοποθετημένοι :



Εικόνα 44: Η Επιλογή 'Overview', πολύ χρήσιμη σε καταστάσεις ZOOM

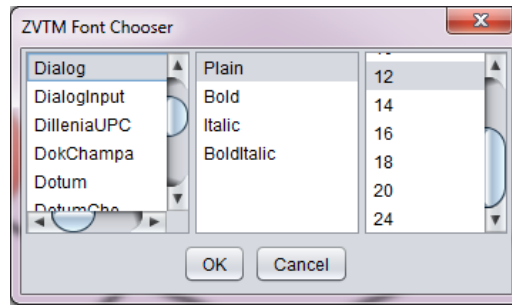
Η επιλογή Find, ανοίγει παράθυρο αναζήτησης στο οποίο μπορούμε να αναζητήσουμε το όνομα του κόμβου, εδώ γονιδίου, που μας ενδιαφέρει.

Σε περίπτωση πολλαπλών αποτελεσμάτων, ο viewer εστιάζει σε ένα κόμβο κάθε φορά και με τα κουμπιά Previous και Next είναι δυνατή η περιήγηση ανάμεσα σε όλους τους κόμβους που βρέθηκαν με το όνομα αυτό :



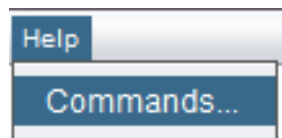
Εικόνα 45: Η Επιλογή 'Find' για Αναζήτηση

Τέλος η επιλογή Set Font... ανοίγει παράθυρο επιλογής γραμματοσειράς :

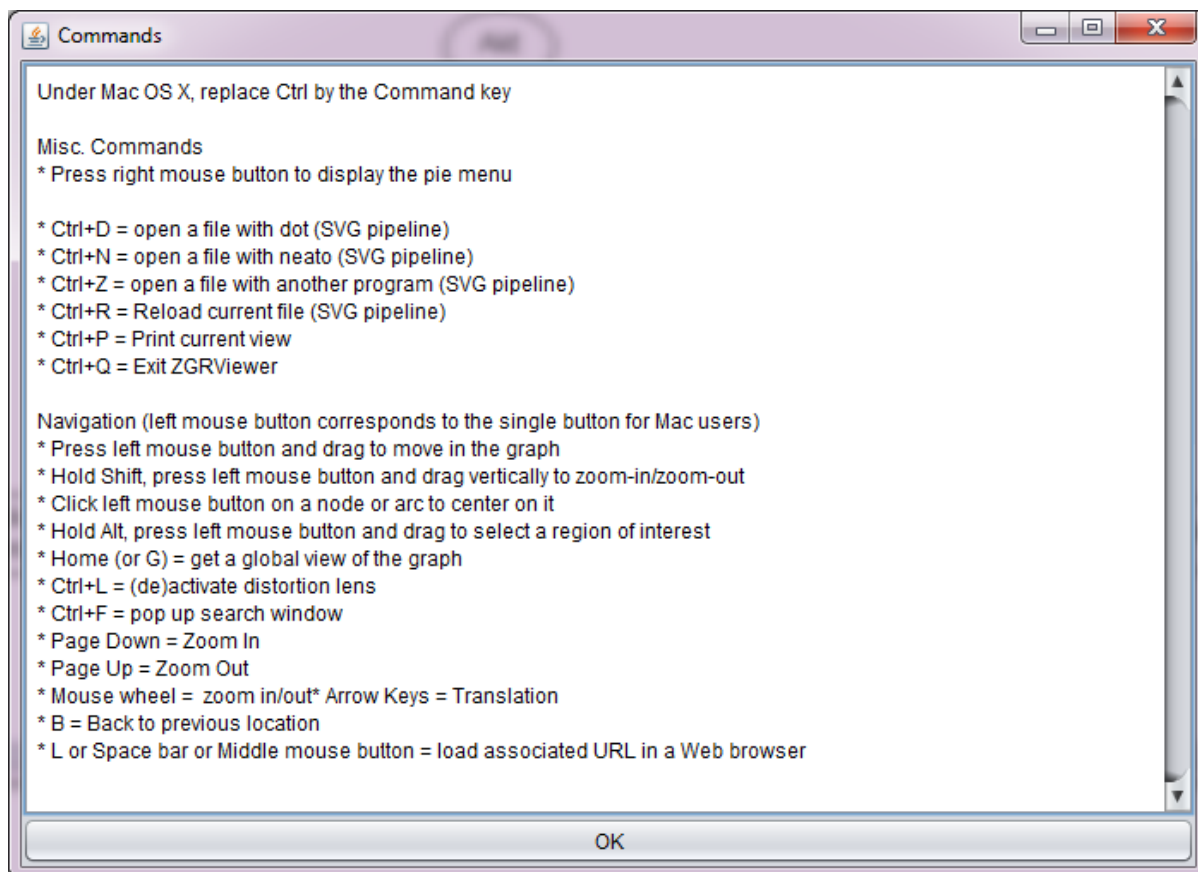


**Εικόνα 46: Η Επιλογή ‘SetFont’
διαθέσιμη από το ZVTM toolkit**

Στο Μενού Help, με την επιλογή Commands ανοίγει παράθυρο με οδηγίες για τις λειτουργίες και τις συντομεύσεις πληκτρολογίου (keyboard shortcuts) του viewer :

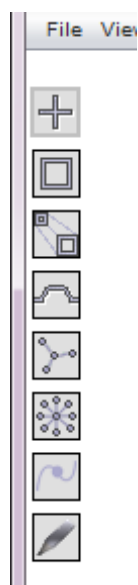


Εικόνα 47: Το Μενού ‘Help’



Εικόνα 48: Οι Οδηγίες για Εντολές(Commands) που χρησιμοποιούνται στην εφαρμογή

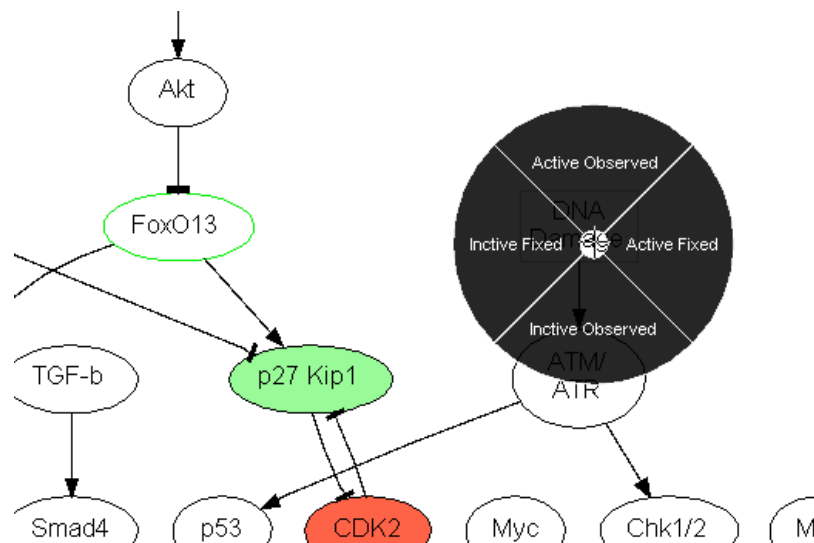
Ένα πολύ σημαντικό χαρακτηριστικό του viewer είναι το toolbar που εμφανίζεται στην αριστερή πλευρά του, όταν μετακινήσουμε κατάλληλα το ποντίκι στο άκρο :



Εικόνα 49: Toolbar για διαφορετικές δυνατότητες προβολής και επεξεργασίας

Ουσιαστικά τα εργαλεία του toolbar υλοποιούν τα χαρακτηριστικά που αναφέρθηκαν στην ενότητα 4.4.2 για τον ZGRViewer και αφορούν διαφορετικές δυνατότητες προβολής, περιήγησης, εστίασης και επεξεργασίας του εκάστοτε γράφου.

Με παρατεταμένο αριστερό κλικ και μετακίνηση του ποντικιού είναι δυνατή η περιήγηση σε όλο το δίκτυο, ενώ με δεξί κλικ στον γράφο εμφανίζεται το μενού των καταστάσεων των γονιδίων, η λειτουργικότητα του οποίου όμως δεν έχει ολοκληρωθεί σε αυτή την έκδοση :



Εικόνα 50: 'PieMenu' το οποίο εμφανίζεται με δεξί Κλικ

Κεφάλαιο 6: Έλεγχος – Αξιολόγηση Συστήματος

6.1 Εισαγωγή	69
6.2 Έλεγχος Συστήματος	69
6.3 Αξιολόγηση - Αποτελέσματα	70

6.1 Εισαγωγή

Ο έλεγχος και η αξιολόγηση του Πληροφοριακού Συστήματος, είναι χωρίς αμφιβολία απαραίτητα στάδια πριν την τελική έκδοσή του συστήματος και την παράδοση του στους τελικούς χρήστες. Προτού το παραδώσουμε στους χρήστες, πρέπει να βεβαιωθούμε για την αξιοπιστία και αποτελεσματικότητα που παρέχει το σύστημα. Στις φάσεις αυτές συνήθως εντοπίζονται πιθανά λάθη τα οποία δε έγιναν αντιληπτά κατά τη διάρκεια υλοποίησής του. Στο κεφάλαιο αυτό παρατηρήθηκαν με την απαραίτητη προσοχή οι μέθοδοι ελέγχου και αξιολόγησης που χρησιμοποιήσαμε, καθώς και τα αποτελέσματά που λάβαμε.

6.2 Έλεγχος Συστήματος

Πλήθος δοκιμών απαιτούνται για την διασφάλιση ενός ολοκληρωμένου ελέγχου. Ο έλεγχος αυτός πραγματοποιήθηκε για να είμαστε βέβαιοι ότι δεν υπάρχουν απώλειες δεδομένων, ότι γίνεται με αποτελεσματικό τρόπο η χρήση του συστήματος, σε καμία περίπτωση δεν πρέπει να χάσουμε την εμπιστευτικότητα από τον χρήστη όπως και να διατηρήσουμε την ιδιωτικότητα των δεδομένων και πειραματικών δεδομένων που μας δόθηκαν για το testing.

Για την πραγματοποίηση των ελέγχων ορθότητας και ποιότητας της διεπαφής, εξετάστηκαν και ελέγχθηκαν διάφορα τυπικά σενάρια χρήσης για την ορθή λειτουργικότητα του συστήματος και την εύρεση πιθανών σφαλμάτων. Όσο περισσότερα σενάρια δημιουργήσουμε, τόσο το καλύτερα αφού η πιθανότητα σφάλματος που έχει το σύστημα μας μειώνετε. Παρατηρήσαμε ότι λειτουργεί κανονικά, χωρίς ανωμαλίες ή απώλειες δεδομένων.

Μετά των έλεγχου του συστήματος, πρέπει να ακολουθήσει η αξιολόγηση με βάση τα αποτελέσματα που παρατηρήσαμε κατά τη διαδικασία του ελέγχου(testing).

6.3 Αξιολόγηση - Αποτελέσματα

Για την αξιολόγηση του συστήματος, μετά την ολοκλήρωση των κατάλληλων ελέγχων, θα κάναμε διάφορες ερωτήσεις στη χρήστη σχετικά με το πόσο ικανοποιημένος ήταν με αυτό, πόσο εύκολα και γρήγορα μπορεί να μάθει το νέο σύστημα, όπως επίσης και πόσο ευχαριστημένος έχει μείνει.

.

Κεφάλαιο 7: Συμπεράσματα – Μελλοντική Εργασία

7.1 Συμπεράσματα	71
7.2 Μελλοντική Εργασία	72

7.1 Συμπεράσματα

Συμπερασματικά, μετά από αρκετή μελέτη των αποτελεσμάτων και του ελέγχου, έγινε δυνατή η εξαγωγή των ακόλουθων συμπερασμάτων σχετικά με το σύστημα BioCLC.

7.1.1 Ικανοποίηση Απαιτήσεων Χρήστη

Με εξαίρεση την μη ολοκλήρωση της λειτουργίας προσθαφαίρεσης κόμβων, το μεγαλύτερο μέρος των απαιτήσεων του χρήστη υλοποιήθηκαν επακριβώς.

Από την άλλη πλευρά, η επιλογή των εργαλείων που χρησιμοποιήθηκαν αποδείχθηκε κατάλληλη όσον αφορά την απεικόνιση γιατί ήταν ξεκάθαρη και έκανε τη διεπαφή φιλική προς το χρήστη.

7.1.2 Ευχρηστία Συστήματος

Το σύστημα που παράχθηκε, έχει σχετικά γρήγορη και υψηλή απόδοση, είναι εύκολο και φιλικό στη χρήση, ενώ έτσι όπως έχει σχεδιαστεί διευκολύνεται η εκμάθηση και η κατανόηση του περιεχομένου του.

7.1.3 Δυνατότητες Επέκτασης Συστήματος

Εξ' αρχής και από την προηγούμενη υλοποίηση, BioCT, στην οποία βασίστηκε η τρέχουσα, BioCLC, το σύστημα αξιοποιεί πλήρως τα χαρακτηριστικά της αντικειμενοστρεφούς ανάπτυξης. Είναι υλοποιημένο με τέτοιο τρόπο, ώστε να μπορεί να δεχτεί τυχόν μετατροπές, βελτιώσεις, επεκτάσεις, καθώς και οποιεσδήποτε νέα απαίτηση, συναφή με τις προηγούμενες, μπορεί να προκύψει στα πλαίσια μελλοντικής εργασίας. Αυτό το έχουμε

πετύχει χρησιμοποιώντας βιβλιοθήκες ελεύθερου λογισμικού, το οποίο θεωρείτε αναπόσπαστο μέρος ενός ερευνητικού πεδίου όπως η Βιοπληροφορική.

7.2 Μελλοντική Εργασία

Το παρόν σύστημα προσφέρει ένα πολύ ευρύ πεδίο για περαιτέρω έρευνα και ανάπτυξη, ενώ σίγουρα υπάρχουν αρκετές μελλοντικές επεκτάσεις που θα βοηθούσαν στην ικανοποίηση περισσότερων αναγκών και απαιτήσεων.

7.2.1 Ολοκλήρωση των Λειτουργιών που δεν Υλοποιήθηκαν

Στην παρούσα υλοποίηση, παραμένει σε εκκρεμότητα η ολοκλήρωση της προσθαφαίρεσης των κόμβων/ακμών στο δίκτυο. Αυτή θα μπορούσε με βάση τις οθόνες που υλοποιήθηκαν να ακολουθήσει την ακόλουθη μεθοδολογία.

- Ο χρήστης εισάγει τα δεδομένα που απαιτούνται και καθορίζουν ποιος κόμβος ή ακμή θα εισαχθεί και σε ποιο σημείο
- Το πρόγραμμα διαβάζει τα δεδομένα και δημιουργεί ένα νέο dot αρχείο
- Το νέο dot αρχείο μετατρέπεται μέσω του GraphViz σε εικόνα (png και svg)
- Η νέα εικόνα προβάλλεται στο χρήστη

Επιπλέον, εκκρεμεί η ολοκλήρωση των λειτουργιών του ZGRViewer, όπως αναφέρεται στη συνέχεια.

7.2.2 Πλήρης Ενσωμάτωση του ZGRViewer – Ενοποίηση Όλων των Λειτουργιών

Ο ZGRViewer, με την custom υλοποίησή του BioViewer, παρέχει ένα μεγάλο εύρος δυνατοτήτων στην εφαρμογή που ουσιαστικά μπορούν να αφαιρέσουν την στατικότητα που την χαρακτηρίζει.

Στην παρούσα διεπαφή συνυπάρχουν επί της ουσίας τμήματα της προηγούμενης, με ορισμένες αδυναμίες, καθώς και μερική υλοποίηση των λειτουργιών του ZGRViewer.

Προτείνεται η ενοποίηση όλων των λειτουργιών κάτω από το πλαίσιο του BioViewer, με τέτοιο τρόπο ώστε να μην χρειάζεται η δημιουργία και προβολή στατικών εικόνων png, αλλά να διασφαλίζεται η απευθείας εισαγωγή και προβολή των δεδομένων και των αποτελεσμάτων τους στον BioViewer.

7.2.3 Εισαγωγή των Στοιχείων του Απαγωγικού Λογικού Προγραμματισμού

Στο παρόν στάδιο, ουσιαστικά το σύστημα συνεχίζει να μην είναι λειτουργικό, όπως και στην υλοποίηση του BioCT. Για να γίνει λειτουργικό και να μπορούν να παραχθούν σωστά οι ανάλογες υποθέσεις, θα πρέπει να εισαχθούν τρόποι με τους οποίους να μπορούν να γίνουν οι υπολογισμοί αυτοί. Υπάρχουν διάφορες εναλλακτικές για την επίτευξη του στόχου αυτού. Αφενός, η σύνδεση της εφαρμογής με μια γλώσσα λογικού προγραμματισμού, και η εισαγωγή στοιχείων απαγωγικού λογικού προγραμματισμού θα μπορούσε να αποτελέσει λύση. Αφετέρου, θα μπορούσε να πραγματοποιηθεί μια πλήρης προσομοίωση με χρήση των υπάρχοντων εργαλείων της Java και πιθανή αναζήτηση επιπλέον βιβλιοθηκών που να καλύπτουν πιο ιδιαίτερες ανάγκες, εάν αυτό απαιτηθεί.

Βιβλιογραφία

- [1] Antonis C. Kakas, Bert Van Nuffelen, Marc Denecker: A-System: Problem Solving through Abduction. IJCAI 2001, 591-596
- [2] Cell Signaling Technology , <http://www.cellsignal.com/>
- [3] Ανάπτυξη Επιστημονικών Θεωριών Βιολογίας στο Πλαίσιο της Υπολογιστικής Λογικής: αναπαράσταση της ενδοκυτταρικής διαδικασίας της απόπτωσης με χρήση Απαγωγικού Λογικού Μοντέλου – Σωτηράκης Λαζάρου Γεωργίου, Διατριβή Master, Πανεπιστήμιο Κύπρου, Τμήμα Πληροφορικής, 2012
- [4] Μετατροπή, Ενοποίηση και Οπτικοποίηση Μεταπληροφορίας για Γονιδιακά Ρυθμιστικά Δίκτυα - Ιωάννη Μαράκη, Διπλωματική Εργασία, Πολυτεχνείο Κρήτης, Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών
- [5] Lawrence Hunter: Molecular Biology for Computer Scientists – Chapter 1
- [6] S. Lazarou, A. C. Kakas, and C. Neophytou, A. Contantinou: Logical Modeling of Cancer and Chemoprevention
- [7] Εφαρμογές Ελεύθερου Λογισμικού στη Βιοπληροφορική – Ελευθέριος Πιλάλης, Ομάδα Μεταβολικής Μηχανικής και Βιοπληροφορικής Ινστιτούτο Βιολογικών Ερευνών και Βιοτεχνολογία, Εθνικό Ίδρυμα Ερευνών
- [8] Γραφική Διεπαφή για Αυτόματο Επιστημονικό Βοηθό Μοριακής Βιολογίας – Ειρήνη Ανδρεού, Ατομική Δοπλβματική Εργασία, Πανεπιστήμιο Κύπρου, Τμήμα Πληροφορικής, 2014
- [9] KEGG: Kyoto Encyclopedia of Genes and Genomes, www.genome.jp/kegg/
- [10] ABDUCTIVE LOGIC PROGRAMMING, <http://www.math-info.univ-paris5.fr/~moraitis/webpapers/ALP-Intro.pdf>
- [11] «Κυτταρικός Κύκλος, Βιότοπος, Ιστότοπος από βιολόγους – εκπαιδευτικούς, Θεματική ενότητα : Το Κύτταρο» (<http://www.edubiosite.gr/index.php/thematikes-enotites/to-kyttaro/o-kyttarikos-kyklos>)
- [12] Βασιλική Μακροδήμητρα, «Δημιουργία Διαδραστικής Βάσης Δεδομένων για Ινδολοκαρβοζολικά και Φλαβονοειδικά Παράγωγα. Σχέσεις Δομής-Δράσης.», Διατριβή, Πανεπιστήμιο Πατρών, 2008
- [13] Μ. Ανδρίκουλα, Γ. Βαρθολομάτος, «Απόπτωση. Ο κυτταρικός θάνατος προϋπόθεση για τη ζωή.», Αρχεία Ελληνικής Ιατρικής, 18(5):496-513, 2001
- [14] Γ. Δ. Κλούβας, «Μονοπάτια Καρκινογένεσης (Πολλαπλασιασμός, Κυτταρικός Κύκλος, Αναδιπλασιασμός του DNA, Αθανατοποίηση, Αντι-Απόπτωση,

Αγγειογένεση, Διήθηση-Μετάσταση)», Ελληνική Χειρουργική Εταιρεία (<http://www.exe1928.gr/>)

- [15] Ευτυχία Σ. Νανά, «Η χρήση της Βιοπληροφορικής στη μελέτη της νόσου του Καρκίνου», Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών, 2009
- [16] C. Bachmaier, U. Brandes, F. Schreiber, « Handbook of Graph Drawing and Visualization», p. 621-646, CRC Press, 2014
- [17] A. Ken, J. Gosling, D. Holmes. «The Java programming language», Vol. 2. Reading: Addison-wesley, 1996
- [18] Bill Venners, «Inside the Java virtual machine», McGraw-Hill Inc., 1996
- [19] H. Schildt, «Java 2: the complete reference», McGraw-Hill Professional, 2000.
- [20] P. Keegan et al, « NetBeans (TM) IDE Field Guide: Developing Desktop, Web, Enterprise, and Mobile Applications », Prentice Hall PTR, 2006
- [21] Graphviz Java API, Created by Laszlo Szathmary, alias Jabba Laci, 2003—2011, (<http://www.loria.fr/~szathmar/off/projects/java/GraphVizAPI/index.php>)
- [22] GraphViz (<http://www.graphviz.org/>)
- [23] GraphStream (<http://graphstream-project.org>)
- [24] ZGRViewer (<http://zvtn.sourceforge.net/zgrviewer.html>)
- [25] ZVTM (<http://zvtn.sourceforge.net/>)
- [26] E. Κεχρής, «Κύκλος Ζωής Ανάπτυξης Λογισμικού» (http://www.teicm.gr/repository/repository/sygg_lyiko/didaktiko%20yliko%20gia%20pylh/%CE%91%CE%A1%CE%A7%CE%95%CE%99%CE%9F%20%CE%94%CE%99%CE%94%CE%91%CE%9A%CE%A4%CE%99%CE%9A%CE%9F%CE%A5%20%CE%A5%CE%9B%CE%99%CE%9A%CE%9F%CE%A5%20%CE%A4%CE%9C%CE%97%CE%9C%CE%91%CE%A4%CE%9F%CE%A3%20%CE%94.%CE%95/%CE%95.%20%CE%9A%CE%95%CE%A7%CE%A1%CE%97%CE%A3/%CE%A0%CE%9B%CE%97%CE%A1%CE%9F%CE%A6%CE%9F%CE%A1%CE%99%CE%91%CE%9A%CE%91%20%CE%A3%CE%A5%CE%A3%CE%A4%CE%97%CE%9C%CE%91%CE%A4%CE%91%20%CE%9C%CE%92%CE%91/SDL C.pdf)
- [27] Β. Βεσκούκης, «Σημειώσεις για το μάθημα Τεχνολογία Λογισμικού», Πανεπιστήμιο Πειραιώς, 2000

- [28] OMG Unified Language Specification, version 1.3, June 1999 (<http://www.omg.org/>)
- [29] Γ. Ρήγας, Α. Αθανασίου: Ο κυτταρικός κύκλος, Κεφάλαιο 9
- [30] Εισαγωγή στη Γλώσσα Προγραμματισμού Java, Ενότητα 1 – Βασικά Στοιχεία της Γλώσσας, Εργαστήριο Τεχνολογίας Πολυμέσων ΕΜΠ, (http://aetos.it.teithe.gr/~sfetsos/java_book_EMP.pdf)
- [31] Σημειώσεις Μαθήματος «ΕΠΛ361 – Τεχνολογία Λογισμικού», Ενότητες 03-04, Γ. Καπιτσάκη, Τμήμα Πληροφορικής Παν. Κύπρου, 2012-2013
- [32] Σημειώσεις Μαθήματος «ΕΠΛ361 – Τεχνολογία Λογισμικού», Ενότητες 12-16, Γ. Καπιτσάκη, Τμήμα Πληροφορικής Παν. Κύπρου, 2012-2013

Παράρτημα Α

Ερωτηματολόγιο

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΑΝΑΝΕΩΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ApoCelSys

Υπεύθυνος καθηγητής: Αντώνης Κάκας

Φοιτήτριες Διπλωματικής εργασίας: Ειρήνη Ανδρέου, Έμιλη Ταλιαδώρου

1. Θα θέλατε όταν μπαίνετε στο σύστημα να σας ζητείτε κωδικός πρόσβασης, ούτως ώστε να μην μπορούν άλλοι να έχουν πρόσβαση στα πειράματά σας ;

α) ΝΑΙ

β) ΟΧΙ

2. Το σύστημα στην παρούσα φάση έχει ένα συγκεκριμένο τρόπο εισαγωγής των πειραματικών δεδομένων (δεξί ή διπλό κλικ -> επιλογή active observed, inactive observed κλπ). Είσαστε ικανοποιημένη από αυτή τη λειτουργία ; α) ΝΑΙ

β) ΟΧΙ

2.1. Αν όχι, τι πώς θα θέλατε να γίνεται εναλλακτικά αυτή η εισαγωγή ;

2.2. Θα θέλατε να υπάρχει η δυνατότητα να σημαδεύετε εσείς από την αρχή κάποιες ουσίες/πρωτεΐνες/διεργασίες που πιστεύετε ότι θα επηρεαστούν από το φάρμακο, ούτως ώστε να μπορείτε να κάνετε τις ανάλογες συγκρίσεις ;

α) ΝΑΙ

β) ΟΧΙ

3. Στο ανανεωμένο σύστημα θα προστεθεί η λειτουργία του cell cycle. Γι' αυτό το σημείο σκεφτήκαμε να υπάρχει επιλογή στην αρχή του συστήματος για το ποια λειτουργία από τις δύο θέλετε και στη συνέχεια, αν θέλετε να μεταβείτε στο cell cycle να υπάρχει μια συγκεκριμένη επιλογή στο menu επιλογών που θα ανοίγει ένα παράθυρο παράλληλα με την απόπτωση στο οποίο θα φαίνονται τα pathways του cell cycle.

Επίσης, αν πατήσετε δεξί κλικ σε μια ουσία/πρωτεΐνη/διεργασία θα υπάρχει η επιλογή "Go to cell cycle" και αν η συγκεκριμένη ουσία/πρωτεΐνη/διεργασία συμμετέχει και στο cell cycle, να ανοίγει το παράλληλο παράθυρο του cell cycle με έμφαση στη συγκεκριμένη ουσία/πρωτεΐνη/διεργασία.

Συμφωνείτε, διαφωνείτε ή θέλετε να προσθέσετε κάτι σχετικά με αυτή μας την απόφαση ; (π.χ. το πώς θέλετε να γίνεται η έμφαση στη συγκεκριμένη ουσία/πρωτεΐνη/διεργασία κλπ)

4. Πώς θα θέλετε να εμφανίζονται τα σχήματα των διαφορετικών κόμβων; Σχεδιάστε (πρόχειρα) πιο κάτω :

α) πρωτεΐνες :

β) διεργασίες :

γ) άλλες ουσίες :

5. Πώς θα θέλατε να εμφανίζεται το propagation ;

α) Όπως είναι στο υπάρχον σύστημα

β) Με διαφορετικό τρόπο. Προσθέστε πώς :

6. Θα θέλατε να υπάρχει η επιλογή «find», με την οποία θα γράφετε το όνομα της ουσίας/πρωτεΐνης/διεργασίας που ψάχνετε και θα σας παραπέμπει με κάποιο τρόπο σε αυτή ;

α) ΝΑΙ

β) ΟΧΙ

7. Ένα ζήτημα που μας απασχολεί σχετικά με το σύστημα είναι το zoom. Πώς θα θέλατε να γίνεται ;

α) Να υπάρχουν σημεία ένδειξης : (+) για zoom-in και (-) για zoom-out ;

β) Να γίνεται επιλογή από εσάς σε συγκεκριμένες ουσίες/διεργασίες και να γίνεται εστίαση μόνο στα μονοπάτια που αφορούν άμεσα τη συγκεκριμένη ουσία/διεργασία που επιλέξατε (τους κοντινούς γείτονες μόνο) ;

7.1. Αν επιλέξατε β, πώς θα θέλατε να εμφανίζονται τα υπόλοιπα σημεία εκτός της μεγεθυμένης περιοχής ;

α) Να μην φαίνονται καθόλου

β) Να φαίνονται μόνο συγκεκριμένες πρωτεΐνες/διεργασίες (σημειώστε ποιες :

7.2. Θα θέλατε να υπάρχει η επιλογή του zoom και κατά την εισαγωγή των δεδομένων και κατά την εμφάνιση των αποτελεσμάτων ;

α) ΝΑΙ

β) ΟΧΙ

7.3. Προσθέστε κάποια δικά σας σχόλια ως προς το πώς θέλετε να γίνεται το zoom:

8. Μας είπατε ότι μελετάτε την πρωτεΐνη survivin.

α) Θα θέλατε να υπάρχει κάποια επιλογή που να δείχνει μόνο το συγκεκριμένο pathway όποτε θέλετε εσείς ;

β) Θα προτιμούσατε να υπάρχει μια επιλογή όπου να επιλέγετε ανά πάσα στιγμή ποια πρωτεΐνη/διεργασία σας ενδιαφέρει, και το σύστημα να εντοπίζει και να εμφανίζει αυτόματα το pathway που σχετίζεται με αυτή ;

γ) Κανένα από τα πιο πάνω (δε θέλω διαφοροποίηση ως προς τα pathways)

δ) Και το α και το β

Αν επιλέξατε α, β ή δ , πώς θα θέλατε να εμφανίζει το σύστημα τα συγκεκριμένα pathways

(π.χ. με διαφορετικό χρώμα κλπ.) ;

9. Στο ανανεωμένο σύστημα θα προστεθεί η λειτουργία για πρόσθεση νέων κόμβων και συνδέσεων (νέες υποθέσεις) . Θα υπάρχει μια επιλογή «add» η οποία όταν επιλεχθεί θα εμφανίζεται ένα παράθυρο στο οποίο θα γράφετε το όνομα του νέου κόμβου, θα επιλέγετε αν είναι πρωτεΐνη, διεργασία ή άλλη ουσία και θα προσθέτετε τις ανάλογες συνδέσεις με άλλες ουσίες καθώς και τον τύπο των συνδέσεων. Επίσης θα υπάρχει η επιλογή για πρόσθεση νέας σύνδεσης ανάμεσα σε υπάρχοντες κόμβους. Συμφωνείτε, διαφωνείτε ή θέλετε να προσθέσετε κάτι σχετικά με αυτή μας την απόφαση ;

Παράρτημα Β

Οδηγίες Εγκατάστασης Συστήματος (README.txt)

Οδηγίες Α : Εγκατάσταση για Χρήστες

1. Εγκατάσταση της Java
2. Εγκατάσταση του εργαλείου Graphviz, version 2.30 για Windows:
<http://www.graphviz.org/pub/graphviz/stable/windows/graphviz-2.30.1.msi>
3. Αποσυμπίεση αρχείου BioCLC_bin.zip
4. Εκκίνηση του συστήματος (αρχείο BioCLC.jar)
(password : **bioclc**)

Οδηγίες Β : Εγκατάσταση για Προγραμματιστές

1. Εγκατάσταση της Java
2. Εγκατάσταση του εργαλείου Graphviz, version 2.30 για Windows:
<http://www.graphviz.org/pub/graphviz/stable/windows/graphviz-2.30.1.msi>
3. Εγκατάσταση του λογισμικού NetBeans IDE 8.0 για Windows
<https://netbeans.org/downloads/start.html?platform=windows&lang=en&option=javaee>
4. Αποσυμπίεση αρχείου BioCLC_src.zip
5. Άνοιγμα του λογισμικού NetBeans->File->Import Project και επιλογή του αρχείου BioCT.zip

Παράρτημα Γ

Graphviz Java API

```
package GraphViz;

// GraphViz.java - a simple API to call dot from Java programs

/*$Id$*/
/*
*****
*
*          (c) Copyright 2003 Laszlo Szathmary
*
* This program is free software; you can redistribute it and/or modify it
* under the terms of the GNU Lesser General Public License as published by
* the Free Software Foundation; either version 2.1 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful, but
* WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY
* or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public
* License for more details.
*
* You should have received a copy of the GNU Lesser General Public License
* along with this program; if not, write to the Free Software Foundation,
* Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
*
*****
*/

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileWriter;
import java.io.InputStreamReader;
import java.lang.System;

/**
 * <dl>
 * <dt>Purpose: GraphViz Java API
 * <dd>
 * </dd>
 * <dt>Description:
 * <dd> With this Java class you can simply call dot
 *      from your Java programs
 * </dd>
 * <dt>Example usage:
 *   <dd>
 *     <pre>
 *       GraphViz gv = new GraphViz();

```

```

*     gv.addln(gv.start_graph());
*     gv.addln("A -> B;");
*     gv.addln("A -> C;");
*     gv.addln(gv.end_graph());
*     System.out.println(gv.getDotSource());
*
*     String type = "gif";
*     File out = new File("out." + type);    // out.gif in this example
*     gv.writeGraphToFile( gv.getGraph( gv.getDotSource(), type ), out );
* </pre>
* </dd>
*
* </dl>
*
* @version v0.4, 2011/02/05 (February) -- Patch of Keheliya Gallaba is added. Now you
* can specify the type of the output file: gif, dot, fig, pdf, ps, svg, png, etc.
* @version v0.3, 2010/11/29 (November) -- Windows support + ability
* to read the graph from a text file
* @version v0.2, 2010/07/22 (July) -- bug fix
* @version v0.1, 2003/12/04 (December) -- first release
* @author Laszlo Szathmary (<a href="mailto:jabba.laci@gmail.com">jabba.laci@gmail.com</a>)
*/

public class GraphViz
{
    /**
     * The dir. where temporary files will be created.
     */
    //private static String TEMP_DIR = "/tmp"; // Linux
    //private static String TEMP_DIR = "C:\\Users\\user\\Documents\\Temp1";    //
Windows
    //private          static          String          TEMP_DIR          =
"C:\\Users\\Lisa\\Documents\\NetBeansProjects\\BioCT\\Temp";
    private static String workDir = System.getProperty("user.dir");
    private static String TEMP_DIR = workDir+"\\Temp";
    /**
     * Where is your dot program located? It will be called externally.
     */
    private static String DOT = getDotPath();
    /**
     * The source of the graph written in dot language.
     */
    private StringBuilder graph = new StringBuilder();

    /**
     * Constructor: creates a new GraphViz object that will contain
     * a graph.
     */
    public GraphViz() {
    }

    public static String getDotPath(){
        String DOT1 = "C:\\Program Files (x86)\\Graphviz2.30\\bin\\dot.exe";    //
Windows
        String DOT2 = "C:\\Program Files\\Graphviz2.30\\bin\\dot.exe";
        String DOT3 = "/usr/bin/dot";    // Linux
        String dotPath = "";

        File f = new File(DOT1);
        if(f.exists() && !f.isDirectory())
        {
            dotPath = DOT1;
        }else{

```

```

        f = new File(DOT2);

        if(f.exists() && !f.isDirectory())
        {
            dotPath = DOT2;
        }else{
            f = new File(DOT3);

            if(f.exists() && !f.isDirectory())
            {
                dotPath = DOT3;
            }else{
                dotPath = "";
            }
        }
    }
    //System.out.println(dotPath);
    return dotPath;
}
/**
 * Returns the graph's source description in dot language.
 * @return Source of the graph in dot language.
 */
public String getDotSource() {
    return graph.toString();
}

/**
 * Adds a string to the graph's source (without newline).
 */
public void add(String line) {
    graph.append(line);
}

/**
 * Adds a string to the graph's source (with newline).
 */
public void addln(String line) {
    graph.append(line + "\n");
}

/**
 * Adds a newline to the graph's source.
 */
public void addln() {
    graph.append('\n');
}

/**
 * Returns the graph as an image in binary format.
 * @param dot_source Source of the graph to be drawn.
 * @param type Type of the output image to be produced, e.g.: gif, dot, fig, pdf,
ps, svg, png.
 * @return A byte array containing the image of the graph.
 */
public byte[] getGraph(String dot_source, String type)
{
    File dot;
    byte[] img_stream = null;

    try {
        dot = writeDotSourceToFile(dot_source);
        if (dot != null)

```

```

        {
            img_stream = get_img_stream(dot, type);
            if (dot.delete() == false)
                System.err.println("Warning: " + dot.getAbsolutePath() + " could not be
deleted!");
            return img_stream;
        }
        return null;
    } catch (java.io.IOException ioe) { return null; }
}

/**
 * Writes the graph's image in a file.
 * @param img    A byte array containing the image of the graph.
 * @param file    Name of the file to where we want to write.
 * @return Success: 1, Failure: -1
 */
public int writeGraphToFile(byte[] img, String file)
{
    File to = new File(file);
    return writeGraphToFile(img, to);
}

/**
 * Writes the graph's image in a file.
 * @param img    A byte array containing the image of the graph.
 * @param to      A File object to where we want to write.
 * @return Success: 1, Failure: -1
 */
public int writeGraphToFile(byte[] img, File to)
{
    try {
        FileOutputStream fos = new FileOutputStream(to);
        fos.write(img);
        fos.close();
    } catch (java.io.IOException ioe) { return -1; }
    return 1;
}

/**
 * It will call the external dot program, and return the image in
 * binary format.
 * @param dot Source of the graph (in dot language).
 * @param type Type of the output image to be produced, e.g.: gif, dot, fig, pdf,
ps, svg, png.
 * @return The image of the graph in .gif format.
 */
private byte[] get_img_stream(File dot, String type)
{
    File img;
    byte[] img_stream = null;

    try {
        img = File.createTempFile("graph_", "."+type, new File(GraphViz.TEMP_DIR));
        Runtime rt = Runtime.getRuntime();

        // patch by Mike Chenault
        String[] args = {DOT, "-T"+type, dot.getAbsolutePath(), "-o",
img.getAbsolutePath()};
        Process p = rt.exec(args);

        p.waitFor();
    }

```

```

        FileInputStream in = new FileInputStream(img.getAbsolutePath());
        img_stream = new byte[in.available()];
        in.read(img_stream);
        // Close it if we need to
        if( in != null ) in.close();

        if (img.delete() == false)
            System.err.println("Warning: " + img.getAbsolutePath() + " could not be
deleted!");
    }
    catch (java.io.IOException ioe) {
        System.err.println("Error:          in I/O processing of tempfile in dir " +
GraphViz.TEMP_DIR+"\n");
        System.err.println("          or in calling external command");
        ioe.printStackTrace();
    }
    catch (java.lang.InterruptedException ie) {
        System.err.println("Error:  the execution of the external program was
interrupted");
        ie.printStackTrace();
    }
    }

    return img_stream;
}

/**
 * Writes the source of the graph in a file, and returns the written file
 * as a File object.
 * @param str Source of the graph (in dot language).
 * @return The file (as a File object) that contains the source of the graph.
 */
private File writeDotSourceToFile(String str) throws java.io.IOException
{
    File temp;
    try {
        temp = File.createTempFile("graph_", ".dot.tmp", new
File(GraphViz.TEMP_DIR));
        FileWriter fout = new FileWriter(temp);
        fout.write(str);
        fout.close();
    }
    catch (Exception e) {
        System.err.println("Error: I/O error while writing the dot source to temp
file!");
        return null;
    }
    return temp;
}

/**
 * Returns a string that is used to start a graph.
 * @return A string to open a graph.
 */
public String start_graph() {
    return "digraph G {";
}

/**
 * Returns a string that is used to end a graph.
 * @return A string to close a graph.
 */
public String end_graph() {
    return "}";
}

```

```

    }

    /**
     * Read a DOT graph from a text file.
     *
     * @param input Input text file containing the DOT graph
     * source.
     */
    public void readSource(String input)
    {
        StringBuilder sb = new StringBuilder();

        try
        {
            FileInputStream fis = new FileInputStream(input);
            DataInputStream dis = new DataInputStream(fis);
            BufferedReader br = new BufferedReader(new InputStreamReader(dis));
            String line;
            while ((line = br.readLine()) != null) {
                sb.append(line);
            }
            dis.close();
        }
        catch (Exception e) {
            System.err.println("Error: " + e.getMessage());
        }

        this.graph = sb;
    }
} // end of class GraphViz

```