

Ατομική Διπλωματική Εργασία

**ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ ΓΙΑ ΑΥΤΟΜΑΤΟ ΕΠΙΣΤΗΜΟΝΙΚΟ ΒΟΗΘΟ
ΜΟΡΙΑΚΗΣ ΒΙΟΛΟΓΙΑΣ**

Ειρήνη Ανδρέου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αύγουστος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ ΓΙΑ ΑΥΤΟΜΑΤΟ ΕΠΙΣΤΗΜΟΝΙΚΟ ΒΟΗΘΟ ΜΟΡΙΑΚΗΣ ΒΙΟΛΟΓΙΑΣ

Ειρήνη Ανδρέου

Επιβλέπων Καθηγητής

Αντώνης Κάκας

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

Αύγουστος 2014

Ευχαριστίες

Με την παρούσα διπλωματική εργασία, οι προπτυχιακές σπουδές μου στο Πανεπιστήμιο Κύπρου που αποτέλεσαν ένα σημαντικό κεφάλαιο της ζωής μου, λαμβάνουν τέλος. Στο σημείο αυτό θεωρώ υποχρέωσή μου να εκφράσω τις ευχαριστίες μου σε όλους όσους συνέβαλαν στο να επιτύχω την ολοκλήρωσή της και κατ' επέκταση στην απόκτηση του Πτυχίου μου.

Πρώτα απ' όλα, θα ήθελα να ευχαριστήσω τον επιβλέπον καθηγητή μου κο. Αντώνη Κάκα για τη συνεργασία που είχαμε, που μου εμπιστεύτηκε την παρούσα διπλωματική και που με στήριζε και με βοηθούσε όποτε τον χρειαζόμουν.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω την υποψήφια διδάκτωρ του τμήματος Βιολογίας του Πανεπιστημίου Κύπρου, κα. Χριστιάννα Νεοφύτου, για το χρόνο που, ως βασική χρήστης, μου αφιέρωσε καθ' όλη τη διάρκεια της ανάπτυξης του συστήματος, καθώς και για τη βοήθειά της σχετικά με την κατανόηση του βιολογικού υπόβαθρου της παρούσας διπλωματικής εργασίας.

Επίσης, θεωρώ πως οφείλω να ευχαριστήσω όλα τα άτομα που στάθηκαν δίπλα μου αυτά τα τέσσερα χρόνια και που με τη στήριξή τους με βοήθησαν να αντεπεξέλθω σε όλες τις δυσκολίες που αντιμετώπισα. Ευχαριστώ πολύ την οικογένειά μου, ιδιαίτερα τη μητέρα μου, και τους φίλους μου για την κατανόησή τους και την ηθική υποστήριξη που μου παρείχαν καθ' όλη τη διάρκεια των τελευταίων χρόνων και ιδιαίτερα κατά τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Ευχαριστώ,
Ειρήνη Ανδρέου.

Περίληψη

Η παρούσα διπλωματική εργασία, ασχολείται με ένα τομέα της Βιοπληροφορικής, τη Συστημική Βιολογία. Η επιστήμη αυτή, αφορά την απεικόνιση και επεξεργασία βιολογικών δεδομένων και κατ' επέκταση, συμβάλλει στη διάγνωση και θεραπεία διαφόρων ασθενειών, όπως του καρκίνου.

Πιο συγκεκριμένα, το θέμα που θα μας απασχολήσει είναι η δημιουργία του γραφικού περιβάλλοντος ενός αυτόματου επιστημονικού βοηθού μοριακής βιολογίας για μελέτη διεργασιών καρκινικών κυττάρων από Βιολόγους. Η γραφική αυτή διεπαφή, απεικονίζει τα βιολογικά δίκτυα (μονοπάτια) που σχετίζονται με τις δύο βασικές κυτταρικές διεργασίες, τη διεργασία της απόπτωσης και τη διεργασία του πολλαπλασιασμού του κυττάρου. Επίσης, παίρνει ως είσοδο κάποια πειραματικά δεδομένα από το χρήστη και δείχνει πώς θα διαμορφωθεί η απεικόνιση του δικτύου, μετά από διάφορους υπολογισμούς που θα γίνονταν στον επιστημονικό βοηθό, σχετικά με υποθέσεις που γίνονται για την κατάσταση των διαφόρων γονιδίων μετά από το πείραμα, καθώς και μετά τη διάδοση των διαφόρων σημάτων ανάμεσά τους.

Για τη δημιουργία της γραφικής αυτής διεπαφής, χρειάζονταν να γίνουν διάφορες έρευνες, οι οποίες θα περιγραφούν. Οι έρευνες αυτές αφορούν το βιολογικό υπόβαθρο του θέματος, καθώς και ήδη υπάρχοντα συστήματα που έχουν παρόμοιους σκοπούς. Στη συνέχεια, θα αναφερθούν και θα αναλυθούν οι απαιτήσεις του συστήματος, τις οποίες πήραμε μέσω ερωτηματολογίου και συναντήσεων με τη βασική χρήστη του συστήματος. Έπειτα, γίνεται μια περιγραφή των διαφόρων τεχνολογιών και εργαλείων που δοκιμάστηκαν μέχρι να βρεθούν τα κατάλληλα για να ικανοποιούνται οι απαιτήσεις του χρήστη και γίνεται μια περιγραφή και ανάλυση του σχεδιασμού και της υλοποίησης του συστήματος. Τέλος, περιγράφεται η διαδικασία που ακολουθήθηκε για τον έλεγχο του συστήματος, τα αποτελέσματα που πήραμε, μερικά συμπεράσματα, καθώς και προτάσεις για μελλοντική εργασία.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Γενική Εισαγωγή	1
	1.2 Στόχος Διπλωματικής Εργασίας	2
	1.3 Δομή Διπλωματικής Εργασίας	3
Κεφάλαιο 2	Περιγραφή Προβλήματος.....	4
	2.1 Εισαγωγή	4
	2.2 Συνοπτική Περιγραφή Προβλήματος	4
	2.3 Βιολογική Υπόσταση Προβλήματος	5
	2.4 Επεξήγηση Απεικονισμένου Κυτταρικού Δικτύου	7
	2.5 Υπάρχοντα Συστήματα	9
	2.6 Προσέγγιση Προβλήματος	14
Κεφάλαιο 3	Ανάλυση Απαιτήσεων-Προδιαγραφών.....	15
	3.1 Εισαγωγή	15
	3.2 Ανάλυση Απαιτήσεων - Προδιαγραφών	15
Κεφάλαιο 4	Απαιτούμενη Γνώση και Τεχνολογίες.....	19
	4.1 Εισαγωγή	19
	4.2 Σύνολο Τεχνολογιών 1 : GUI σε MATLAB, NetworkX, Python	20
	4.3 Σύνολο Τεχνολογιών 2 : GUI σε wxPython, NetworkX	23
	4.4 Σύνολο Τεχνολογιών 3 : GUI σε Tkinter(Python),Graphviz σε Λ.Σ Linux	24
	4.5 Σύνολο Τεχνολογιών 4 : GUI σε Java, Graphviz	25
Κεφάλαιο 5	Το Σύστημα BioCT (Biological Cell-Processing Tool).....	28
	5.1 Εισαγωγή	28
	5.2 Μοντέλο Κύκλου Ζωής Λογισμικού	28
	5.3 Σχεδιασμός Συστήματος	33
	5.4 Υλοποίηση Συστήματος	35

Κεφάλαιο 6 Έλεγχος – Αξιολόγηση Συστήματος	48
6.1 Εισαγωγή	48
6.2 Έλεγχος	48
6.2 Αξιολόγηση - Αποτελέσματα	49
 Κεφάλαιο 7 Συμπεράσματα – Μελλοντική Εργασία.....	50
7.1 Συμπεράσματα	50
7.2 Μελλοντική Εργασία	51
 Βιβλιογραφία	53
 Παράρτημα Α.....	A-1
 Παράρτημα Β.....	B-6
 Παράρτημα Γ.....	Γ-12

Κατάλογος με Εικόνες

Εικόνα 1 : Παράδειγμα απεικόνισης τριών πρωτεϊνών ενός κυτταρικού δικτύου.....	7
Εικόνα 2 : Απεικόνιση Δικτύου Απόπτωσης από τη σελίδα της Cell Signaling.....	9
Εικόνα 3 : Παράδειγμα σύντομου μενού του συστήματος ApoCelSys.....	12
Εικόνα 4 : Παράδειγμα ολοκληρωμένου μενού του συστήματος ApoCelSys.....	12
Εικόνα 5 : Αναπαράσταση Μονοπατιών και Μενού Εντολών του συστήματος ApoCelSys.....	12
Εικόνα 6 : Στιγμιότυπο του συστήματος ApoCelSys μετά τη διάδοση του σήματος.....	13
Εικόνα 7 : Σχεδιάγραμμα Απλού Σειριακού Μοντέλου.....	29
Εικόνα 8 : Σχεδιάγραμμα Μοντέλου του Καταρράκτη.....	30
Εικόνα 9 : Σχεδιάγραμμα Αυξητικού Μοντέλου.....	30
Εικόνα 10 : Σχεδιάγραμμα Εξελικτικού Μοντέλου.....	31
Εικόνα 11 : Σχεδιάγραμμα Σπειροειδούς Μοντέλου.....	31
Εικόνα 12 : Σχεδιάγραμμα Μοντέλου Πρωτοτυποποίησης.....	32
Εικόνα 13 : UML Διάγραμμα Χρήσης (Use Case) του Συστήματος BioCT.....	34
Εικόνα 14 : Οθόνη Πρόσβασης στο Σύστημα BioCT.....	36
Εικόνα 15 : Μήνυμα λάθους εισαγωγής κωδικού πρόσβασης.....	36
Εικόνα 16 : Αρχική Οθόνη Συστήματος BioCT.....	37
Εικόνα 17 : Οθόνη Απόπτωσης – Αρχική Εικόνα.....	38

Εικόνα 18 : Οθόνη Πολλαπλασιασμού – Αρχική Εικόνα.....	39
Εικόνα 19 : Μενού επιλογών για εισαγωγή πειραματικών δεδομένων στην οθόνη απόπτωσης.....	39
Εικόνα 20 : Μενού επιλογών για εισαγωγή πειραματικών δεδομένων στην οθόνη πολλαπλασιασμού.....	39
Εικόνα 21 : Οθόνη πολλαπλασιασμού, στην οποία έγινε παραπομπή από την οθόνη της απόπτωσης, μέσω του γονιδίου(διεργασίας) Cell-Cycle.....	40
Εικόνα 22 : Μήνυμα που θα εμφανιστεί εάν στην οθόνη του πολλαπλασιασμού δεν υπάρχει το γονίδιο που επιλέξαμε για παραπομπή.....	40
Εικόνα 23 : Οθόνη Απόπτωσης μετά από εισαγωγή των πειραματικών δεδομένων.....	41
Εικόνα 24 : Οθόνη Πολλαπλασιασμού μετά από εισαγωγή των πειραματικών δεδομένων..	41
Εικόνα 25 : Οθόνη Απόπτωσης μετά από τον υπολογισμό της υπόθεσης.....	42
Εικόνα 26 : Οθόνη Πολλαπλασιασμού μετά από τον υπολογισμό της υπόθεσης.....	43
Εικόνα 27 : Οθόνη Απόπτωσης μετά από τη διάδοση του σήματος.....	43
Εικόνα 28 : Οθόνη Πολλαπλασιασμού μετά από τη διάδοση του σήματος.....	44
Εικόνα 29 : Παράθυρο για άνοιγμα υπάρχοντος πειράματος.....	44
Εικόνα 30 : Παράθυρο για αποθήκευση πειράματος.....	44
Εικόνα 31 : Παράθυρο για αναζήτηση γονιδίου.....	45
Εικόνα 32 : Μήνυμα όταν το γονίδιο που αναζητήθηκε δεν ανήκει στο μονοπάτι.....	45
Εικόνα 33 : Οθόνη Απόπτωσης όταν αναζητήθηκε σε αυτή το γονίδιο με το όνομα : «JNK».....	45

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Εισαγωγή	1
1.2 Στόχος Διπλωματικής Εργασίας	2
1.3 Δομή Διπλωματικής Εργασίας	3

1.1 Γενική Εισαγωγή

Η Βιοπληροφορική είναι μια επιστήμη ευρέως διαδεδομένη στις μέρες μας, που εξελίσσεται με πολύ γρήγορους ρυθμούς, ειδικά τα τελευταία χρόνια. Η επιστήμη αυτή, συνδυάζει τις επιστήμες των Μαθηματικών, της Χημείας, της Φυσικής, της Ιατρικής και φυσικά της Βιολογίας και της Πληροφορικής. Ως διεπιστημονικό πεδίο ερευνών, ασχολείται με τη συλλογή και οργάνωση βιολογικών δεδομένων. Μέσω της επιστήμης της Πληροφορικής, αναπτύσσονται και χρησιμοποιούνται υπολογιστικά εργαλεία, με σκοπό την απεικόνιση, ανάλυση και ερμηνεία βιολογικών δεδομένων. Μ' αυτό τον τρόπο, δύσκολα προβλήματα που αφορούν τη Βιολογία, πλέον μπορούν να επιλυθούν. Ένα από αυτά τα προβλήματα είναι και η απεικόνιση και επεξεργασία κυτταρικών δικτύων (μονοπατιών), το οποίο θα μελετήσουμε σε αυτή τη διπλωματική εργασία.

Πιο συγκεκριμένα, η μελέτη των κυτταρικών δικτύων θα εστιαστεί στη μελέτη δικτύων καρκινικών κυττάρων, των οποίων οι Βιολόγοι μελετούν τις ιδιότητες, κάνοντας διάφορα πειράματα σε αυτά. Σ' αυτά τα πειράματα, προσθέτουν διάφορες ουσίες στο κύτταρο για να δουν πώς αντιδρά σε αυτές, με απώτερο σκοπό να ανακαλυφθεί μια θεραπεία για τον καρκίνο.

1.2 Στόχος Διπλωματικής Εργασίας

Απώτερος μας στόχος, είναι η δημιουργία ενός αυτόματου επιστημονικού βοηθού Μοριακής Βιολογίας, για τη μελέτη διεργασιών καρκινικών κυττάρων από Βιολόγους. Στα πλαίσια της παρούσας διπλωματικής εργασίας, όμως, θα επικεντρωθούμε στη δημιουργία μιας εύχρηστης γραφικής διεπαφής για τον επιστημονικό αυτό βοηθό. Η γραφική διεπαφή αυτή, θα πρέπει να έχει την κατάλληλη μορφή και να είναι εύχρηστη, έτσι ώστε να διευκολύνει το Βιολόγο στις έρευνες και στα πειράματα που θα κάνει σχετικά με τα καρκινικά κύτταρα.

Επιμέρους στόχοι, με απώτερο σκοπό την επίτευξη του βασικού μας στόχου, είναι η απόκτηση γνώσης σχετικά με την επιστήμη της Βιοπληροφορικής και πιο συγκεκριμένα σχετικά με τον τομέα της Συστημικής Βιολογίας [15] [34]. Η γνώση αυτή θα μας βοηθήσει στο να καταλάβουμε καλύτερα τις απαιτήσεις που θα έχει ο χρήστης (Βιολόγος) για το σύστημά μας. Για την εξαγωγή των απαιτήσεων αυτών, θα πρέπει να γίνουν διάφορες ενέργειες, όπως η δημιουργία κατάλληλων ερωτηματολογίων που θα δοθούν στο χρήστη, και μέσω διαφόρων συναντήσεων μαζί του, να αποκτήσουμε μια πλήρη εικόνα για το τι περιμένει να έχει το σύστημά μας μετά την ολοκλήρωσή του.

Μ' αυτές τις γνώσεις που θα πάρουμε από τις έρευνες και από τις συναντήσεις μας με το χρήστη, θα μπορούμε να προχωρήσουμε στον επόμενο επιμέρους στόχο, που είναι η έρευνα για να βρεθούν οι κατάλληλες μέθοδοι και τα κατάλληλα εργαλεία που θα βοηθήσουν στην ανάπτυξη του αυτόματου επιστημονικού βοηθού. Η έρευνα αυτή θα γίνει στους τομείς της Τεχνολογίας Λογισμικού και της Συστημικής Βιολογίας και επιπρόσθετα, θα πρέπει να μελετηθούν διάφορα εργαλεία που αφορούν την απεικόνιση βιολογικών δεδομένων, καθώς και διάφορες γλώσσες προγραμματισμού, μέχρι να καταλήξουμε στο κατάλληλο σύνολο τεχνολογιών που θα καλύπτει τις απαιτήσεις του χρήστη μας.

Όταν θα έχουμε υλοποιήσει το σύστημα, θα πρέπει να γίνει ο έλεγχος του συστήματός μας, ο οποίος θα γίνει μέσω συνάντησης με το χρήστη. Από τη συνάντηση αυτή, θα πάρουμε τις εντυπώσεις του και σχόλια σχετικά με την εμφάνιση και λειτουργία του συστήματος. Επίσης, τυχόν προβλήματα που μπορεί να υπάρξουν, θα εντοπιστούν και αν κάτι σχετικά με οποιοδήποτε μέρος του συστήματος, σύμφωνα με το χρήστη, χρειάζεται αλλαγή, θα μελετηθεί αν η αλλαγή αυτή είναι εφικτή κι αν είναι, θα γίνει.

Τέλος, θα πρέπει να μελετήσουμε με ποιους τρόπους θα μπορεί η έρευνα και κατ' επέκταση το σύστημά μας να επεκταθούν μελλοντικά για να καλύπτουν περισσότερες λειτουργίες από αυτές που θα έχουν υλοποιηθεί.

1.3 Δομή Διπλωματικής Εργασίας

Η Δομή της παρούσας Διπλωματικής Εργασίας, εξαρτάται από τα στάδια τα οποία περάσαμε μέχρι την ολοκλήρωσή της. Στα επόμενα κεφάλαια θα περιγραφούν τα στάδια αυτά, ένα στάδιο στο κάθε ένα.

Στο Κεφάλαιο 2 θα περιγραφεί το στάδιο της έρευνας σχετικά με το πρόβλημα. Θα γίνει μια περιγραφή του προβλήματος, μαζί με το βιολογικό υπόβαθρό του, με σκοπό να δώσει στον αναγνώστη τις κατάλληλες γνώσεις για την κατανόησή του. Επιπρόσθετα, θα γίνουν αναφορές και περιγραφές παλαιότερων, υπάρχοντων συστημάτων που έχουν παρόμοιους στόχους με το σύστημα που θα δημιουργήσουμε.

Στο Κεφάλαιο 3 θα περιγραφεί το στάδιο της εξαγωγής και ανάλυσης των απαιτήσεων και προδιαγραφών του συστήματος, όπως τις πήραμε από τη βασική χρήστη του συστήματος, κα. Χριστιάνα Νεοφύτου.

Το Κεφάλαιο 4 αποτελεί την έρευνα που έγινε σχετικά με τα διάφορα εργαλεία και τεχνολογίες, μέχρι να καταλήξουμε στο κατάλληλο σύνολο τεχνολογιών που θα χρησιμοποιήσουμε.

Το Κεφάλαιο 5 είναι το βασικότερο κεφάλαιο, στο οποίο παρουσιάζεται η σχεδίαση και η υλοποίηση του συστήματος.

Τα Κεφάλαια 6 και 7 είναι τα τελευταία στάδια για την ολοκλήρωση της παρούσας διπλωματικής εργασίας, τα στάδια του ελέγχου του συστήματος, της εξαγωγής αποτελεσμάτων και συμπερασμάτων, καθώς και προτάσεις για μελλοντική εργασία.

Κεφάλαιο 2

Περιγραφή Προβλήματος

2.1 Εισαγωγή	4
2.2 Συνοπτική Περιγραφή Προβλήματος	4
2.3 Βιολογική Υπόσταση Προβλήματος	5
2.4 Επεξήγηση Απεικονισμένου Κυτταρικού Δικτύου	7
2.5 Υπάρχοντα Συστήματα	9
2.6 Προσέγγιση Προβλήματος	14

2.1 Εισαγωγή

Σ' αυτό το κεφάλαιο, θα γίνει μια γενική περιγραφή του προβλήματος, αρχίζοντας με μια συνοπτική περιγραφή του. Στη συνέχεια, θα γίνει μια ανάλυση του προβλήματος με βάση τη βιολογική του υπόσταση, χωρίζοντάς το σε δύο υποπροβλήματα : το πρόβλημα της απόπτωσης και το πρόβλημα του πολλαπλασιασμού. Έπειτα, στην υποενότητα 2.4 θα περιγραφούν υπάρχοντα συστήματα, πώς κάποια από αυτά έλυσαν εν μέρει το πρόβλημα καθώς και ποιες ήταν οι ελλείψεις τους σχετικά με αυτό. Στην τελευταία υποενότητα, θα αναλυθεί η δική μου προσέγγιση για το πρόβλημα, εξηγώντας πώς κατέληξα στη λύση του.

2.2 Συνοπτική Περιγραφή Προβλήματος

Το πρόβλημα που ασχολείται η παρούσα διπλωματική εργασία, αφορά την επιστήμη της Βιοπληροφορικής και είναι η απεικόνιση και επεξεργασία κυτταρικών δικτύων (μονοπατιών), με απώτερο σκοπό την ηλεκτρονική επεξεργασία βιολογικών πειραμάτων πάνω σε καρκινικά κύτταρα. Ειδικότερα, ο χρήστης, που στην περίπτωσή μας είναι ο βιολόγος που κάνει τα πειράματα, θέλει να έχει τη δυνατότητα μέσω του συστήματος, να βλέπει τις διάφορες ουσίες, πρωτεΐνες ή διεργασίες ενός κυττάρου και να εισάγει τα πειραματικά δεδομένα σχετικά με αυτά. Ακολουθώντας, θα πρέπει να μπορεί να δει πώς αυτά τα δεδομένα αλλάζουν τις δύο βασικές διεργασίες που σχετίζονται με την καρκινογένεση, την

απόπτωση και τον πολλαπλασιασμό, όπως ανέφερα και πιο πάνω. Οι διεργασίες αυτές θα επεξηγηθούν λεπτομερώς στη συνέχεια.

2.3 Βιολογική Υπόσταση Προβλήματος – Συστημική βιολογία[15] [34]

Η συστημική βιολογία είναι ένας αρκετά νέος επιστημονικός κλάδος που σύμφωνα με τους βιολόγους που ασχολούνται με αυτή είναι ένας ριζικά καινούριος τρόπος σκέψης για τη βιολογία. Αντί να διερευνούνται τα χαρακτηριστικά των επιμέρους μερών ενός κυττάρου ή ενός οργανισμού, όπως οι βιοχημικοί και οι μοριακοί βιολόγοι έκαναν για πολλά χρόνια, οι συστημικοί βιολόγοι εστιάζονται σε ολόκληρο το βιολογικό σύστημα. Χαρακτηριστικό της συστημικής βιολογίας είναι η συνεργασία επιστημόνων από διάφορα πεδία, αφού για τη δόμηση και επαλήθευση κάθε θεωρίας απαιτούνται γνώσεις από το πεδίο της βιολογίας, των μαθηματικών, της φυσικής και της επιστήμης της πληροφορικής (βιοπληροφορική), καθώς και άλλους επιμέρους τομείς.

Ένα κύτταρο αποτελείται από διάφορες διεργασίες, πρωτεΐνες και ουσίες. Δύο σημαντικές διεργασίες του είναι η διαδικασία της απόπτωσης και του πολλαπλασιασμού οι οποίες αποτελούν τα δύο υποπροβλήματα της μελέτης αυτής.

2.3.1 Κυτταρικός Κύκλος και Απόπτωση [42]

Ο πολλαπλασιασμός του κυττάρου ή κυτταρικός κύκλος [39], είναι η διεργασία κατά την οποία ένα κύτταρο αναπαράγεται ακολουθώντας μια καθορισμένη ακολουθία γεγονότων τα οποία διπλασιάζουν το περιεχόμενο του κυττάρου και το διαιρούν. Αποτελεί τη βασική διεργασία με την οποία αναπαράγονται τα κύτταρα όλων των πολυκύτταρων οργανισμών. Για να εξασφαλισθεί η ορθή λειτουργία του κυτταρικού κύκλου, αναπτύχθηκε μέσα στα κύτταρα ένα περίπλοκο σύστημα ρυθμιστικών πρωτεϊνών, το σύστημα ελέγχου κυτταρικού κύκλου. Ένα είδος αυτών των ρυθμιστικών πρωτεϊνών είναι οι κυκλίνες. Στο κέντρο του συστήματος ελέγχου 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.

Στον κύκλο ζωής του κυττάρου, υπάρχουν ουσιαστικά δύο περίοδοι, η περίοδος μεσόφασης κατά την οποία το κύτταρο δε διαιρείται και η περίοδος διαίρεσης Μ η οποία για τα σωματικά κύτταρα είναι η μίτωση και η κυτταροκίνηση. Η περίοδος μεσόφασης υποδιαιρείται σε τρεις επιμέρους φάσεις, τη φάση G1 τη φάση S και τη φάση G2. Ουσιαστικά, τα κύτταρα αναπτύσσονται κατά τη διάρκεια της φάσης G1, το γενετικό τους υλικό αντιγράφεται κατά τη φάση S και κατά τη φάση G2 ολοκληρώνεται η ανάπτυξη και το κύτταρο προετοιμάζεται για την κυτταρική διαίρεση. Στη μίτωση, δηλαδή τη διαίρεση του πυρήνα, το γενετικό υλικό που είναι ήδη διπλασιασμένο και οργανωμένο σε χρωμοσώματα διαχωρίζεται, ποιοτικά και ποσοτικά, στα δύο θυγατρικά κύτταρα. Η φάση αυτή χωρίζεται σε έξι φάσεις. Η 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.

Η Απόπτωση [11], είναι μια φυσιολογική κυτταρική διαδικασία που χρειάζεται μεταξύ άλλων, για τη σωστή ανάπτυξη των οργάνων και την απομάκρυνση των ελαττωματικών ή πιθανώς βλαβερών κυττάρων στον ανθρώπινο οργανισμό. Ουσιαστικά, για κάθε κύτταρο υπάρχει η στιγμή της έναρξης της ζωής του και η στιγμή του θανάτου του. Πρόκειται για ένα είδος κυτταρικού θανάτου στον οποίο το κύτταρο φτάνει, ακολουθώντας ένα συγκεκριμένο γονιδιακό πρόγραμμα. Το πρόγραμμα αυτό ενεργοποιείται λαμβάνοντας συγκεκριμένα σήματα κυτταρικού θανάτου από το εσωτερικό ή εξωτερικό περιβάλλον του κυττάρου. Τα κύρια χαρακτηριστικά της απόπτωσης είναι η συρρίκνωση της χρωματίνης και η συρρίκνωση του κυττάρου χωρίς διάσπαση της κυτταρικής μεμβράνης. Τα κύρια ένζυμα της απόπτωσης είναι μια ομάδα πρωτεασών, οι κασπάσες.

Ένα κύτταρο θεωρείται αποπτωτικό όταν διακόπτονται η επικοινωνία του με τα γειτονικά κύτταρα και η αντιγραφή, μεταγραφή και επανόρθωση του DNA. Επίσης, χαρακτηριστικό είναι η εμφάνιση φωσφατιδυλο-σερίνης στην εξωτερική πλευρά της κυτταρικής μεμβράνης, απελευθερώνεται κυττόχρωμα C (Cyto C) από τα μιτοχόνδρια, γίνεται ενεργοποίηση ενδονουκλεασών και κασπασών (Caspase), γίνεται κατακερματισμός του DNA (DNA Fragmentation), τεμαχισμός των λαμινών και εκπέμπονται σήματα προσέλκυσης μακροφάγων για απομάκρυνση του κυττάρου.

Το πρόβλημα της καρκινογένεσης, το οποίο μελετάμε, είναι ένα θέμα που απασχολεί πολύ τις επιστήμες της Ιατρικής και της Βιολογίας. Χαρακτηρίζεται από τον ανεξέλεγκτο πολλαπλασιασμό καρκινικών κυττάρων, τα οποία ενώ σε φυσιολογικές συνθήκες θα έπρεπε να πέθαιναν με ένα αυστηρά ελεγχόμενο τρόπο, συνεχίζουν να πολλαπλασιάζονται με τον

τρόπο που αναφέραμε, χωρίς να φτάνουν ποτέ στο θάνατό τους, δηλαδή στη διαδικασία της απόπτωσης που εξηγήθηκε πιο πάνω.

Μ' αυτό τον τρόπο δημιουργούνται οι όγκοι ή κοινώς ο καρκίνος. Γενικότερα, τα καρκινικά κύτταρα χαρακτηρίζονται από το ότι δε φτάνουν ποτέ σε απόπτωση, δε φτάνουν ποτέ σε σήματα που σταματούν τον πολλαπλασιασμό, είναι ανθεκτικά στο μηχανισμό γήρανσης, κάνουν μεταστάσεις και δημιουργούν αγγεία για την αιμάτωσή τους.

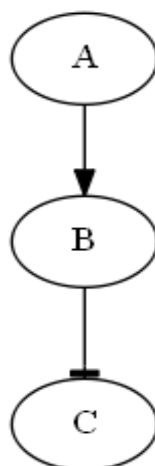
Οι Βιολόγοι, μελετούν τις ιδιότητες αυτών των κυττάρων, κάνοντας διάφορα πειράματα σε πραγματικά καρκινικά κύτταρα. Σ' αυτά τα πειράματα, προσθέτουν διάφορες ουσίες στο κύτταρο για να δουν πώς αντιδρά σε αυτές, με απώτερο σκοπό να ανακαλύψουν τρόπους με τους οποίους το κύτταρο αυτό να γίνει ξανά φυσιολογικό. Μ' αυτά τα πειράματα ασχολείται και η Βιολόγος και μελλοντική χρήστης του συστήματός μας, κα Χριστιάνα Νεοφύτου, στα πλαίσια των ερευνητικών της εργασιών. [9] [25]

2.4 Επεξήγηση Απεικονισμένου Κυτταρικού Δικτύου

Ένα κυτταρικό δίκτυο αποτελείται από πρωτεΐνες, διεργασίες και διάφορες άλλες ουσίες. Αυτά τα μέλη του κυττάρου μπορούν να υπάρξουν σε τρεις καταστάσεις. Μπορούν να είναι είτε ενεργά (active), είτε απενεργοποιημένα (inactive), είτε ουδέτερα (neutral).

Για την καλύτερη κατανόηση της απεικόνισης του κυττάρου, ας δούμε ένα παράδειγμα.

Ας υποθέσουμε ότι έχουμε τρεις πρωτεΐνες : πρωτεΐνη Α, πρωτεΐνη Β και πρωτεΐνη C.



Εικόνα 1 : Παράδειγμα απεικόνισης τριών πρωτεϊνών ενός κυτταρικού δικτύου

Για την απεικόνιση ενός κυτταρικού δικτύου θα πρέπει να ορίσουμε τα δύο είδη ακμών : την ενεργητική κατευθυνόμενη ακμή (activator), όπως φαίνεται από το Α στο Β και την ανασταλτική κατευθυνόμενη ακμή (inhibitor), όπως φαίνεται από το Β στο C.

Αν δύο πρωτεΐνες συνδέονται με ενεργητική ακμή, τότε όποια κι αν είναι η κατάσταση της πρωτεΐνης από την οποία προέρχεται η ακμή, μεταφέρεται στην πρωτεΐνη στην οποία κατευθύνεται η ακμή. Στο παράδειγμά μας δηλαδή, αν η πρωτεΐνη Α είναι ενεργή τότε θα είναι και η Β και αν η Α είναι απενεργοποιημένη, τότε θα είναι και η Β.

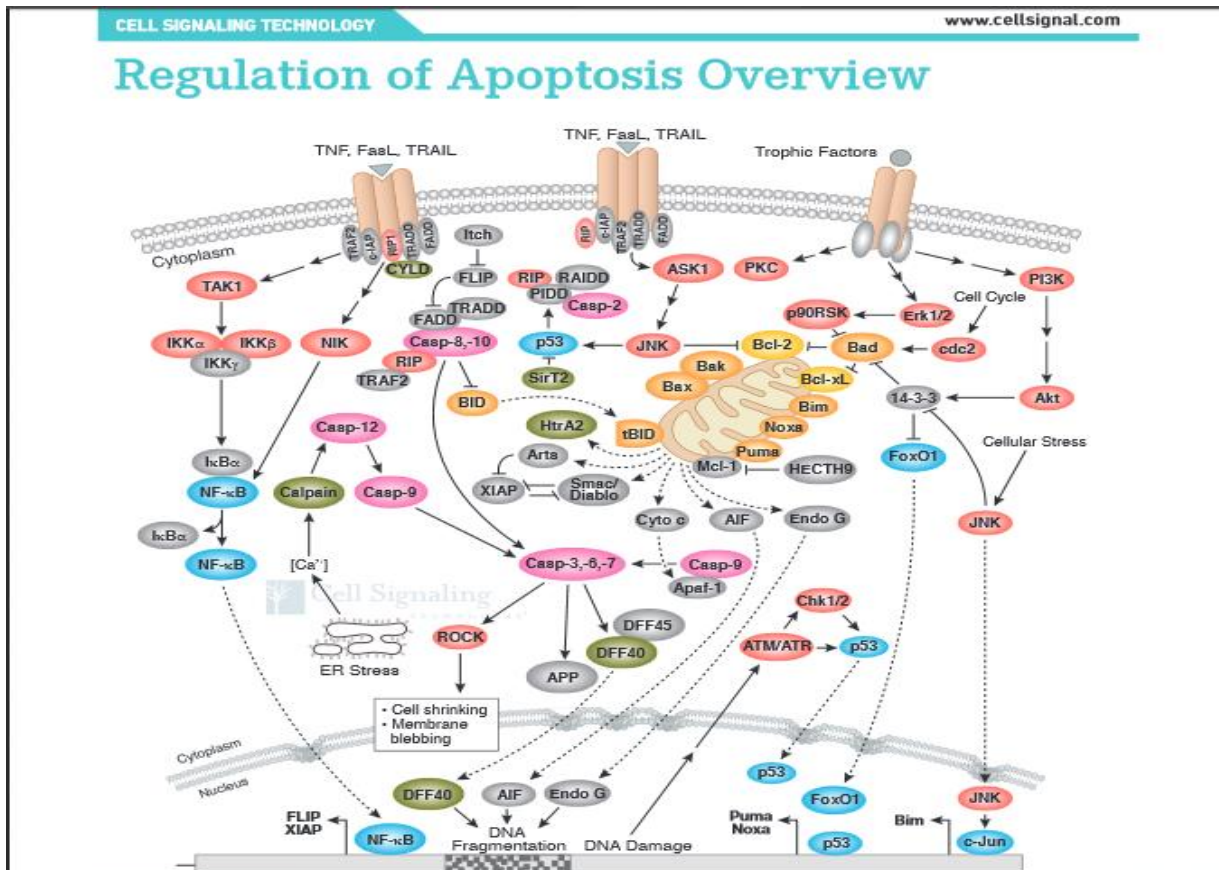
Αν δύο πρωτεΐνες συνδέονται με ανασταλτική ακμή και η πρωτεΐνη από την οποία προέρχεται η ακμή είναι ενεργή, τότε αυτόματα απενεργοποιείται η πρωτεΐνη στην οποία κατευθύνεται η ακμή. Στο παράδειγμα, αν η Β είναι ενεργή τότε η C απενεργοποιείται. Αν η Β είναι απενεργοποιημένη, τότε δε μπορεί να απενεργοποιήσει 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.

Αξίζει να σημειωθεί ότι εάν κατευθύνονταν περισσότερες από μία ακμές σε μία πρωτεΐνη, τότε οι ακμές με το ανασταλτικό σήμα θα κυριαρχούσαν αυτών με το ενεργητικό.

2.5 Υπάρχοντα Συστήματα

2.5.1 Η ιστοσελίδα Cell Signaling [6]

Η Cell Signaling είναι μια ιστοσελίδα η οποία μεταξύ άλλων υπηρεσιών προς Βιολόγους, παρέχει και απεικονίσεις διαφόρων διεργασιών- μονοπατιών- των κυττάρων. Ανάμεσα σ' αυτές είναι και οι διεργασίες που μελετάμε, της απόπτωσης και του πολλαπλασιασμού. Οι απεικονίσεις αυτές όμως, παρόλο που είναι πολύ χρήσιμες στους Βιολόγους, δεν παρέχουν τη δυνατότητα επεξεργασίας των μονοπατιών μετά από πειράματα.



Εικόνα 2 : Απεικόνιση Δικτύου Απόπτωσης από τη σελίδα της Cell Signaling [6]

2.5.2 Το σύστημα Tulip [3]

Το σύστημα Tulip είναι ένα σύστημα οπτικοποίησης πληροφοριών, αφιερωμένο στην ανάλυση και παρουσίαση σχεσιακών δεδομένων. Κύριος στόχος του είναι να παρέχει στους προγραμματιστές μια ολοκληρωμένη βιβλιοθήκη, που υποστηρίζει το σχεδιασμό διαδραστικών εφαρμογών για απεικόνιση πληροφοριών από σχεσιακά δεδομένα και που να μπορεί να προσαρμοστεί σε προβλήματα που ο προγραμματιστής αντιμετωπίζει.

Είναι γραμμένο σε C++ και επιτρέπει την ανάπτυξη αλγορίθμων, οπτικών κωδικοποιήσεων, διαδραστικών τεχνικών, μοντέλων δεδομένων και απεικονίσεων για συγκεκριμένους τομείς, όπως στην περίπτωση μας η απεικόνιση του κυττάρου. Ένας από τους στόχους του Tulip είναι η διευκόλυνση της επαναχρησιμοποίησης των συστατικών μελών και επιτρέπει στον προγραμματιστή να επικεντρωθεί στον προγραμματισμό της εφαρμογής. Είναι κατάλληλο για πρωτότυπα ερευνών, καθώς και για την ανάπτυξη εφαρμογών για τον τελικό χρήστη.

2.5.3 Το σύστημα CellDesigner [12]

Το σύστημα CellDesigner είναι ένα δομημένο πρόγραμμα επεξεργασίας διαγραμμάτων για σχεδιασμό βιοχημικών δικτύων. Τα δίκτυα είναι σχεδιασμένα βάση του διαγράμματος διεργασίας, με σύστημα σημειογραφίας γραφικών προτεινόμενο από τον Kitano, και είναι καταχωρημένα χρησιμοποιώντας τη «Γλώσσα Σήμανσης Συστημικής Βιολογίας» (Systems Biology Markup Language (SBML)) [18] που είναι πρότυπο για την αναπαράσταση μοντέλων βιοχημικών δικτύων. Τα δίκτυα μπορούν να συνδεθούν με προσομοιώσεις και άλλα πακέτα ανάλυσης μέσω του Systems Biology Workbench (SBW) [31].

Το CellDesigner υποστηρίζει προσομοιώσεις και σάρωση παραμέτρων με την ενσωμάτωση άλλων εργαλείων. Με τη χρήση του, μπορούμε να περιηγηθούμε και να επεξεργαστούμε ήδη υπάρχοντα SBML μοντέλα με αναφορές σε υπάρχουσες βάσεις δεδομένων.

2.5.4 Το σύστημα PyDSTool [8]

Το σύστημα PyDSTool είναι σχεδιασμένο για την ανάπτυξη, προσομοίωση και ανάλυση δυναμικών μοντέλων συστημάτων, συγκεκριμένα για βιολογικές εφαρμογές. Σε αντίθεση με τα περισσότερα εργαλεία προσομοίωσης γενικών σκοπών, το PyDSTool παρέχει ένα προγραμματιστικό περιβάλλον κατάλληλο για διερεύνηση δεδομένων και υποθέσεων με γνώμονα προβλήματα κατασκευής βιολογικών μοντέλων.

Το PyDSTool είναι μια ανεξάρτητη πλατφόρμα, γραμμένη κυρίως σε Python με βάσεις σε C και Fortran για πιο γρήγορη απόδοση. Κάνει εκτεταμένη χρήση των βιβλιοθηκών numpy και scipy. Είναι εντελώς open source με BSD licence [5].

2.5.4 Το σύστημα ApoCelSys [38]

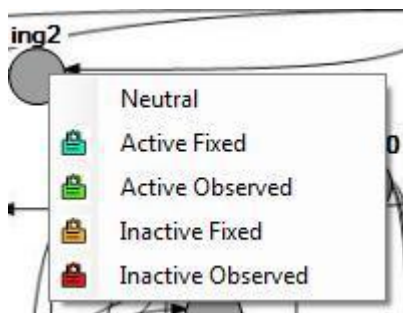
Το σύστημα ApoCelSys δημιουργήθηκε το 2010 στα πλαίσια της μεταπτυχιακής διατριβής του κου. Σωτήρη Λαζάρου στο Πανεπιστήμιο Κύπρου. Είναι το σύστημα το οποίο χρησιμοποιούσε μέχρι τώρα ο χρήστης μας και χρειαζόταν αναβάθμιση.

Το σύστημα ουσιαστικά, προσομοιώνει ένα κύτταρο που διαθέτει μόνο τη βασική λειτουργία της απόπτωσης και δημιουργεί υποθέσεις ως προς την επίδραση κάποιου φαρμάκου στα γονίδιά του, μετά από παρατηρήσεις που έκαναν σε αυτό οι βιολόγοι. Λειτουργεί με Απαγωγικό Λογικό Προγραμματισμό[1] [21] για να μπορέσει να κάνει τις υποθέσεις και να βγάλει τα ανάλογα συμπεράσματα.

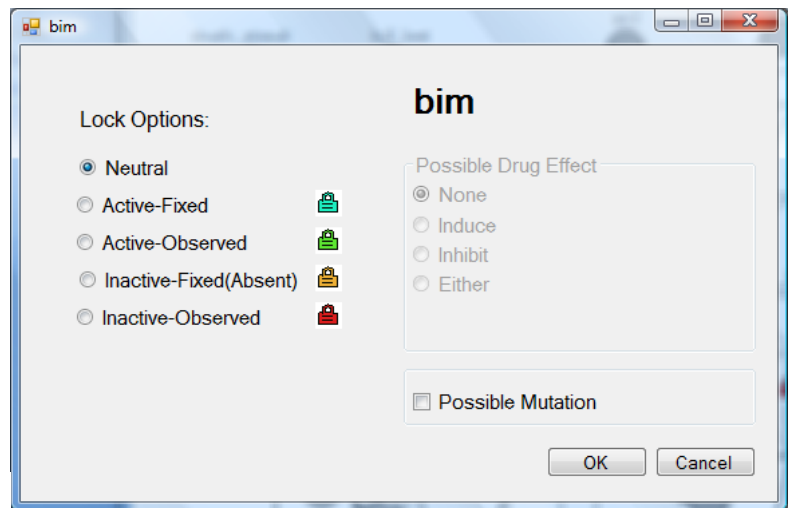
Σ' αυτό το σημείο, αξίζει να ορίσουμε τι είναι ο Απαγωγικός Λογικός Προγραμματισμός (Abductive Logic Programming - ALP) [1] [21]. Ορίζεται ως ένα πλαίσιο υψηλού επιπέδου για εκπροσώπηση γνώσης, που μπορεί να χρησιμοποιηθεί για επίλυση προβλημάτων με δηλωτικό τρόπο, βασιζόμενο σε απαγωγικό λογισμό. Θεωρείται επέκταση του κανονικού λογικού προγραμματισμού, η οποία επιτρέπει κάποια κατηγορήματα να μην είναι εντελώς ορισμένα, ορίζοντάς τα ως *abducible*. Ο Απαγωγικός Λογικός Προγραμματισμός, χρησιμοποιείται για την επίλυση προβλημάτων στους τομείς της Διάγνωσης, του Σχεδιασμού, της Φυσικής Γλώσσας και της Μηχανικής Μάθησης.

Όσον αφορά τις λειτουργίες του συστήματος ApoCelSys, αρχικά, ο βιολόγος εισάγει στο σύστημα τα δεδομένα που σχετίζονται για το κύτταρο και την κατάσταση του κάθε συστατικού του μέλους (ενεργό, απενεργοποιημένο). Η εισαγωγή γίνεται είτε με διπλό κλικ πάνω στο γονίδιο είτε με δεξί κλικ. Στη συνέχεια, με τη χρήση του απαγωγικού λογικού προγραμματισμού, επιστρέφονται στο χρήστη τα αποτελέσματα που είναι κάποιες επεξηγήσεις και υποθέσεις. Ο χρήστης τότε, αναμένει να δει το σύνολο των στοιχείων που υποθετικά επέδρασε το φάρμακο, καθώς και κάποια στατιστικά.

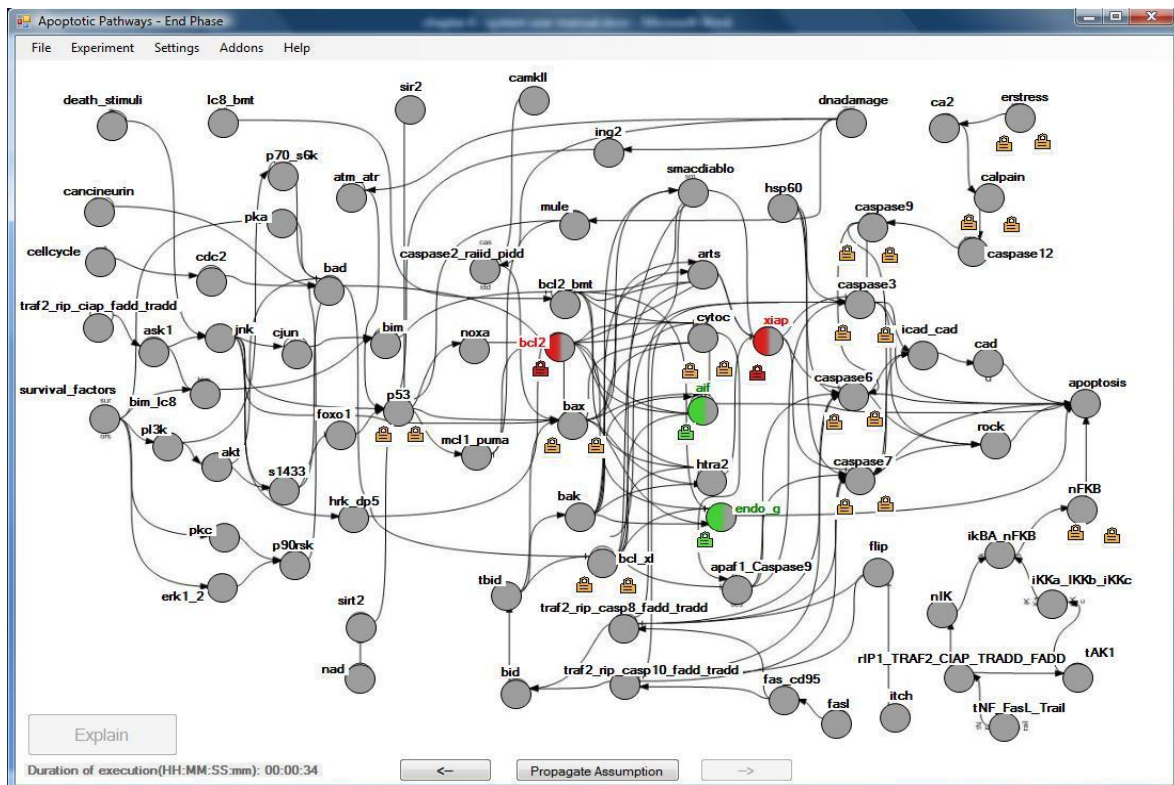
Παρακάτω φαίνονται οι δύο διαφορετικοί τρόποι εισαγωγής των πειραματικών δεδομένων, ένα στιγμιότυπο του συστήματος και ένα στιγμιότυπο μετά τη διάδοση του σήματος (propagation).



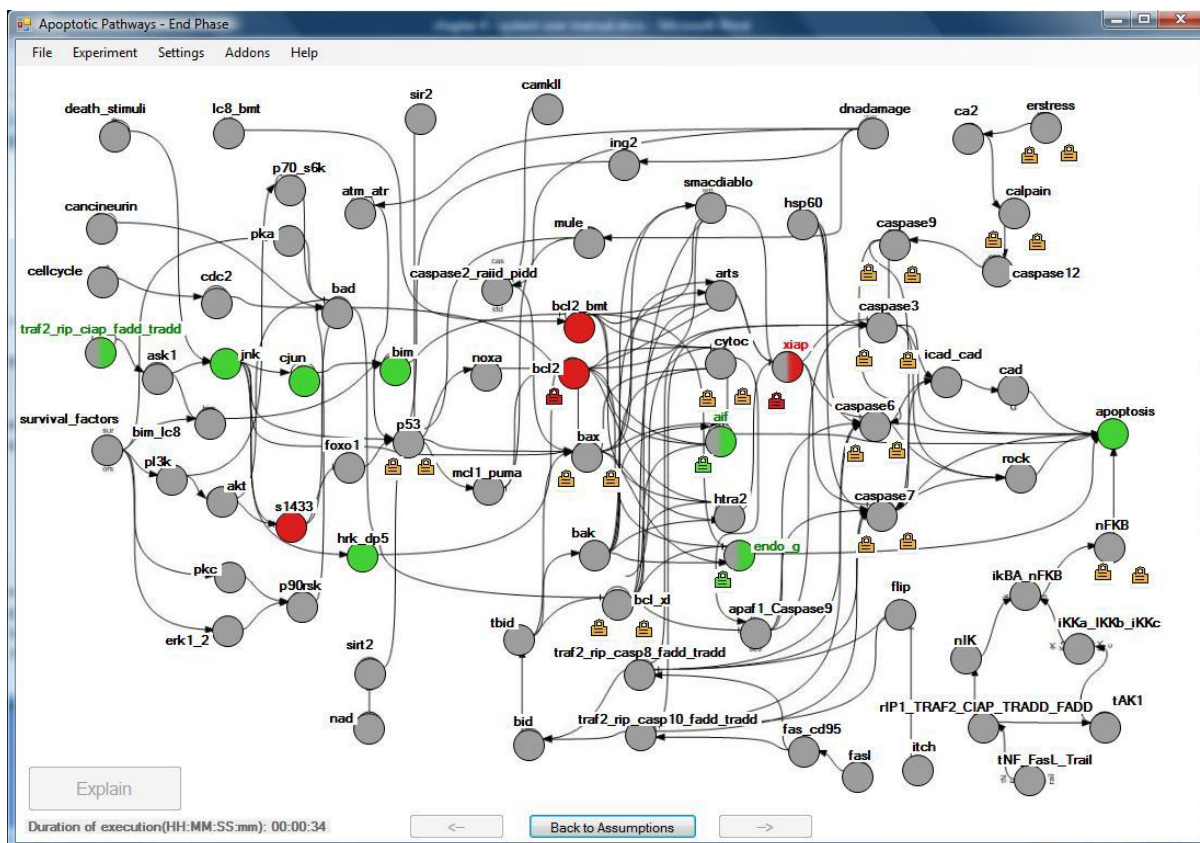
Εικόνα 3 [38] : Παράδειγμα σύντομου μενού για επιλογή της κατάστασης ενός γονιδίου (εμφάνισή του με δεξί κλικ πάνω στο γονίδιο)



Εικόνα 4 [38]: Παράδειγμα ολοκληρωμένου μενού για επιλογή της κατάστασης ενός γονιδίου (εμφάνισή του με διπλό κλικ πάνω στο γονίδιο)



Εικόνα 5 [38] : Αναπαράσταση Μονοπατιών και Μενού Εντολών



Εικόνα 6 [38] : Στιγμιότυπο του συστήματος μετά τη διάδοση του σήματος (propagation)

2.6 Προσέγγιση Προβλήματος

Όσον αφορά τη δική μου προσέγγιση για το πρόβλημα, αρχικά θα συνεχίζαμε από το σημείο που έμεινε το τελευταίο σύστημα (σύστημα ApoCelSys). Θα γινόταν μια ανανέωση του συστήματος, με την προσθήκη νέων απαιτήσεων που θα λαμβάνονταν από το χρήστη μέσω ενός ερωτηματολογίου (βλ. Παράρτημα Α).

Με την πάροδο του χρόνου και έχοντας μαζέψει όλες τις απαιτήσεις του χρήστη, σκεφτήκαμε πως θα ήταν καλύτερο αν δημιουργούσαμε ένα καινούργιο σύστημα από την αρχή, κι έτσι η προσέγγιση ξεκίνησε εντελώς από το μηδέν. Γι' αυτό το λόγο, το λειτουργικό κομμάτι του συστήματος, δηλαδή το κομμάτι εισαγωγής του απαγωγικού λογικού προγραμματισμού, δε θα μας απασχολήσει στα πλαίσια αυτής της διπλωματικής εργασίας. Θα πρέπει όμως να σχεδιαστεί και να υλοποιηθεί μια βελτιωμένη γραφική διεπαφή, που θα έχει ως βάση την εμφάνιση του συστήματος ApoCelSys, στην οποία τα μονοπάτια του κυττάρου θα είναι απεικονισμένα με περισσότερη σαφήνεια και θα γίνουν βελτιώσεις σύμφωνα με τις απαιτήσεις του χρήστη.

Για τη δημιουργία του νέου συστήματος θα πρέπει να βρεθούν τα κατάλληλα εργαλεία απεικόνισης κυτταρικών δικτύων, καθώς και εργαλεία για τη δημιουργία γραφικής διεπαφής χρήστη (GUI), έτσι ώστε να ικανοποιούνται οι απαιτήσεις του χρήστη που θα περιγραφούν στο επόμενο κεφάλαιο.

Κεφάλαιο 3

Απαιτήσεις - Προδιαγραφές

3.1 Εισαγωγή	15
3.2 Ανάλυση Απαιτήσεων – Προδιαγραφών	15

3.1 Εισαγωγή

Η εύρεση των απαιτήσεων και προδιαγραφών για το σύστημα έγινε μέσω συναντήσεων με τη βασική μας χρήστη. Πιο συγκεκριμένα, με τη δημιουργία του κατάλληλου ερωτηματολογίου (βλ. Παράρτημα Α) πήραμε τις απαντήσεις που ζητούσαμε, οι οποίες μας βοήθησαν πολύ στο να έχουμε μια πιο ολοκληρωμένη άποψη για τις απαιτήσεις του χρήστη από το σύστημα. Σ' αυτό το κεφάλαιο θα γίνει αναφορά σ' αυτές τις απαιτήσεις και προδιαγραφές, καθώς και ανάλυσή τους μια προς μια.

3.2 Ανάλυση Απαιτήσεων - Προδιαγραφών

3.2.1 Ασφαλής Είσοδος στο Σύστημα

Το σύστημα θα πρέπει να χρησιμοποιείται μόνο από επικυρωμένους χρήστες, γιατί μέσω αυτού θα διαβάζονται πειραματικά δεδομένα τα οποία θα είναι προσωπικά για τον κάθε χρήστη. Έτσι, σε κάθε χρήστη που θα εγκαθίσταται το σύστημα θα του δίνουμε εμείς ένα προσωπικό κωδικό, ο οποίος θα ζητείται κάθε φορά κατά την εκκίνηση του συστήματος. Ο κωδικός αυτός θα είναι μοναδικός και θα καθορίζεται από πριν.

3.2.2 Αρχική Οθόνη

Όταν ο χρήστης εισάγει τον κωδικό πρόσβασής του, θα ανοίγει την αρχική οθόνη του συστήματος. Η οθόνη αυτή θα πρέπει να περιέχει τέσσερα κουμπιά τα οποία θα δίνουν από μια επιλογή στο χρήστη. Οι επιλογές αυτές θα είναι οι ακόλουθες :

3.2.2.1 Μονοπάτι Απόπτωσης

Το μονοπάτι αυτό θα περιέχει τις πρωτεΐνες, διεργασίες και ουσίες, καθώς και τις ακμές ανάμεσά τους, όπως φαίνονται στην ιστοσελίδα της Cell Signaling, συγκεκριμένα στο μονοπάτι που απεικονίζει τη ρύθμιση της απόπτωσης.

3.2.2.2 Μονοπάτι Πολλαπλασιασμού

Το μονοπάτι αυτό θα περιέχει ένα κομμάτι του μονοπατιού για το σημείου ελέγχου του κυτταρικού πολλαπλασιασμού, όπως φαίνεται στην ιστοσελίδα της Cell Signaling. Το κομμάτι αυτό καθορίστηκε από το χρήστη.

3.2.2.3 Δημιουργία Νέου Κυτταρικού Δικτύου/Μονοπατιού

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

3.2.2.4 Επιλογή για Έξοδο

Μ' αυτή την επιλογή, θα γίνεται έξοδος από το σύστημα.

3.2.3 Αντιστοιχία Μεταξύ των Διαφόρων Οθονών

Όταν ο χρήστης βρίσκεται είτε στην οθόνη της απόπτωσης, είτε στην οθόνη του πολλαπλασιασμού, θα πρέπει πατώντας δεξί κλικ πάνω σε μια ουσία, πρωτεΐνη ή διεργασία, να του δίνεται η δυνατότητα να μεταβεί στην ίδια ουσία, πρωτεΐνη ή διεργασία, στην αντίστοιχη άλλη οθόνη. Όταν γίνει αυτή η μετάβαση, η ουσία, πρωτεΐνη ή διεργασία θα πρέπει να είναι τονισμένη με διαφορετικό χρώμα για να ξεχωρίζει από τις υπόλοιπες. Αν η ουσία που επιλέχθηκε δεν υπάρχει στην αντίστοιχη άλλη οθόνη, θα εμφανίζεται ένα ανάλογο μήνυμα.

3.2.4 Επιλογές Open και Save

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

3.2.5 Έξοδος από μια Οθόνη

Σε κάθε μια 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.

3.2.6 Επιλογή Αναζήτησης

Θα πρέπει ο χρήστης να μπορεί να αναζητήσει μια πρωτεΐνη, ουσία ή διεργασία μέσα στο μονοπάτι που βρίσκεται. Η αναζήτηση αυτή θα γίνεται μέσω εισαγωγής του ονόματός της. Αν υπάρχει, το σύστημα θα πρέπει να τον παραπέμπει με κάποιο τρόπο σε αυτή. Αν όχι, θα βγάζει το ανάλογο μήνυμα.

3.2.7 Zoom

Σε κάθε μια από τις προαναφερόμενες οθόνες, θα πρέπει να υπάρχει η επιλογή για zoom με σημεία ένδειξης (+) για μεγέθυνση (zoom-in) και (-) για σμίκρυνση (zoom-out) για διευκόλυνση του χρήστη.

3.2.8 Απεικόνιση Μονοπατιών

Όσον αφορά τη μορφή που θα έχουν οι κόμβοι σε κάθε μονοπάτι, σύμφωνα με τις προτιμήσεις του χρήστη, 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.

3.2.9 Εισαγωγή Πειραματικών Δεδομένων

Η εισαγωγή των πειραματικών δεδομένων θα γίνεται με δεξί κλικ πάνω σε μια πρωτεΐνη, ουσία ή διεργασία. Πατώντας δεξί κλικ, θα πρέπει ο χρήστης να 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 είναι active fixed, το περίγραμμα της θα γίνεται γαλάζιο, αν είναι inactive fixed το περίγραμμα της θα γίνεται πορτοκαλί, αν είναι active observed το περίγραμμα της θα γίνεται πράσινο και αν είναι inactive observed, το περίγραμμα της θα γίνεται κόκκινο.

3.2.10 Απεικόνιση Διάδοσης Σήματος

Όταν θα γίνεται η διάδοση του σήματος, οι κόμβοι από τους οποίους πέρασε σήμα ενεργοποίησης, θα γεμίζουν με πράσινο χρώμα και οι κόμβοι από τους οποίους πέρασε σήμα απενεργοποίησης, θα γεμίζουν με κόκκινο χρώμα.

3.2.11 Δυνατότητα Πρόσθεσης Νέων Κόμβων και Συνδέσεων

Στο σύστημα θα υπάρχει μια επιλογή «add» η οποία όταν επιλεγθεί θα εμφανίζεται ένα παράθυρο στο οποίο 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 άλλες ουσίες καθώς και τον τύπο των συνδέσεων. Επίσης θα υπάρχει η επιλογή για πρόσθεση νέας σύνδεσης ανάμεσα σε υπάρχοντες κόμβους.

Κεφάλαιο 4

Απαιτούμενη Γνώση και Τεχνολογίες

4.1 Εισαγωγή	19
4.2 Σύνολο Τεχνολογιών 1 : GUI σε MATLAB, NetworkX, Python	20
4.3 Σύνολο Τεχνολογιών 2 : GUI σε wxPython, NetworkX	23
4.4 Σύνολο Τεχνολογιών 3 : GUI σε Tkinter(Python), Graphviz και Λ.Σ. Linux	24
4.5 Σύνολο Τεχνολογιών 4 : GUI σε Java, Graphviz	25

4.1 Εισαγωγή

Στην εποχή που ζούμε, όλο και περισσότερα επιστημονικά συστήματα και εργαλεία αναπτύσσονται καθημερινά, πράγμα που κάνει τη ζωή των προγραμματιστών εφαρμογών ευκολότερη. Αυτή τη στιγμή υπάρχουν πολλά εργαλεία που βοηθούν στη δημιουργία γραφικών διεπαφών, καθώς και εργαλεία επεξεργασίας και απεικόνισης βιολογικών δικτύων.

Για τη δημιουργία του συστήματος, έγινε εκτεταμένη έρευνα σχετικά με εργαλεία απεικόνισης και επεξεργασίας βιολογικών δικτύων. Μελετήθηκαν όλα τα εργαλεία που αναφέρθηκαν στο κεφάλαιο 2.5 και καταλήξαμε στο συμπέρασμα ότι κανένα από αυτά δε θα ήταν βοηθητικό στην περίπτωσή μας, λόγω του ότι ο χρήστης θα χρειαζόταν εξειδικευμένες γνώσεις στην πληροφορική και επίσης όλα τα συστήματα που αναφέρθηκαν δεν είχαν τις δυνατότητες εκτεταμένης επεξεργασίας των βιολογικών δικτύων για τη δημιουργία υποθέσεων.

Για να υπάρξει η δυνατότητα ικανοποίησης όλων των απαιτήσεων του χρήστη, έπρεπε να βρεθούν τα κατάλληλα εργαλεία τα οποία να δημιουργούν γράφους, έτσι ώστε να γίνει σωστή απεικόνιση του βιολογικού δικτύου. Επιπρόσθετα, έπρεπε να γίνει μελέτη για να βρεθούν γλώσσες προγραμματισμού, οι οποίες να δίνουν τη δυνατότητα δημιουργίας μιας καλής και εύχρηστης γραφικής διεπαφής (GUI) για το χρήστη.

Στα επόμενα υποκεφάλαια, θα γίνει περιγραφή τεσσάρων συνόλων τεχνολογιών και εργαλείων τα οποία μελετήθηκαν ένα προς ένα, ξεκινώντας από το σύνολο του υποκεφαλαίου 4.2 και καταλήγοντας στο τελικό σύνολο που περιγράφεται στο υποκεφάλαιο 4.5.

4.2 Σύνολο Τεχνολογιών 1 : GUI σε MATLAB, NetworkX, Python

4.2.1 Γλώσσα Προγραμματισμού MATLAB [24]

Η γλώσσα προγραμματισμού MATLAB είναι μια γλώσσα υψηλού επιπέδου και διαδραστικό περιβάλλον για αριθμητικούς υπολογισμούς, οπτικοποίηση και προγραμματισμό. Χρησιμοποιώντας τη MATLAB, μπορεί να γίνει ανάλυση δεδομένων, ανάπτυξη αλγορίθμων και δημιουργία μοντέλων και εφαρμογών. Η γλώσσα, τα εργαλεία και οι ενσωματωμένες μαθηματικές συναρτήσεις, επιτρέπουν στο χρήστη να εξερευνήσει πολλαπλές προσεγγίσεις και να φτάσει σε λύση πιο γρήγορα από ότι με τα λογιστικά φύλλα ή τις παραδοσιακές γλώσσες προγραμματισμού, όπως η C / C++ ή Java™.

Η MATLAB μπορεί να χρησιμοποιηθεί για ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένης της επεξεργασίας σήματος και επικοινωνίες, επεξεργασίας εικόνας και βίντεο, των σημάτων ελέγχου, δοκιμών και μετρήσεων, υπολογιστικής χρηματοδότησης και υπολογιστικής βιολογίας. Περισσότεροι από ένα εκατομμύριο μηχανικοί και επιστήμονες στον τομέα της βιομηχανίας και της ακαδημαϊκής κοινότητας χρησιμοποιούν τη MATLAB.

Η γλώσσα αυτή, μελετήθηκε στα πλαίσια της εύρεσης των κατάλληλων εργαλείων για την ανάπτυξη του συστήματος, αρχικά για τα εργαλεία υπολογιστικής βιολογίας που παρέχει η ίδια. Τα εργαλεία αυτά είναι δύο, το SBTOOLBOX1 [16] και το SBPOP [35] που περιέχει μέσα το SBTOOLBOX2 [33]. Τα δύο αυτά εργαλεία όμως, θα μπορούσαν να χρησιμοποιηθούν μόνο για την απαίτηση της παραγωγής στατιστικών αποτελεσμάτων, γιατί δεν παρέχουν δυνατότητες απεικόνισης βιολογικών συστημάτων.

Μια δεύτερη προδιαγραφή της MATLAB που μελετήθηκε, είναι η δυνατότητα δημιουργίας μιας καλής και εύχρηστης γραφικής διεπαφής για το χρήστη. Στα πλαίσια αυτής της έρευνας, μελετήθηκε το εργαλείο GUIDE (GUI development environment) που παρέχει η MATLAB.

Το εργαλείο GUIDE, παρέχει εργαλεία για σχεδιασμό διεπαφών χρήστη για συγκεκριμένες εφαρμογές. Χρησιμοποιώντας τον GUIDE Layout Editor, μπορεί να γίνει γραφική σχεδίαση των διεπαφών. Όταν γίνει η σχεδίαση, το GUIDE δημιουργεί αυτόματα τον κώδικα σε MATLAB για τη δημιουργία της διεπαφής, τον οποίο μπορούμε να επεξεργαστούμε για να δώσουμε στην εφαρμογή μας τη μορφή και τις λειτουργίες που θέλουμε. Το εργαλείο αυτό, προγραμματιστικά είναι πολύ χρήσιμο για το σύστημα που θα αναπτύξουμε, γιατί μέσα στον κώδικα που δημιουργείται αυτόματα, συμπεριλαμβάνονται και η υλοποίηση zoom-in και zoom-out, καθώς και η υλοποίηση των Open και Save.

4.2.2 Γλώσσα προγραμματισμού Python [36]

Η Python είναι μια ευρέως διαδεδομένη γλώσσα προγραμματισμού υψηλού επιπέδου, για γενικούς σκοπούς. Δημιουργήθηκε από τον Ολλανδό Guido van Rossum το 1990. Η σχεδιαστική φιλοσοφία της, δίνει έμφαση στην αναγνωσιμότητα του κώδικα, και η σύνταξή της επιτρέπει στον προγραμματιστή να εκφράσει έννοιες με λιγότερες γραμμές κώδικα από ότι θα τις εξέφραζε με άλλες γλώσσες προγραμματισμού όπως τη C. Η γλώσσα αυτή, παρέχει δομές που προορίζονται για να είναι δυνατή η δημιουργία σαφών προγραμμάτων, τόσο με μικρό μέγεθος κώδικα, όσο και με μεγάλο.

Η Python υποστηρίζει πολλαπλά πρότυπα προγραμματισμού, συμπεριλαμβανομένων του αντικειμενοστραφή προγραμματισμού (object-oriented), του επιτακτικού προγραμματισμού (imperative) και του προγραμματισμού με συναρτήσεις ή διαδικασίες (functional / procedural). Διαθέτει δυναμικό σύστημα τύπων και αυτόματη διαχείριση μνήμης και μια μεγάλη και ένα μεγάλο και ολοκληρωμένο πρότυπο βιβλιοθήκης.

Όπως και άλλες δυναμικές γλώσσες, η Python συχνά χρησιμοποιείται ως scripting language [26], αλλά επίσης, χρησιμοποιείται ευρέως και σε non-scripting πλαίσια. Οι διερμηνείς της Python είναι διαθέσιμοι για πολλά λειτουργικά συστήματα.

Η Python συμπεριλήφθηκε στην έρευνά μας γιατί προσφέρει πληθώρα βιβλιοθηκών για βιολογικά εργαλεία, καθώς και για εργαλεία που αφορούν την απεικόνιση γράφων, δικτύων και γραφημάτων. Η έκδοση Python που χρησιμοποιήθηκε είναι η Python 2.7 λόγω του ότι οι περισσότερες βιβλιοθήκες αφορούν αυτή την έκδοση.

Οι βιβλιοθήκες που μελετήθηκαν είναι οι BioPython [7], CellProfiler [20] και libSBML [4] που αφορούν βιολογικά συστήματα και οι NetworkX [17], Matplotlib [19] και Graphviz

[10] που αφορούν απεικόνιση γράφων, δικτύων και γραφημάτων. Οι πρώτες τρεις που αφορούν βιολογικά συστήματα, δε χρησιμοποιήθηκαν γιατί οι απεικονίσεις που προσέφεραν δεν ήταν οι κατάλληλες για να ικανοποιούνται οι απαιτήσεις του χρήστη μας. Γι' αυτό το λόγο, η προσέγγιση γίνεται πιο γενική, ξεκινώντας από την απεικόνιση του βιολογικού δικτύου από την αρχή, ως ένας γράφος. Οι δύο υποψήφιες βιβλιοθήκες για το σκοπό αυτό είναι οι Graphviz που θα μελετήσουμε αργότερα και NetworkX, που λειτουργεί μέσω της Matplotlib, την οποία θα αναλύσουμε.

4.2.3 Το εργαλείο NetworkX [17]

Το NetworkX είναι ένα πακέτο λογισμικού για τη γλώσσα Python, για τη δημιουργία, το χειρισμό και τη μελέτη της δομής, των δυναμικών και των λειτουργιών πολύπλοκων δικτύων. Δημιουργήθηκε το Μάιο του 2002. Η αρχική έκδοση σχεδιάστηκε και υλοποιήθηκε από τους Aric Hagberg, Dan Schult και Pieter Swart το 2002 και το 2003. Η πρώτη δημόσια έκδοσή του έγινε τον Απρίλιο του 2005. Είναι ελεύθερο λογισμικό και επιτρέπεται η αναδιανομή ή/και τροποποίησή του, κάτω από τους όρους της BSD License [5].

Με το NetworkX μπορούμε να φορτώσουμε και να αποθηκεύσουμε δίκτυα σε συνηθισμένες και μη-συνηθισμένες μορφές δεδομένων, να δημιουργήσουμε πολλά είδη τυχαίων και κλασικών δικτύων, να κτίσουμε μοντέλα δικτύων, να σχεδιάσουμε νέους αλγόριθμους δικτύων, να σχεδιάσουμε δίκτυα κ.ο.κ.

Το NetworkX παρέχει εργαλεία για τη μελέτη της δομής των κοινωνικών και βιολογικών δικτύων και των δικτύων υποδομής. Επίσης, το προγραμματιστικό του περιβάλλον και η υλοποίηση γραφημάτων που παρέχει, είναι κατάλληλα για πολλές εφαρμογές και το γρήγορο περιβάλλον ανάπτυξής του μπορεί να χρησιμοποιηθεί για έρευνες σε πολλούς επιστημονικούς τομείς.

Για το σύστημα που θα αναπτύξουμε, το εργαλείο αυτό είναι πολύ χρήσιμο, αφού μπορεί να γίνει η σχεδίαση του βιολογικού δικτύου και στη συνέχεια να παραχθεί η κατάλληλη εικόνα την οποία θα εισάγουμε στη γραφική διεπαφή. Επιπρόσθετα, με τη χρήση της βιβλιοθήκης Matplotlib [19], μπορούμε να απεικονίσουμε και τα στατιστικά συμπεράσματα που συμπεριλαμβάνονται στις απαιτήσεις.

4.2.4 Συνδυασμός MATLAB, Python και NetworkX

Μέσα από τη μελέτη των τριών τεχνολογιών, καταλήγουμε στο συμπέρασμα ότι για την ολοκλήρωση του συστήματος πρέπει οι τρεις αυτές τεχνολογίες να συνδυαστούν με τον κατάλληλο τρόπο.

Αρχικά, θα πρέπει να δημιουργηθεί η γραφική διεπαφή στη MATLAB στην οποία θα εισαχθεί η εικόνα με το απεικονισμένο βιολογικό δίκτυο. Η εικόνα αυτή, θα παραχθεί από ένα script γραμμένο σε Python, με τη χρήση της βιβλιοθήκης του NetworkX και θα καλεστεί στη MATLAB μέσω της εντολής : `python(String, LIST_OF_ARGUMENTS)` , όπου `String` είναι το όνομα του script και `LIST_OF_ARGUMENTS` είναι οι παράμετροι που θα παίρνει το script, για παράδειγμα τα πειραματικά δεδομένα που θα εισάγει ο χρήστης.

Μετά από προσπάθεια εφαρμογής του συνδυασμού αυτού, είδαμε πως δεν είναι εφικτό και χρήσιμο για τη δική μας περίπτωση. Πιο συγκεκριμένα, ο λόγος που δε χρησιμοποιήθηκε τελικά αυτό το σύνολο τεχνολογιών, είναι κυρίως επειδή υπήρξε δυσκολία στον τρόπο που ο χρήστης θα μπορεί να εισάγει τα πειραματικά του δεδομένα. Η επιλογή για δεξί κλικ, όπως παρουσιάζεται στις απαιτήσεις, δεν ήταν εφικτή. Επιπρόσθετα, ο αριθμός των παραμέτρων για το script ήταν πολύ μεγάλος, λόγω του μεγάλου αριθμού κόμβων στο δίκτυο, πράγμα που έκανε την κλήση του πολύ αργή.

4.3 Σύνολο Τεχνολογιών 2 : GUI σε wxPython, NetworkX

4.3.1 Η βιβλιοθήκη wxPython [29]

Η βιβλιοθήκη wxPython είναι ένα σύνολο εργαλείων για δημιουργία γραφικών διεπαφών (GUI) για τη γλώσσα προγραμματισμού Python. Επιτρέπει τους προγραμματιστές να δημιουργήσουν προγράμματα με μια ισχυρή, εξαιρετικά λειτουργική γραφική διεπαφή χρήστη, απλά και εύκολα. Είναι υλοποιημένη ως επέκταση της Python (native code) που περικλείει τη δημοφιλή, ανεξάρτητης πλατφόρμας (cross-platform) βιβλιοθήκη για διεπαφές, wxWidgets, η οποία είναι γραμμένη σε C++.

Όπως η Python και η wxWidgets, η wxPython είναι Open Source, πράγμα που σημαίνει ότι είναι δωρεάν για όλους και ο κώδικάς της είναι διαθέσιμος για επεξεργασία. Οποιοσδήποτε μπορεί να συνεισφέρει με το να κάνει διορθώσεις ή βελτιώσεις σε αυτή.

Η wxPython είναι σύνολο εργαλείων ανεξάρτητης πλατφόρμας (cross-platform). Δηλαδή, ένα πρόγραμμα που τη χρησιμοποιεί μπορεί να τρέξει σε διάφορες πλατφόρμες, χωρίς οποιεσδήποτε μετατροπές. Μέχρι τώρα, υποστηριζόμενες πλατφόρμες είναι 32-bit Windows, τα περισσότερα Unix ή βασιζόμενα σε Unix συστήματα και Macintosh OS X. Τα προγράμματα που είναι γραμμένα με wxPython είναι απλά, εύκολα στη γραφή και εύκολα στην κατανόηση.

4.3.2 Συνδυασμός wxPython και NetworkX

Στο Σύνολο Τεχνολογιών 1 (κεφάλαιο 4.2), το κυρίως πρόβλημα βρίσκεται στη γραφική διεπαφή, γι' αυτό το λόγο θα γίνει καινούργια γραφική διεπαφή χρήστη χρησιμοποιώντας τη βιβλιοθήκη wxPython. Δε θα χρειαστεί να καλείται εξωτερικό script, αφού και τα δύο εργαλεία λειτουργούν μέσω της γλώσσας Python.

Ο συνδυασμός αυτός αποτελεί ένα πιθανό σενάριο υλοποίησης, όμως η έρευνα για τα εργαλεία απεικόνισης συνεχίστηκε μέχρι να καταλήξουμε στο πιο ικανοποιητικό για το χρήστη.

4.4 Σύνολο Τεχνολογιών 3 : GUI σε Tkinter(Python), Graphviz και Λ.Σ. Linux

4.4.1 Το εργαλείο Graphviz (Graph Vizualization Software) [10]

Το Graphviz είναι ένα Open Source λογισμικό οπτικοποίησης γράφων με σημαντικές εφαρμογές στα δίκτυα, στη βιοπληροφορική, στην τεχνολογία λογισμικού, στο σχεδιασμό ιστού και βάσεων δεδομένων και στην οπτικοποίηση για άλλους τεχνικούς τομείς. Δημιουργήθηκε από την AT&T Labs Research για σχεδιασμών γράφων, ορισμένων σε scripts γραμμένα στη γλώσσα DOT.

Η γλώσσα DOT [23] είναι μια απλή γλώσσα για περιγραφή γράφων. Είναι τόσο απλή που μπορεί να χρησιμοποιηθεί και από άτομα που δε σχετίζονται με προγραμματισμό. Τα γραφήματα σε DOT είναι συνήθως αρχεία που τελειώνουν σε .gv ή .dot. Όσον αφορά τη σύνταξή της είναι πάρα πολύ απλή. Για παράδειγμα, αν έχω δύο κόμβους, τον A και τον B, για να δείξω ότι συνδέονται με ακμή ο κώδικας θα είναι : A ->B;

Για να χρησιμοποιηθεί το εργαλείο Graphviz μέσω της Python, θα πρέπει να χρησιμοποιηθεί η διεπαφή του για Python, pyGraphviz.

4.4.2 Η βιβλιοθήκη Tkinter [14]

Η βιβλιοθήκη Tkinter προέρχεται από το σύνολο εργαλείων Tk για γραφικές διεπαφές χρήστη. Γράφτηκε από τον Fredrik Lundh. Είναι η τυπική διασύνδεση της Python με το σύνολο εργαλείων Tk (Tk toolkit) [27] και έρχεται μαζί με το πακέτο της Python. Δίνει τη δυνατότητα στον προγραμματιστή να αναπτύξει γρήγορα εφαρμογές με μια καλή γραφική διεπαφή χρήστη.

4.4.3 Συνδυασμός Graphviz και Tkinter σε Λ.Σ. Linux

Αυτό το σύνολο τεχνολογιών δοκιμάστηκε στο λειτουργικό σύστημα Linux Ubuntu 14.04.1 LTS, παρόλο που όλα τα άλλα σύνολα τεχνολογιών δοκιμάστηκαν σε λειτουργικό σύστημα Windows 7. Ο λόγος ήταν επειδή για το εργαλείο pyGraphviz δεν υπάρχει έκδοση για Windows.

Γι' αυτή την προσέγγιση, δημιουργήθηκε η γραφική διεπαφή με το Tkinter και εισάχθηκαν οι εικόνες για τα μονοπάτια του κυττάρου. Όμως, το εργαλείο Tkinter δεν παρέχει πολλές δυνατότητες όσον αφορά τη διεπαφή και πολλές από τις απαιτήσεις του χρήστη δε θα μπορούσαν να ικανοποιηθούν, όπως η απαίτηση για zoom και η απαίτηση για την αντιστοιχία μεταξύ των διαφόρων οθονών. Το κυριότερο πρόβλημα όμως αυτής της προσέγγισης, είναι ότι η εφαρμογή μας θα τρέχει μόνο σε συστήματα Linux, πράγμα που δεν είναι ευχάριστο για το χρήστη.

4.5 Σύνολο Τεχνολογιών 4 : GUI σε Java, Graphviz

Το σύνολο τεχνολογιών που θα περιγραφεί σε αυτό το υποκεφάλαιο, είναι το σύνολο που τελικά χρησιμοποιήθηκε για την ανάπτυξη του συστήματος, αφού κρίθηκε κατάλληλο για να ικανοποιεί τις απαιτήσεις του χρήστη.

4.5.1 Η Γλώσσα Προγραμματισμού Java [2]

Η Java είναι ίσως η πιο ευρέως διαδεδομένη γλώσσα προγραμματισμού στις μέρες μας. Αρχικά δημιουργήθηκε από τον James Gosling στην Sun Microsystems (η οποία συγχωνεύτηκε στην Oracle Corporation) και κυκλοφόρησε το 1995. Είναι μια αντικειμενοστραφής γλώσσα, βασισμένη σε κλάσεις, και ειδικά σχεδιασμένη για να δημιουργεί όσο το δυνατό λιγότερες εξαρτήσεις υλοποίησης.

Σκοπός της είναι να επιτρέπει στους προγραμματιστές εφαρμογών να γράφουν μια φορά τον κώδικά τους και αυτός να τρέχει παντού, ανεξαρτήτως πλατφόρμας ("write once, run anywhere" (WORA)). Οι εφαρμογές γραμμένες σε Java, μεταγλωττίζονται σε bytecode, ο οποίος μπορεί να τρέξει σε οποιαδήποτε Java Virtual Machine (JVM) [37], ανεξαρτήτως της αρχιτεκτονικής του υπολογιστή στην οποία τρέχει.

4.5.2 GUI σε Java : το λογισμικό NetBeans IDE [22]

Η Java είναι μια γλώσσα προγραμματισμού που υποστηρίζει ένα μεγάλο αριθμό βιβλιοθηκών. Δύο από αυτές είναι οι Swing και AWT [32] οι οποίες υποστηρίζουν σχεδιασμό και υλοποίηση γραφικών διεπαφών χρήστη.

Το λογισμικό NetBeans IDE , είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για εφαρμογές σε Java. Ο σχεδιασμός και η υλοποίηση γραφικών διεπαφών χρήστη με το NetBeans γίνονται πολύ εύκολα. Έχοντας ενσωματωμένες τις βιβλιοθήκες Swing και AWT, το NetBeans παρέχει μια παλέτα που περιέχει όλα τα συστατικά στοιχεία (components) των βιβλιοθηκών αυτών. Το κύριο συστατικό στοιχείο είναι το JFrame, που μπορεί να περιέχει άλλα συστατικά στοιχεία όπως τα JMenus, JButtons, JTextFields κ.α.

Χρησιμοποιώντας το Design View, με ένα απλό drag-and-drop από την παλέτα σε ένα καμβά, ο προγραμματιστής μπορεί να σχεδιάσει τη διεπαφή. Ο κώδικας για τα συστατικά στοιχεία που εισάγονται στον καμβά δημιουργείται αυτόματα και υπάρχει η δυνατότητα επεξεργασίας του μέσω του Source View. Επιπρόσθετα, μπορούν να προστεθούν event handlers σε κάθε συστατικό στοιχείο, όπως είναι το Mouse Clicked, Action Performed κ.α.

4.5.3 Graphviz Java API [13]

Το Graphviz Java API είναι μια κλάση/βιβλιοθήκη (Graphviz.java) γραμμένη σε Java, με την οποία μπορούμε να καλέσουμε προγράμματα σε γλώσσα DOT [23], μέσω προγραμμάτων στη Java. Δημιουργήθηκε για να διευκολύνει τη χρήση του εργαλείου Graphviz, που αναλύθηκε σε προηγούμενο υποκεφάλαιο, μέσα σε προγράμματα της Java. Δημιουργήθηκε το 2003 από τον Laszlo Szathmary, με ψευδώνυμο Jabba Laci, ο οποίος έκανε τροποποιήσεις σε αυτό μέχρι το 2011. Η κλάση βρίσκεται ολοκληρωμένη, ως έχει, στο Παράρτημα Β.

4.5.4 Συνδυασμός GUI σε Java και Graphviz

Μέσω του NetBeans, δημιουργήθηκε μια πολύ εύχρηστη γραφική διεπαφή, στον κώδικα της οποίας συμπεριλήφθηκε η βιβλιοθήκη Graphviz.java για την παραγωγή των ανάλογων απεικονίσεων κυτταρικών δικτύων. Οι απεικονίσεις αυτές, ενσωματώνονται στη διεπαφή με τη μορφή .png εικόνων. Στη συνέχεια, εισάγονται τα πειραματικά δεδομένα χρησιμοποιώντας jPopupMenu και για κάθε αλλαγή, παράγεται και εμφανίζεται μια νέα .png εικόνα περιέχοντας την ανάλογη αλλαγή.

Μ' αυτό το συνδυασμό εργαλείων μπορούν να καλυφθούν όλες οι απαιτήσεις του χρήστη σχετικά με τη γραφική διεπαφή, καθώς και σχετικά με την απεικόνιση των δύο κυτταρικών δικτύων. Γι' αυτό, το λόγο, μετά από αναθεώρηση όλων των συνόλων τεχνολογιών, καταλήξαμε στο συμπέρασμα ότι αυτό το σύνολο τεχνολογιών θα είναι το καταλληλότερο για την περίπτωσή μας.

Κεφάλαιο 5

Το Σύστημα BioCT (Biological Cell-Processing Tool)

5.1 Εισαγωγή	28
5.2 Μοντέλο Κύκλου Ζωής Λογισμικού	28
5.3 Σχεδιασμός Συστήματος	33
5.4 Υλοποίηση Συστήματος	35

5.1 Εισαγωγή

Το σύστημα BioCT (Biological Cell-Processing Tool), αποτελεί το τελικό σύστημα που δημιουργήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας. Είναι ένα σύστημα το οποίο τηρεί ένα μεγάλο μέρος των απαιτήσεων και προδιαγραφών του χρήστη. Σχεδιάστηκε και υλοποιήθηκε για να πετύχει τον στόχο της διπλωματικής εργασίας (Κεφάλαιο 1.3). Στα υποκεφάλαια που θα ακολουθήσουν, αρχικά θα γίνει μια αναφορά και περιγραφή του μοντέλου κύκλου ζωής λογισμικού που ακολουθήσαμε για τη δημιουργία του τελικού συστήματος. Στη συνέχεια, θα αναφερθούν και θα περιγραφούν αναλυτικά όλες οι λεπτομέρειες που αφορούν το σχεδιασμό και την υλοποίηση του συστήματος. Επιπρόσθετα, θα γίνει εκτενής αναφορά στις λειτουργίες που υλοποιήθηκαν, καθώς και σε αυτές που δεν υλοποιήθηκαν και στους λόγους που η υλοποίησή τους δεν ήταν τελικά εφικτή.

5.2 Μοντέλο Κύκλου Ζωής Λογισμικού

5.2.1 Κύκλος Ζωής Λογισμικού

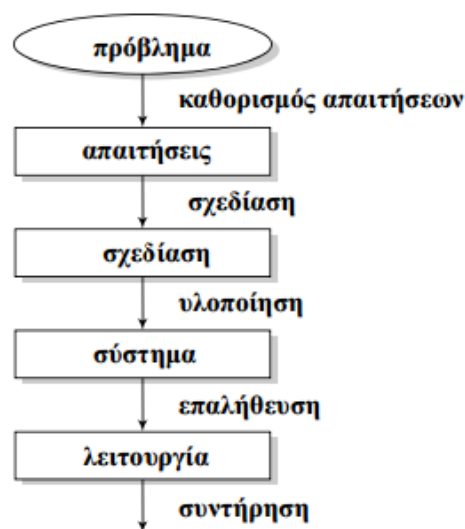
Ως Κύκλος Ζωής Λογισμικού, ορίζεται η διαδικασία ανάπτυξης ενός λογισμικού, συμπεριλαμβανομένων όλων των φάσεων ανάπτυξης που ακολουθούνται. Οι φάσεις αυτές, είναι τα στάδια από τα οποία διέρχεται μια εφαρμογή λογισμικού από τη στιγμή που ανατίθεται ως έργο. Συνεχίζουν κατά τη διάρκεια της δημιουργίας της, της χρήσης της και της συντήρησής της και τελειώνουν τη στιγμή της απόσυρσής της.

Ένα μοντέλο κύκλου ζωής λογισμικού περιλαμβάνει ποιες είναι αυτές οι φάσεις, ποια βήματα γίνονται κατά την εκτέλεση των διαφόρων εργασιών και τι προϊόντα παράγονται σε κάθε βήμα και πώς διαδέχεται η μια φάση την άλλη.

5.2.2 Μοντέλα Κύκλου Ζωής Λογισμικού [40][41]

Σ' αυτό το υποκεφάλαιο, θα δούμε κάποια χαρακτηριστικά μοντέλα κύκλου ζωής λογισμικού που μελετήθηκαν, μέχρι να καταλήξουμε στο τελικό μοντέλο που χρησιμοποιήσαμε.

5.2.2.1 Απλό, σειριακό μοντέλο

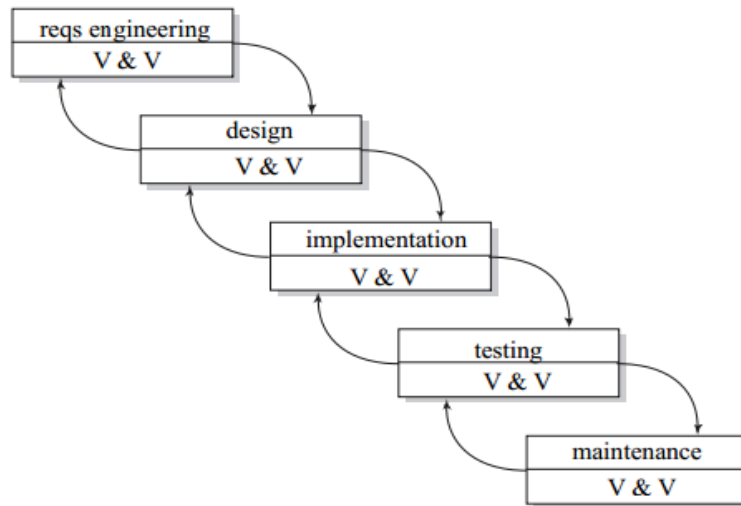


Εικόνα 7 [40] : Σχεδιάγραμμα Απλού Σειριακού Μοντέλου

Για το μοντέλο αυτό, χρειάζεται να γίνει πολύ καλός προγραμματισμός από την αρχή, γιατί δε μπορούν να γίνουν οι οποιεσδήποτε αλλαγές κατά τη διάρκεια. Ο χρήστης εμπλέκεται μόνο κατά τη φάση της μηχανικής απαιτήσεων. Το μοντέλο θα είναι επιτυχές, μόνο εάν σε κάθε βήμα παραδίδεται η ανάλογη τεκμηρίωση.

Στην περίπτωση που μελετάμε, το μοντέλο αυτό δεν είναι ικανοποιητικό, γιατί χρειαζόμαστε την εμπλοκή του χρήστη σε περισσότερες από μία φάσεις, έτσι ώστε να είμαστε σίγουροι ότι θα είναι ικανοποιημένος με το τελικό σύστημα.

5.2.2.2 Μοντέλο Καταρράκτη

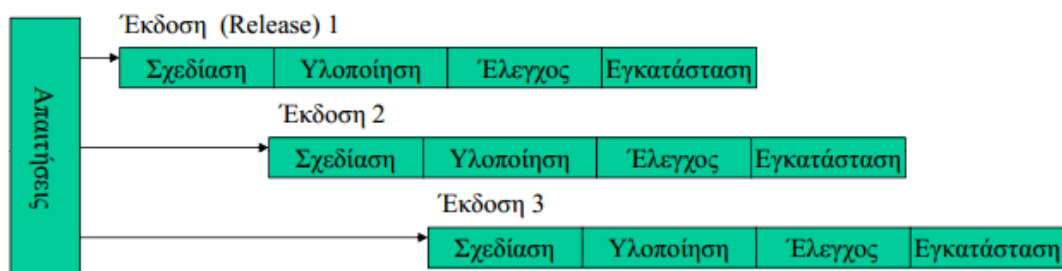


Εικόνα 8 [40] : Σχεδιάγραμμα Μοντέλου του Καταρράκτη

Το μοντέλο του καταρράκτη είναι επίσης σειριακό, όμως μπορούν να γίνουν αλλαγές ανάμεσα σε δύο βήματα, όπως φαίνεται και στο σχήμα. Βασικό του χαρακτηριστικό είναι η δημιουργία προδιαγραφών σε κάθε ένα βήμα, πράγμα που αποτελεί και το βασικό του προτέρημα. Επιπρόσθετα, η μορφή του μοντέλου διευκολύνει τη συντήρηση.

Παρ' όλα αυτά, το μοντέλο αυτό δεν μπορεί να είναι χρήσιμο για την ανάπτυξη του δικού μας συστήματος γιατί και πάλι ο χρήστης συμμετέχει μόνο στην αρχή και δε βλέπει το λογισμικό πριν το τέλος της υλοποίησής του. Ακόμη, το μοντέλο του καταρράκτη ενδείκνυται για μεγαλύτερα έργα λογισμικού από ότι το δικό μας.

5.2.2.3 Αυξητικό Μοντέλο

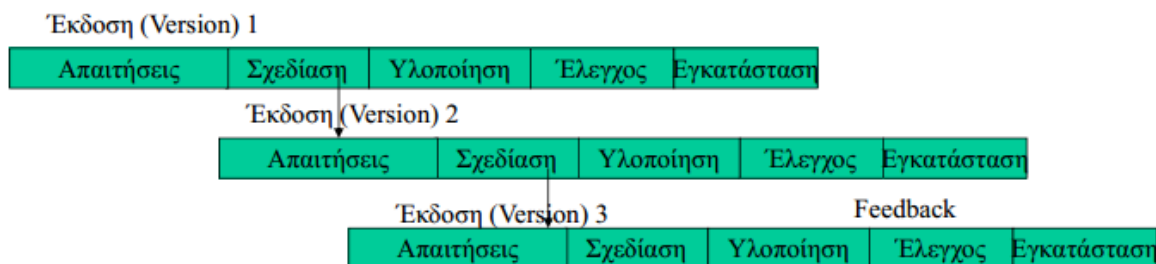


Εικόνα 9 [40] : Σχεδιάγραμμα Αυξητικού Μοντέλου

Σ' αυτό το μοντέλο, κύριο χαρακτηριστικό είναι η δημιουργία διαφορετικών εκδόσεων, οι οποίες βασίζονται στο αρχικό σύνολο απαιτήσεων που πήραμε από το χρήστη. Ξεκινώντας μόνο με ένα μέρος των απαιτήσεων στην πρώτη έκδοση, σε κάθε επόμενη έκδοση προστίθενται και άλλες. Σε κάθε έκδοση έχουμε ένα σύστημα με ορθή λειτουργικότητα.

Το μοντέλο αυτό έχει ως μοναδικό μειονέκτημα ότι οι απαιτήσεις του χρήστη δε μπορούν να αλλάξουν. Όμως είναι δύσκολο να χρησιμοποιηθεί στην περίπτωση που μελετάμε, γιατί θέλουμε ο χρήστης να συμμετέχει σχεδόν σε κάθε στάδιο του κύκλου ζωής και όχι στο τέλος κάθε λειτουργικής έκδοσης.

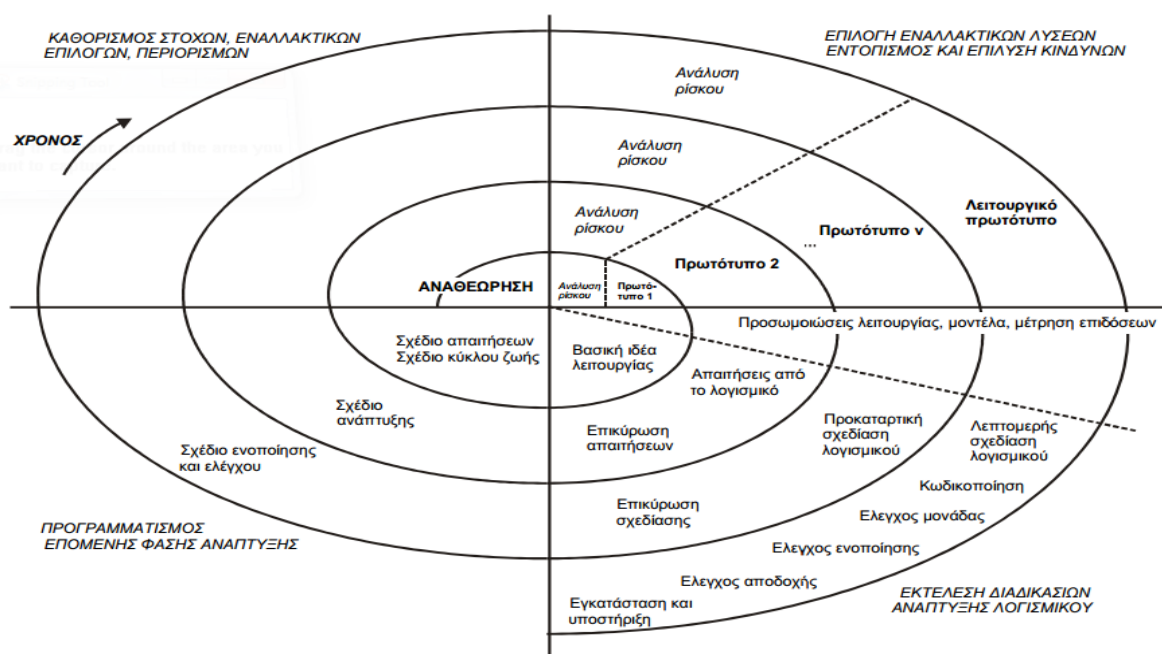
5.2.2.4 Εξελικτικό Μοντέλο



Εικόνα 10 [40] : Σχεδιάγραμμα Εξελικτικού Μοντέλου

Το εξελικτικό μοντέλο μοιάζει πολύ με το αυξητικό, με τη διαφορά ότι σε κάθε νέα έκδοση, ο χρήστης μπορεί να έχει νέες απαιτήσεις οι οποίες θα υλοποιηθούν στην επόμενη έκδοση. Γι' αυτό το λόγο, βασικό πλεονέκτημά του είναι η συνεχής συμμετοχή του χρήστη. Παρ' όλα αυτά όμως, χρησιμοποιώντας αυτό τον τρόπο ανάπτυξης, μπορεί να καταλήξουμε να ξεκινάμε συνέχεια από την αρχή, κάτι που θα μας έβγαζε εκτός χρονικού ορίου.

5.2.2.5 Σπειροειδές Μοντέλο (Spiral)

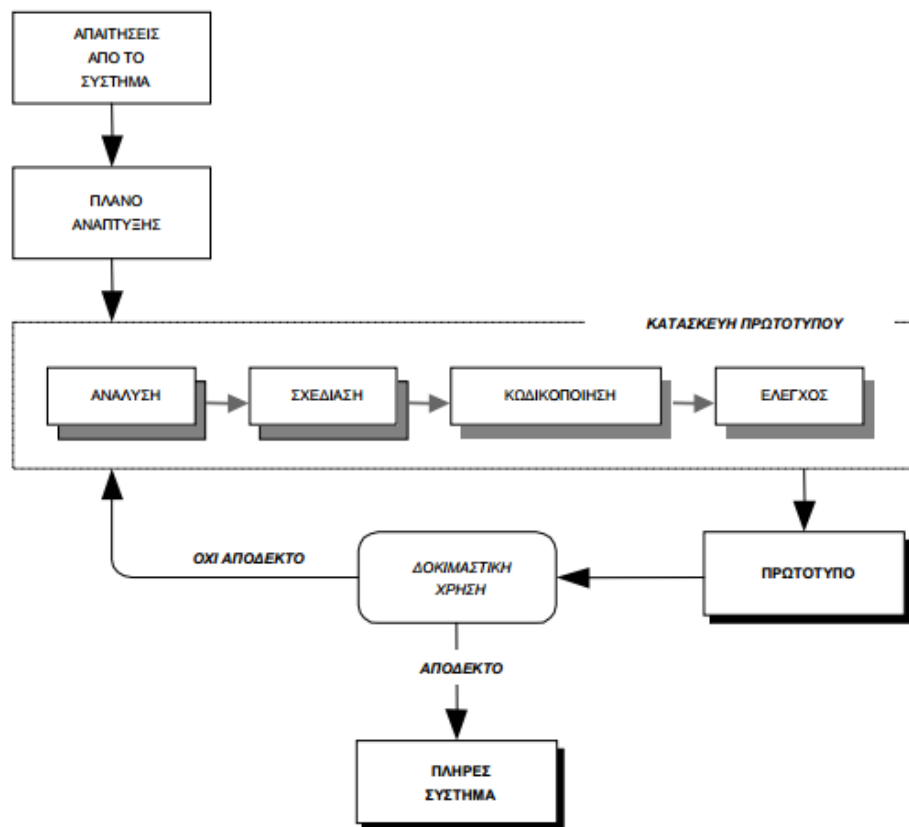


Εικόνα 11 [41] : Σχεδιάγραμμα Σπειροειδούς Μοντέλου

Το μοντέλο αυτό, είναι ένα είδος μοντέλου πρωτοτυποποίησης με συνεχή ανάλυση ρίσκου. Εάν σε οποιαδήποτε φάση η ανάλυση ρίσκου αποτύχει, τότε ολόκληρο το έργο θα διακοπεί, εκτός και αν το έργο έχει προχωρήσει πολύ.

Παρόλο που παρέχει δημιουργία πρωτοτύπων σε συνεχή βάση, το οποίο είναι πολύ βολικό για την περίπτωσή μας, το μοντέλο αυτό είναι πολύ περίπλοκο, γιατί το σύστημά μας είναι μικρό. Η φύση του το κάνει ακατάλληλο για μικρά έργα γι' αυτό και δε θα χρησιμοποιηθεί .

5.2.2.6 Απλό Μοντέλο Πρωτοτυποποίησης (Rapid Prototyping Model)



Εικόνα 12 [41] : Σχεδιάγραμμα Μοντέλου Πρωτοτυποποίησης

Σ' αυτό το μοντέλο, δημιουργείται πρωτότυπο του συστήματος σε λειτουργία, το οποίο δεν έχει την πλήρη λειτουργικότητα που αναμένεται να έχει το τελικό σύστημα. Βλέποντάς το, ο χρήστης θα μπορεί να πάρει μια ιδέα για το πώς θα είναι το τελικό σύστημα. Μ' αυτό τον τρόπο, τυχόν προβλήματα που μπορεί να έχει το σύστημα εντοπίζονται πιο εύκολα, οι ανάγκες του χρήστη ενσωματώνονται καλύτερα και η τελική υλοποίηση γίνεται με λιγότερο κόπο.

Μετά από την απόρριψη όλων των προηγούμενων μοντέλων κύκλου ζωής λογισμικού, καταλήξαμε στο συμπέρασμα ότι το μοντέλο πρωτοτυποποίησης είναι το καταλληλότερο

μοντέλο για την περίπτωση μας γιατί δίνει τη δυνατότητα στο χρήστη να συμμετέχει καθ' όλη τη διάρκεια της ανάπτυξης.

Επιπρόσθετα, η ανάπτυξη πρωτοτύπων γίνεται πολύ εύκολα και όταν έρθει η στιγμή ο χρήστης να δει το πρωτότυπο, μπορεί να δώσει τη γνώμη που χρειάζεται για να καθορίσει εάν το τελικό σύστημα θα βασιστεί στο πρωτότυπο. Αν όχι, ακολουθώντας την ίδια απλή διαδικασία, μπορεί να αναπτυχθεί καινούργιο πρωτότυπο, μέχρι να καταλήξουμε στο επιθυμητό για το χρήστη αποτέλεσμα.

5.3 Σχεδιασμός Συστήματος

Σύμφωνα με το μοντέλο πρωτοτυποποίησης που χρησιμοποιήσαμε, οι πρώτες δύο φάσεις είναι η φάση απαιτήσεων και η φάση πλάνου ανάπτυξης, οι οποίες εκτελέστηκαν και περιγράφηκαν στα κεφάλαια 3 και 4. Σ' αυτό το σημείο, πρέπει να γίνει η ανάλυση και σχεδίαση του συστήματος, η οποία θα γίνει μέσω σχεδιαγραμμάτων UML.

5.3.1 Η γλώσσα UML (Unified Modeling Language) [30]

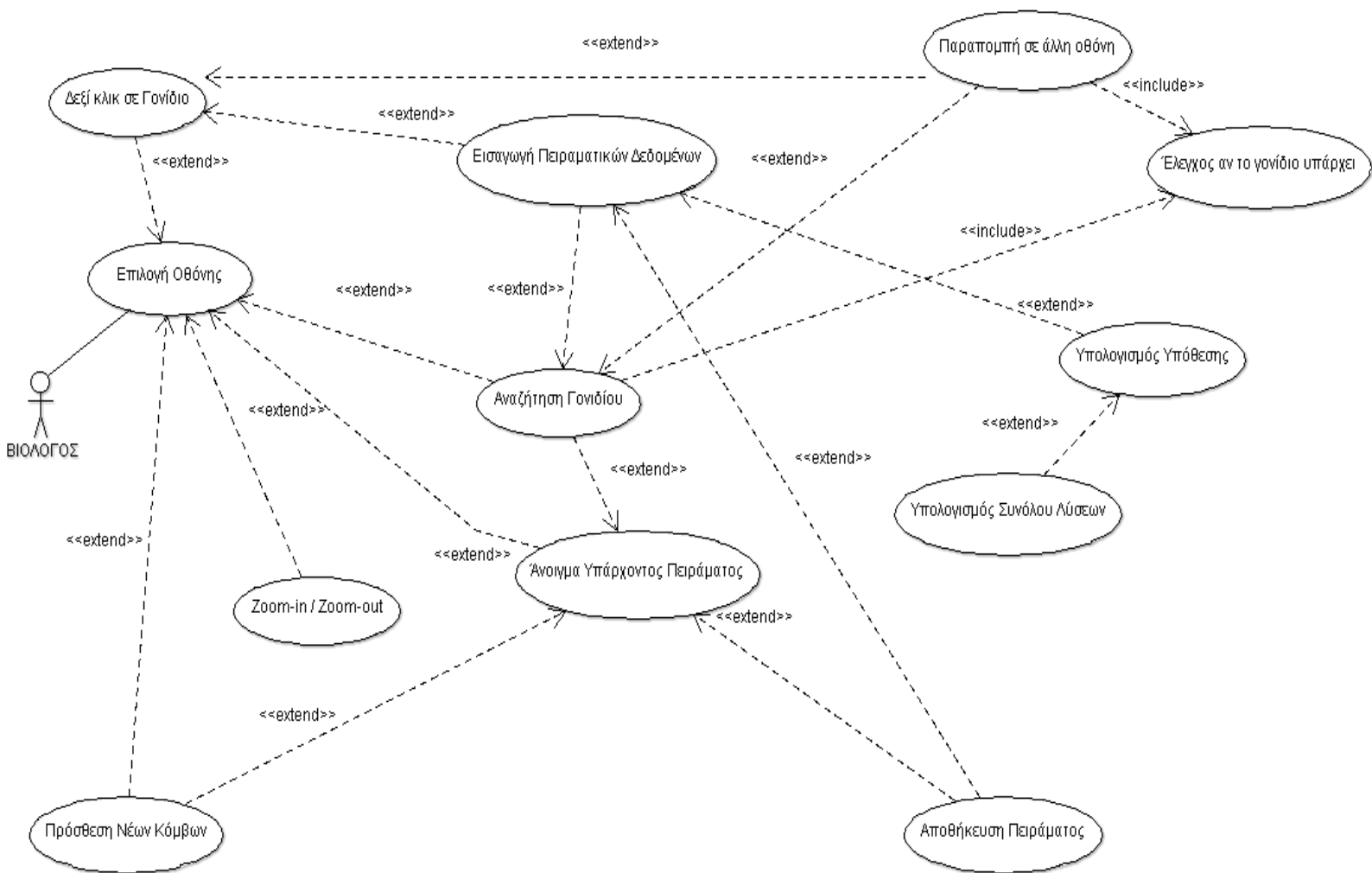
Η γλώσσα UML είναι μια γλώσσα μοντελοποίησης γενικού σκοπού, που χρησιμοποιείται ευρέως στον τομέα της τεχνολογίας λογισμικού. Σχεδιάστηκε για να παρέχει ένα τυποποιημένο τρόπο για την απεικόνιση του σχεδιασμού ενός συστήματος. Δημιουργήθηκε από τους Grady Booch, Ivar Jacobson και James Rumbaugh στην εταιρεία Rational Software το 1994-96. Σήμερα, η γλώσσα ανήκει στην εταιρεία Object Management Group (OMG) η οποία την υιοθέτησε από το 1997.

Υπάρχουν διάφορα είδη διαγραμμάτων UML. Κάποια από αυτά είναι τα διαγράμματα κλάσεων (class diagrams), διαγράμματα συστατικών (component diagrams), διαγράμματα χρήσης (use case diagrams), ακολουθιακά διαγράμματα (sequence diagrams) και διαγράμματα δραστηριοτήτων (activity diagrams).

5.3.2 Σχεδιαγράμματα UML

Για το σχεδιασμό του συστήματος BioCT , κατασκευάσαμε ένα σχεδιάγραμμα χρήσης UML, στο οποίο φαίνονται οι λειτουργίες του.

5.3.2.1 Διάγραμμα Χρήσης (Use Case Diagram)



Εικόνα 13 : UML Διάγραμμα Χρήσης (Use Case) του Συστήματος με τη χρήση του εργαλείου ArgoUML [28]

Στο διάγραμμα χρήσης που βλέπουμε στην εικόνα, ο ρόλος μας (actor) είναι ο Βιολόγος ο οποίος θα χρησιμοποιεί το σύστημα. Μπορούμε να δούμε ότι, η πρώτη επιλογή που κάνει ο χρήστης μας είναι η επιλογή της οθόνης στην οποία θα επεξεργαστεί το ανάλογο κυτταρικό δίκτυο. Μετά από αυτή την περίπτωση χρήσης (δηλαδή την επιλογή της οθόνης), συνδέονται διάφορες σχέσεις «extend», το οποίο συμβολίζει ότι μία περίπτωση χρήσης μπορεί να έχει διαφορετικά σενάρια. Μ' αυτό τον τρόπο, βλέπουμε τα διαφορετικά σενάρια που μπορεί να ακολουθήσει ο χρήστης μας, μετά από την επιλογή της οθόνης που θα κάνει. Επιπρόσθετα, βλέπουμε και μια σχέση «include», η οποία υποδεικνύει ότι δύο ή περισσότερες περιπτώσεις χρήσης έχουν την ίδια συμπεριφορά. Στο σύστημά μας, τόσο στην περίπτωση της αναζήτησης, τόσο και στην περίπτωση της παραπομπής σε άλλη οθόνη, πρέπει να γίνει έλεγχος εάν το γονίδιο που έχουμε επιλεγμένο υπάρχει.

5.4 Υλοποίηση Συστήματος

Η υλοποίηση του συστήματος έγινε με μια σειρά εργασιών και με κύριο συστατικό, την κατασκευή πρωτοτύπων. Οι εργασίες ξεκίνησαν με τη δοκιμή των τεσσάρων συνόλων τεχνολογιών που αναλύσαμε στο Κεφάλαιο 4. Φτάνοντας στο τέταρτο σύνολο τεχνολογιών, δηλαδή το συνδυασμό GUI σε Java – Netbeans IDE [22] και Graphviz [10], καταλήξαμε στο συμπέρασμα ότι θα είναι το κατάλληλο για την υλοποίησή μας.

Στη συνέχεια, δημιουργήθηκε ένα πρωτότυπο του συστήματος με βάση αυτό το συνδυασμό, το οποίο είδε ο χρήστης μας και το ενέκρινε. Η υλοποίηση του συστήματος θα βασιζόταν σ' εκείνο το συνδυασμό κι έτσι, ξεκίνησε η ανάπτυξη των ολοκληρωμένων λειτουργιών του συστήματος.

Η υλοποίηση ξεκίνησε με τη δημιουργία των διαφόρων οθονών της γραφικής διεπαφής χρήστη. Στη συνέχεια, δημιουργήθηκαν τα διάφορα κουμπιά που χρειάζονταν, καθώς και τα βασικά μενού της κάθε οθόνης. Έπειτα, προστέθηκε η ανάλογη λειτουργικότητα σε αυτά και φτάσαμε στο σημείο της δημιουργίας και απεικόνισης των κυτταρικών δικτύων. Οι απεικονίσεις δημιουργήθηκαν με τη χρήση του εργαλείου Graphviz και προστέθηκαν ως εικόνες στις διάφορες οθόνες του συστήματος.

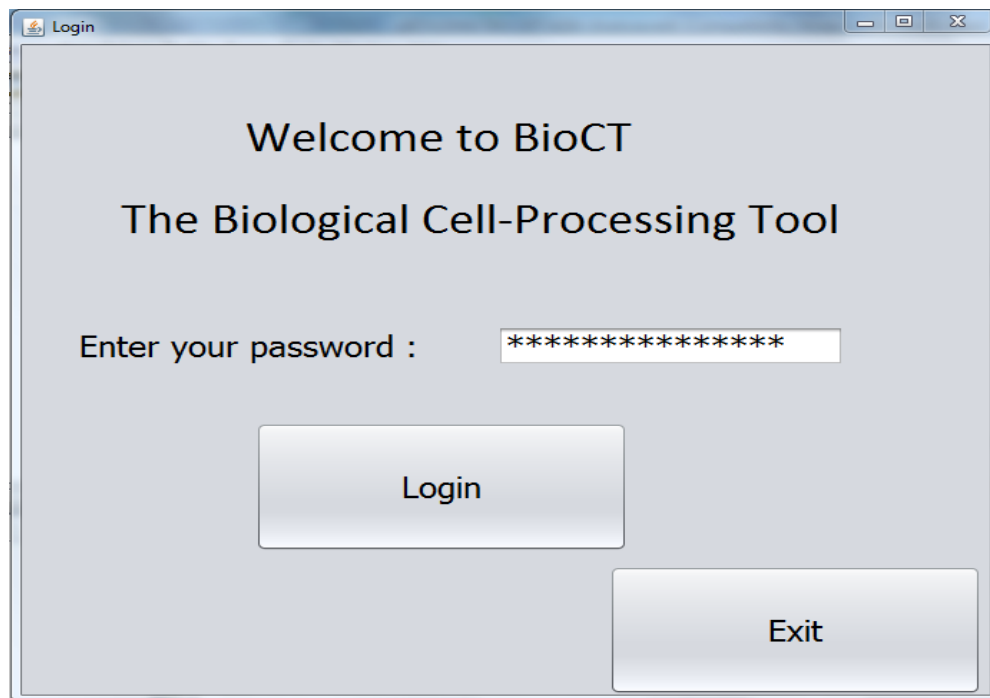
Σ' αυτό το σημείο, έγινε στατική εύρεση των συντεταγμένων κάθε κόμβου και για τον κάθε ένα δημιουργήθηκε ένα pop-up menu, τα οποία θα ενεργοποιούσε ο χρήστης με δεξιά κλικ στον αντίστοιχο κόμβο. Τα μενού αυτά, περιέχουν επιλογές για τις τέσσερις καταστάσεις που μπορεί να πάρει το γονίδιο, καθώς και αντιστοιχία του γονιδίου αυτού με τον εαυτό του σε άλλη οθόνη. Τέλος, υλοποιήθηκαν οι διάφορες λειτουργίες του συστήματος όσον αφορά τη γραφική διεπαφή, για το κάθε γονίδιο ξεχωριστά.

Στα υποκεφάλαια που θα ακολουθήσουν, θα περιγραφούν οι λειτουργίες που υλοποιήθηκαν, καθώς και εκείνες που δεν υλοποιήθηκαν και γιατί.

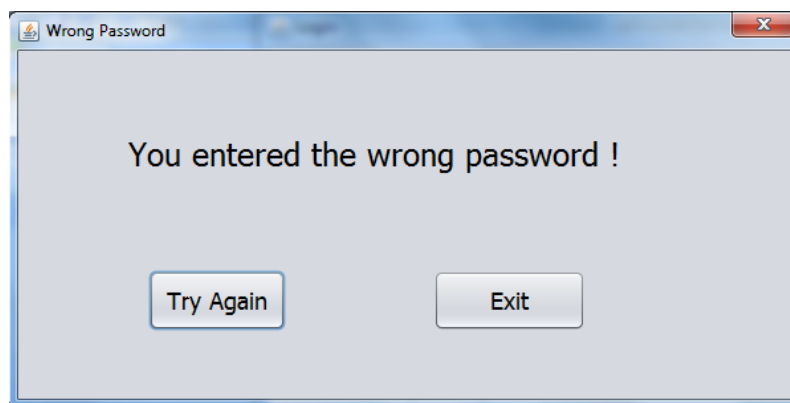
5.4.1 Λειτουργίες που υλοποιήθηκαν

5.4.1.1 Ασφαλής Είσοδος στο Σύστημα

Η πρώτη οθόνη που βλέπει ο χρήστης μας ξεκινώντας το σύστημα, είναι μια φόρμα στην οποία πρέπει να εισάγει τον κωδικό πρόσβασης που του δόθηκε από εμάς. Αν ο κωδικός είναι σωστός, γίνεται είσοδος στο σύστημα. Αν είναι λάθος, εμφανίζεται το ανάλογο μήνυμα λάθους.



Εικόνα 14 : Οθόνη Πρόσβασης στο Σύστημα

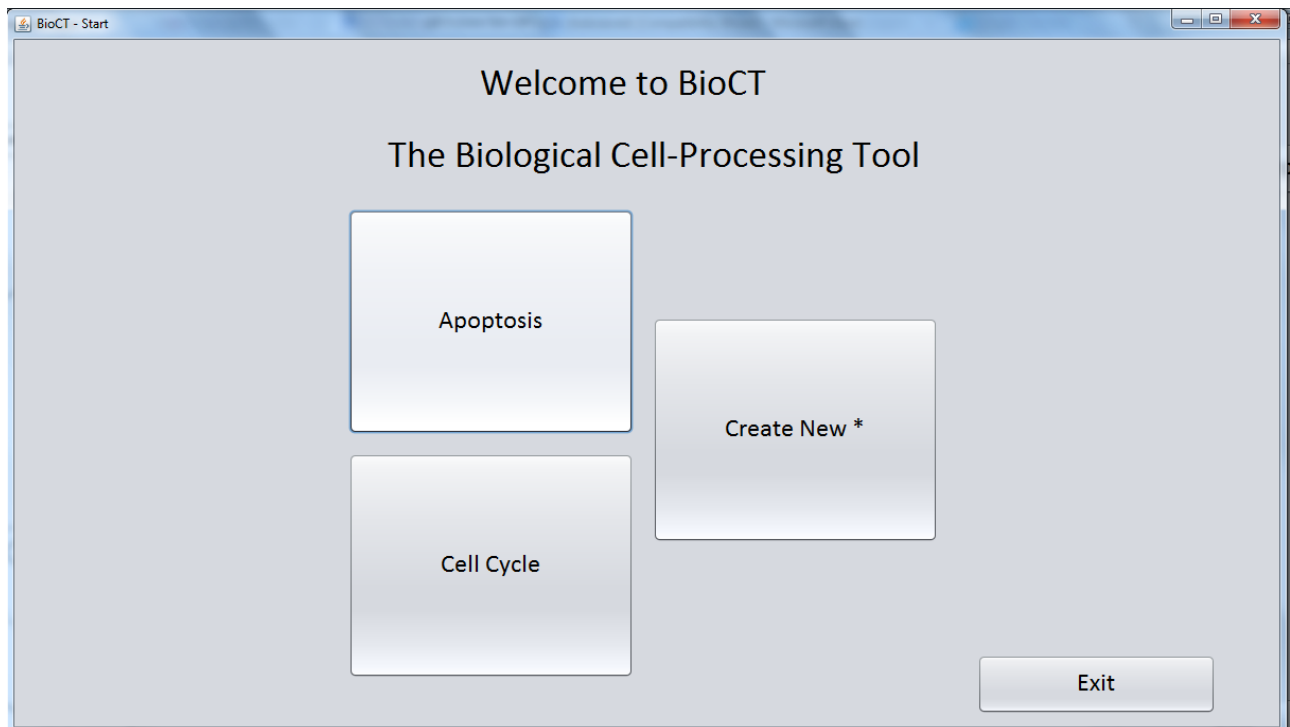


Εικόνα 15 : Μήνυμα λάθους εισαγωγής κωδικού πρόσβασης

5.4.1.2 Αρχική Οθόνη

Μετά την εισαγωγή του χρήστη στο σύστημα, εμφανίζεται η αρχική οθόνη του συστήματος με τις τέσσερις 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 την επιλογή για έξοδο.

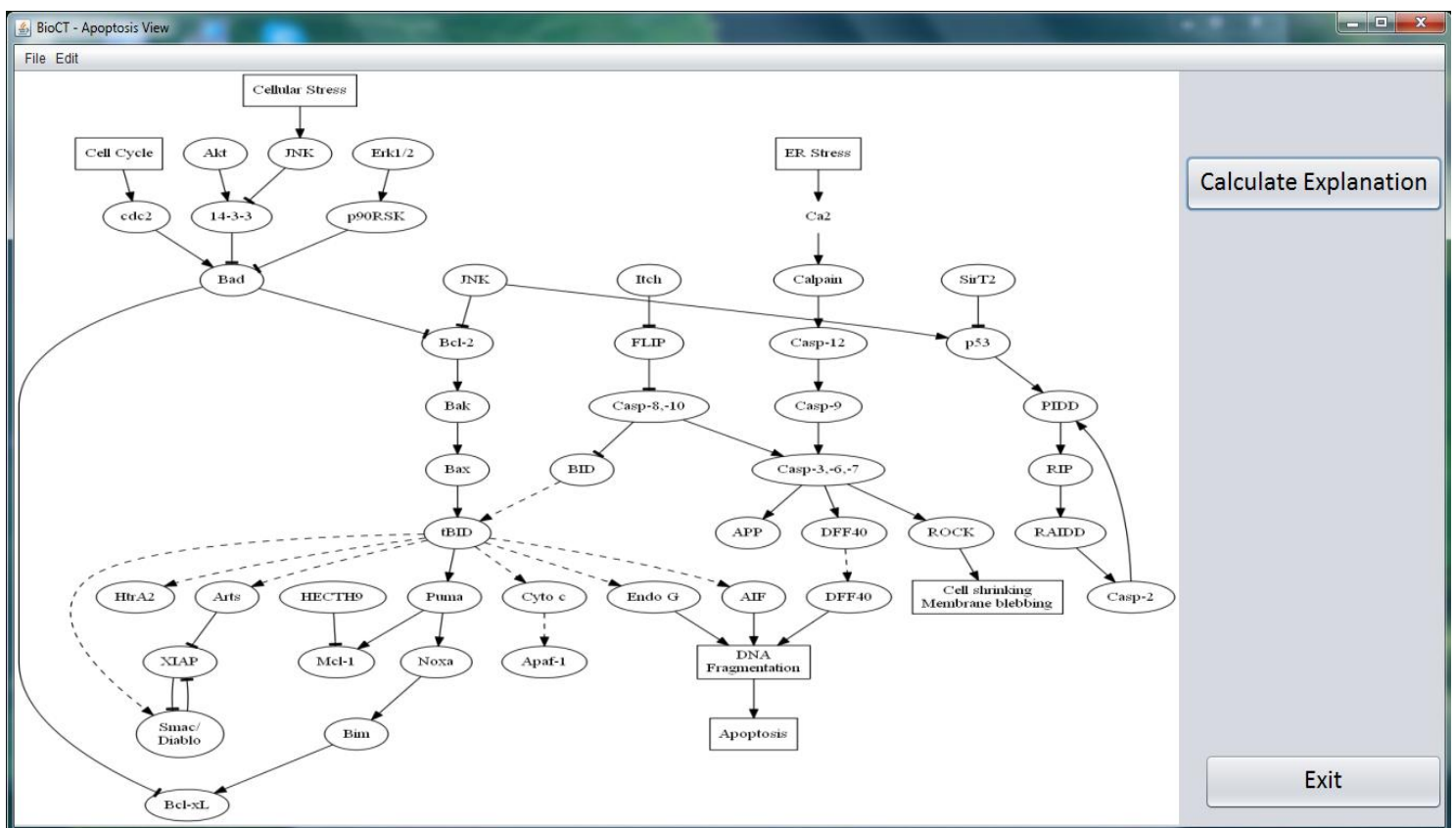
Από τις τέσσερις αυτές επιλογές, η μόνη που δεν υλοποιήθηκε είναι η επιλογή για δημιουργία νέου μονοπατιού, που θα περιγράψουμε σε επόμενο υποκεφάλαιο.



Εικόνα 16 : Αρχική Οθόνη Συστήματος

5.4.1.3 Οθόνη Απόπτωσης – Αρχική εικόνα

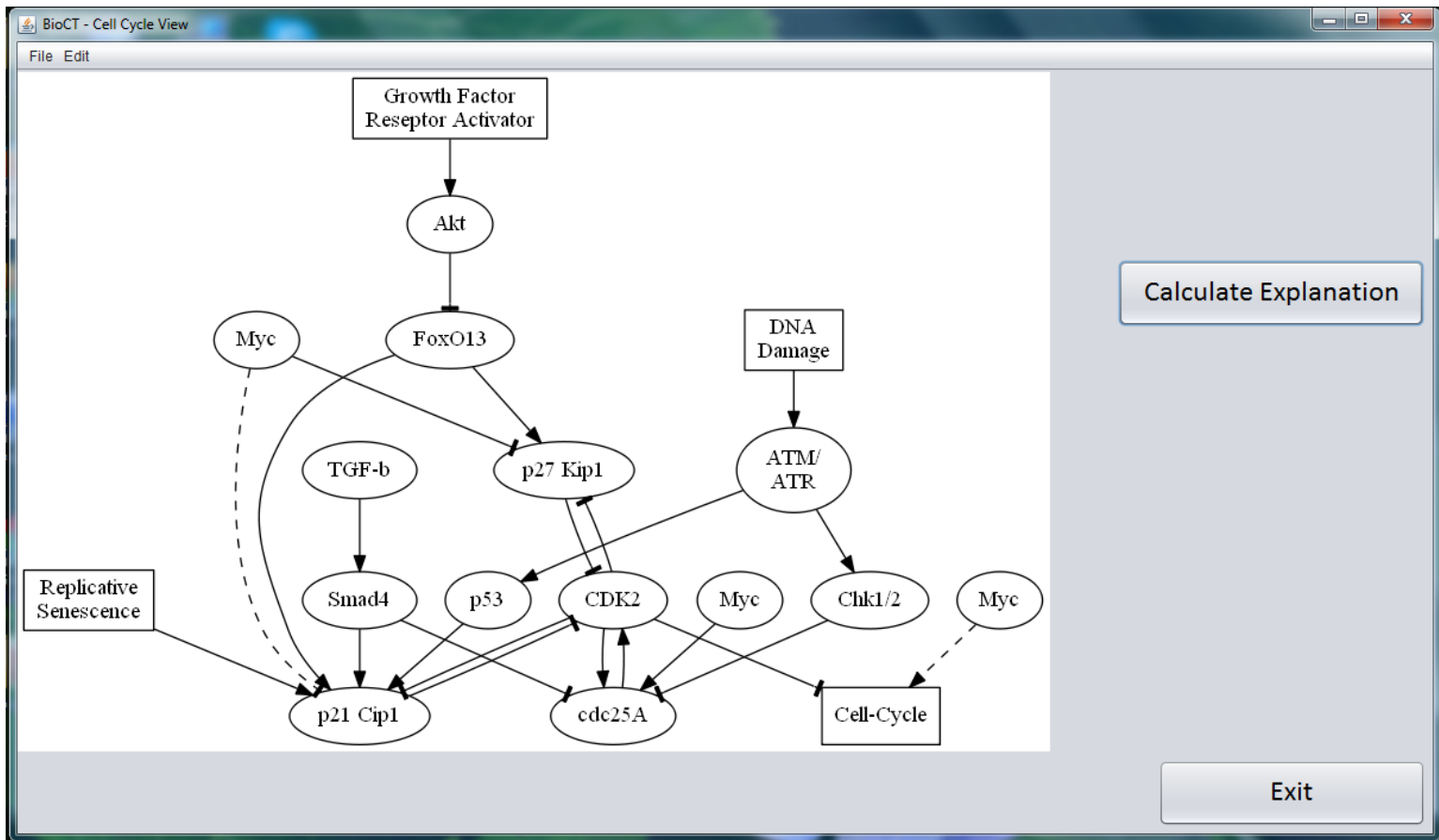
Στην οθόνη με το μονοπάτι της απόπτωσης, απεικονίζεται το μονοπάτι με ξεκάθαρο τρόπο. Η απεικόνιση των διεργασιών γίνεται με ορθογώνιο σχήμα, των πρωτεϊνών με οβάλ σχήμα και των άλλων ουσιών χωρίς περίγραμμα, όπως ακριβώς μας ζητήθηκε. Οι συνδέσεις μεταξύ τους ξεχωρίζουν αν είναι ενεργητικές κατευθυνόμενες ή ανασταλτικές κατευθυνόμενες. Επίσης, στο πάνω μέρος της οθόνης υπάρχει το βασικό μενού επιλογών, πάνω δεξιά βλέπουμε το κουμπί που όταν επιλεγεί θα γίνει ο υπολογισμός της επεξήγησης των καταστάσεων των γονιδίων και κάτω δεξιά το κουμπί για έξοδο από την οθόνη.



Εικόνα 17 : Οθόνη Απόπτωσης – Αρχική Εικόνα

5.4.1.4 Οθόνη Πολλαπλασιασμού – Αρχική εικόνα

Το μονοπάτι για τον πολλαπλασιασμό είναι φανερά μικρότερο από αυτό της απόπτωσης, γι' αυτό και βλέπουμε μια πιο ξεκάθαρη απεικόνιση σε αυτό. Οι απεικονίσεις γίνονται με τον ίδιο τρόπο όπως και στο μονοπάτι για την απόπτωση

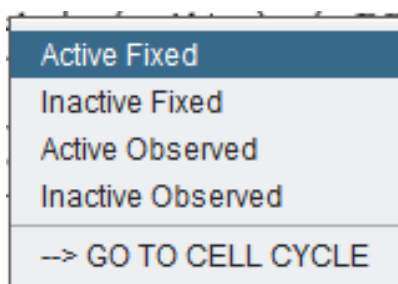


Εικόνα 18 : Οθόνη Πολλαπλασιασμού – Αρχική Εικόνα

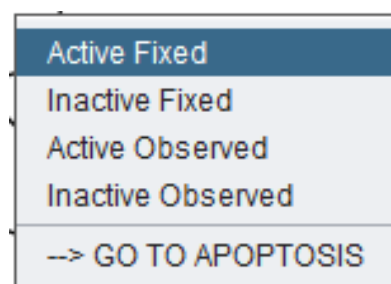
5.4.1.5 Εισαγωγή Πειραματικών δεδομένων και Αντιστοιχία Μεταξύ των διαφόρων Οθονών

Η εισαγωγή των πειραματικών δεδομένων γίνεται 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) ή inactive observed (απενεργοποιημένη μετά από το πείραμα).

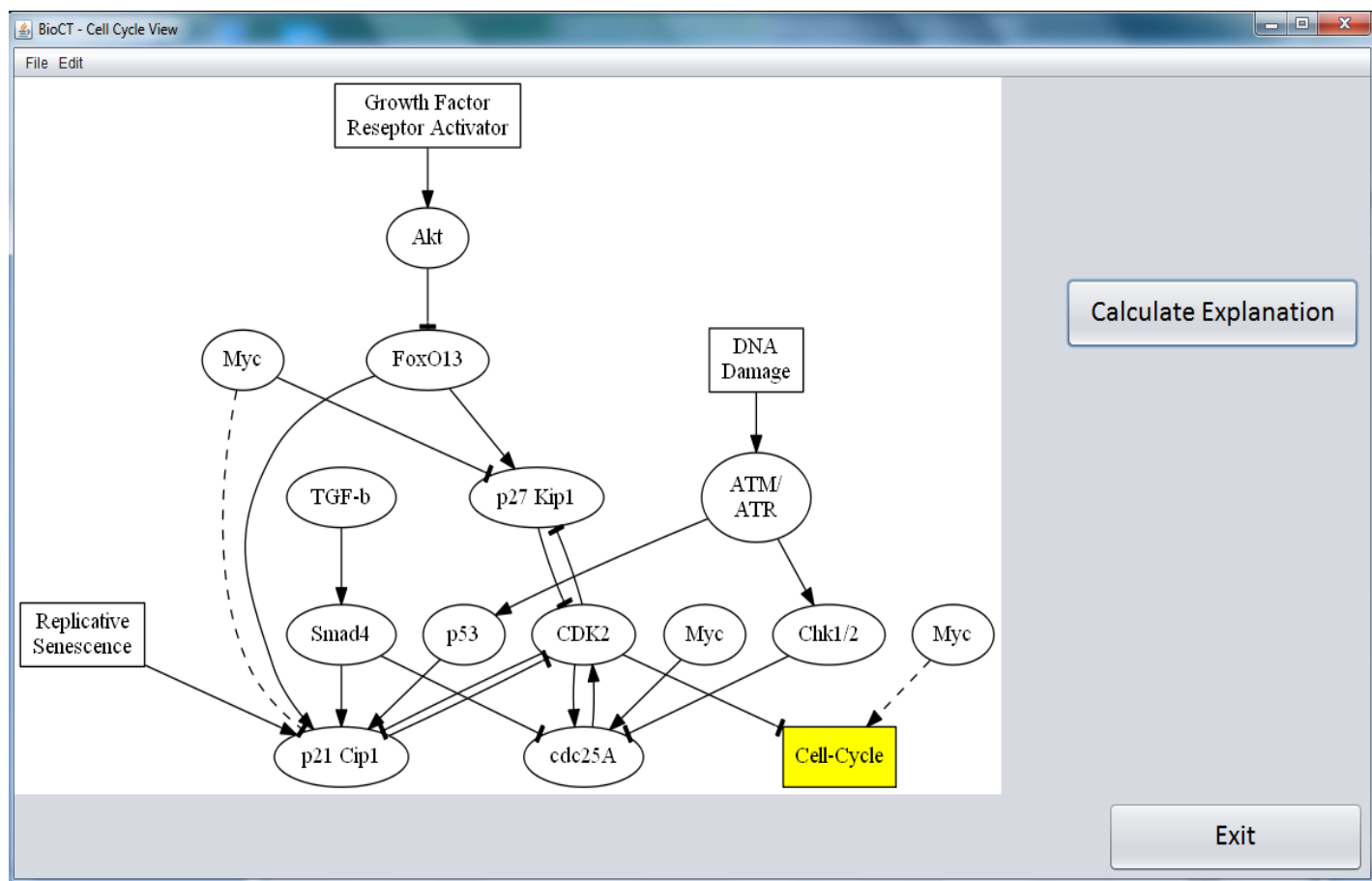
Επίσης, υπάρχει μια τέταρτη επιλογή που παραπέμπει στο γονίδιο στην αντίστοιχη οθόνη. Όταν γίνει η παραπομπή, το γονίδιο εμφανίζεται με διαφορετικό χρώμα (φωσφόρο κίτρινο) για να ξεχωρίζει. Αν το γονίδιο δεν υπάρχει στην αντίστοιχη οθόνη, εμφανίζεται το ανάλογο μήνυμα.



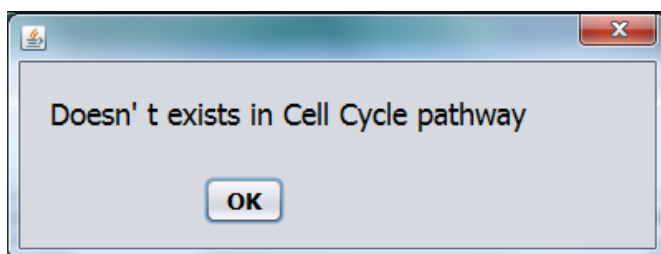
Εικόνα 19 : Μενού επιλογών για εισαγωγή πειραματικών δεδομένων στην οθόνη απόπτωσης



Εικόνα 20 : Μενού επιλογών για εισαγωγή πειραματικών δεδομένων στην οθόνη πολλαπλασιασμού



Εικόνα 21 : Οθόνη πολλαπλασιασμού, στην οποία έγινε παραπομπή από την οθόνη της απόπτωσης, μέσω του γονιδίου(διεργασίας) Cell-Cycle.



Εικόνα 22 : Μήνυμα που θα εμφανιστεί εάν στην οθόνη του πολλαπλασιασμού δεν υπάρχει το γονίδιο που επιλέξαμε για παραπομπή.

5.4.1.6 Οθόνες Απόπτωσης και Πολλαπλασιασμού Μετά την Εισαγωγή των Πειραματικών Δεδομένων

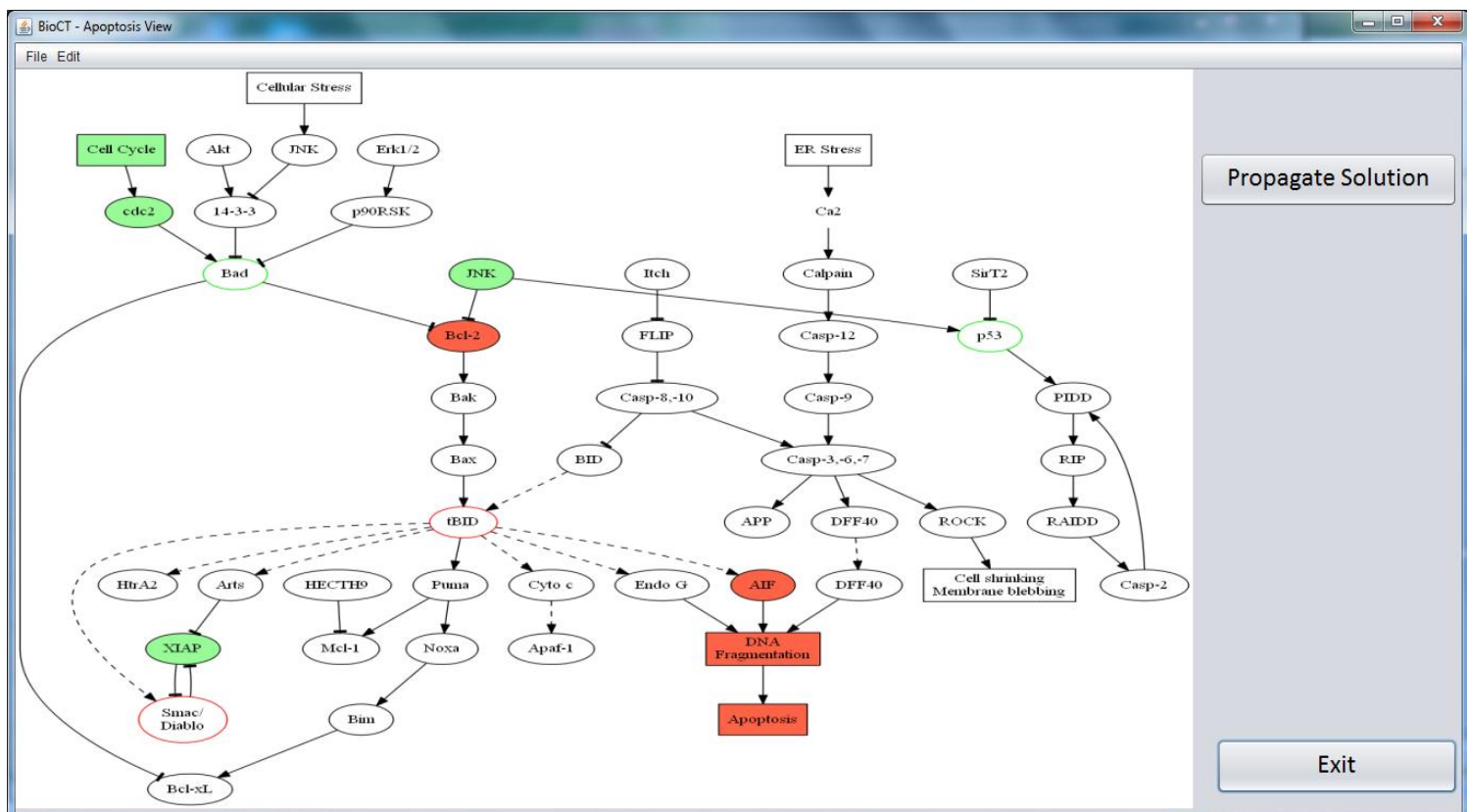
Με βάση τις απαιτήσεις του χρήστη, μετά την εισαγωγή των πειραματικών δεδομένων, τα περιγράμματα των κόμβων μέσα στο κυτταρικό δίκτυο αλλάζουν χρώματα ανάλογα με τις επιλογές του χρήστη. Αν το γονίδιο είναι active fixed, το περίγραμμα του γίνεται γαλάζιο, αν είναι inactive fixed το περίγραμμα του γίνεται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

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

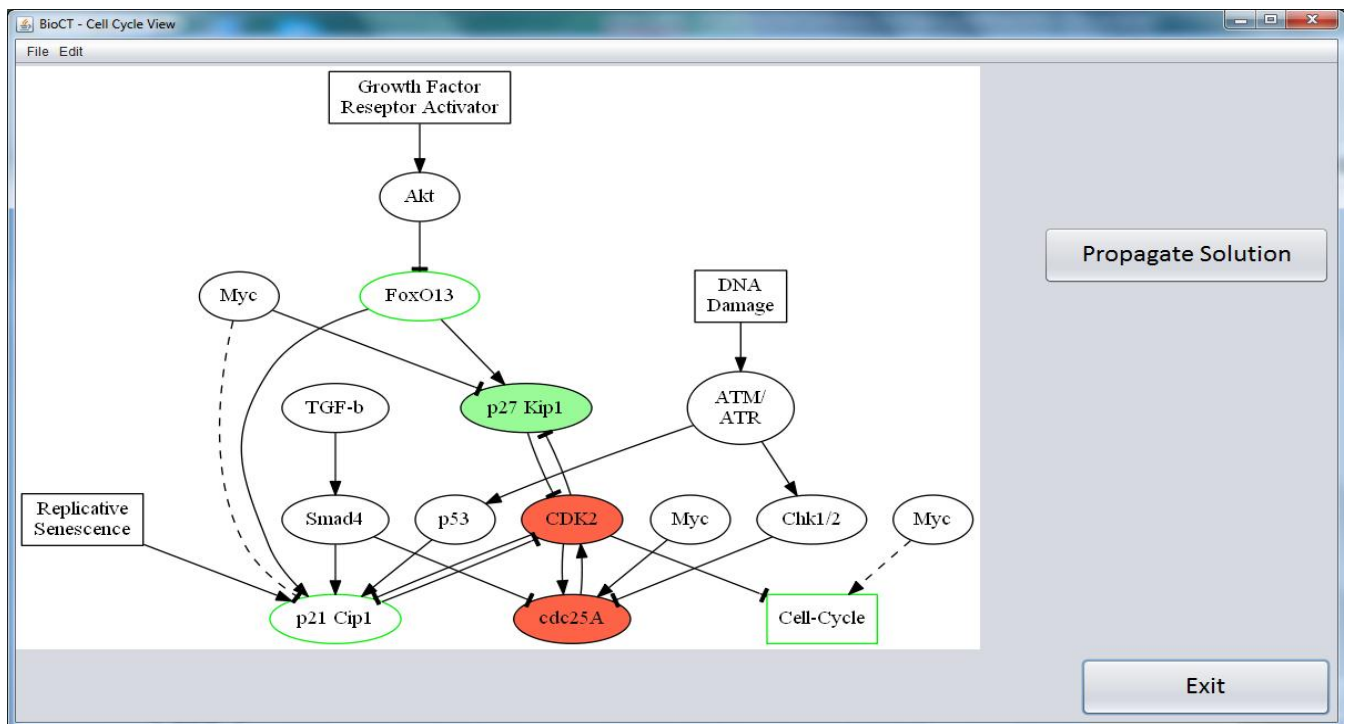
Σ' αυτό το σημείο είναι που θα γινόταν η εισαγωγή του Απαγωγικού Λογικού Προγραμματισμού. Όμως, επειδή αυτό το κομμάτι δε θα μας απασχολήσει στα πλαίσια αυτής της διπλωματικής εργασίας, με το πάτημα του κουμπιού εμφανίζουμε ένα ενδεικτικό τρόπο με τον οποίο θα φαίνεται η εμφάνιση της υπόθεσης όταν θα προστεθεί αυτή η λειτουργία.

Όσα γονίδια επεξηγούν τη μορφή που πήραν τα γονίδια βάσει των πειραματικών δεδομένων, εμφανίζονται με διαφορετικό χρώμα : γεμισμένα με πράσινο εάν ήταν ενεργοποιημένα και γεμισμένα με κόκκινο αν ήταν απενεργοποιημένα.

Επιπρόσθετα, στο σημείο που βρισκόταν το κουμπί Calculate Explanation, τώρα βρίσκεται το κουμπί Propagate Solution, με την επιλογή του οποίου ο χρήστης θα μπορεί να δει τη διάδοση του σήματος σε όλα τα πιθανά μονοπάτια για να δει όλα τα πιθανά σύνολα λύσεων.



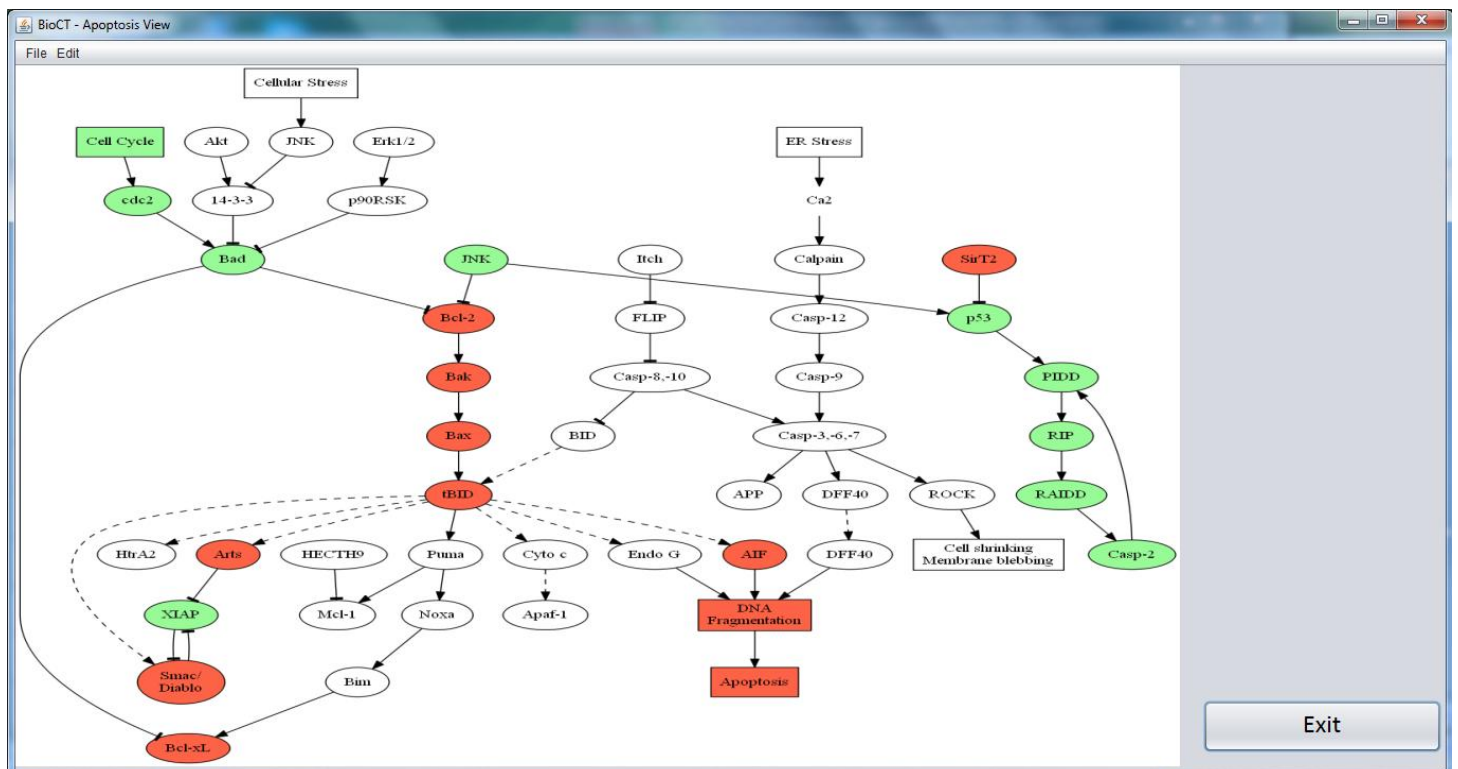
Εικόνα 25 : Οθόνη Απόπτωσης μετά από τον υπολογισμό της επεξήγησης



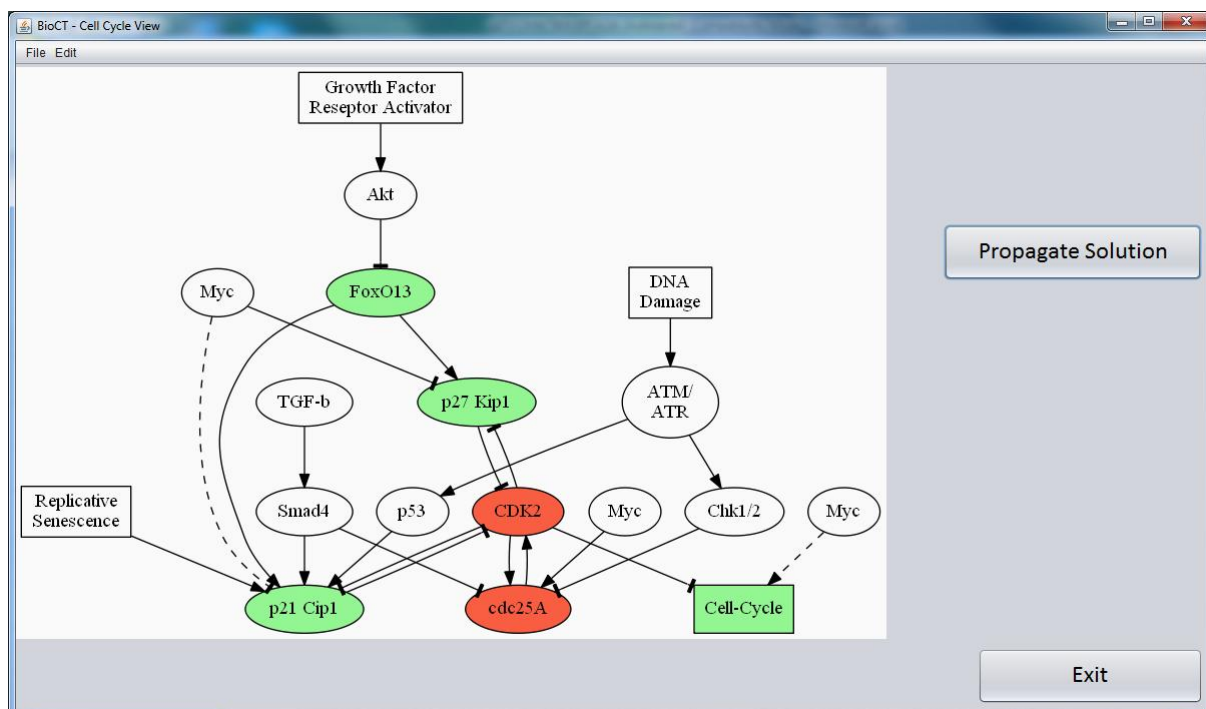
Εικόνα 26 : Οθόνη Πολλαπλασιασμού μετά από τον υπολογισμό της επεξήγησης

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

Σ' αυτό το σημείο, ο υπολογισμός όλων των πιθανών λύσεων μετά τη διάδοση του σήματος, θα έπρεπε επίσης να γινόταν με τη χρήση λογικού προγραμματισμού. Έτσι, αφού δεν υπάρχουν σ' αυτό το σημείο τα κατάλληλα εργαλεία υπολογισμού, και σε αυτή την περίπτωση δείχνουμε μόνο την απεικόνιση που θα έχει το σύστημα για ΜΙΑ πιθανή λύση.



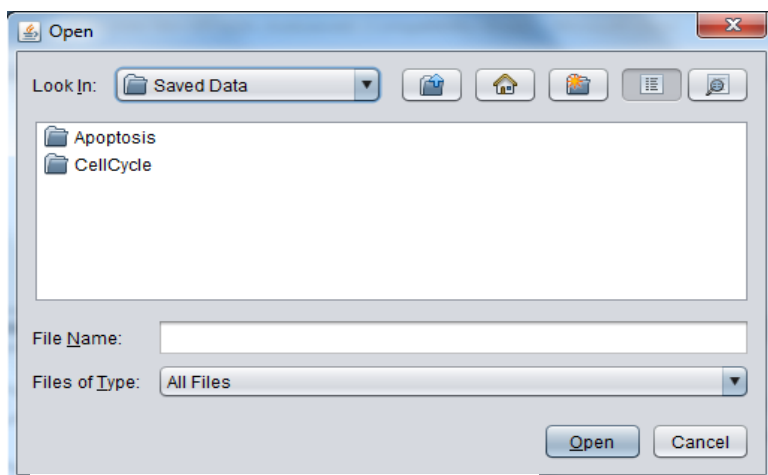
Εικόνα 27 : Οθόνη Απόπτωσης μετά από τη διάδοση του σήματος (διάδοση σήματος απενεργοποίησης : κόκκινο χρώμα, διάδοση σήματος ενεργοποίησης : πράσινο χρώμα)



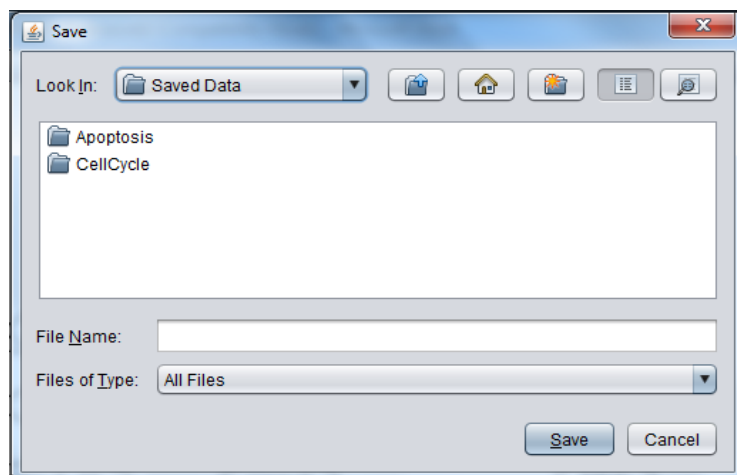
Εικόνα 28 : Οθόνη Πολλαπλασιασμού μετά από τη διάδοση του σήματος (διάδοση σήματος απενεργοποίησης : κόκκινο χρώμα, διάδοση σήματος ενεργοποίησης : πράσινο χρώμα)

5.4.1.9 Επιλογές Open και Save

Στο βασικό μενού επιλογών, κάτω από 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, με προεπιλεγμένο το φάκελο Saved Data που δημιουργήσαμε στο directory που βρίσκεται το σύστημά μας. Τα αρχεία που αναγνωρίζει το σύστημά μας είναι σε μορφή απλού κειμένου (text files) και με την εισαγωγή τους στο σύστημά μας γίνεται αυτόματη μετάφρασή τους σε .dot αρχεία, τα οποία αναγνωρίζει το εργαλείο Graphviz [10] για να μπορέσουν στη συνέχεια να μεταφραστούν στις ανάλογες εικόνες. Σημειώνουμε ότι ο χρήστης, μπορεί να φυλάξει το πείραμά του με οποιοδήποτε όνομα, φτάνει να έχει extension .txt. Αν δεν του δώσει extension, το σύστημα το προσθέτει από μόνο του αυτόματα.



Εικόνα 29 : Παράθυρο για άνοιγμα υπάρχοντος πειράματος από τον προεπιλεγμένο φάκελο Saved Data

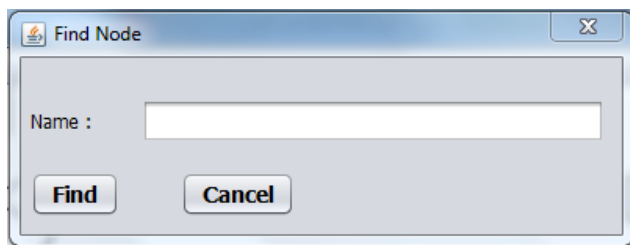


Εικόνα 30 : Παράθυρο για αποθήκευση πειράματος στον προεπιλεγμένο φάκελο Saved Data.

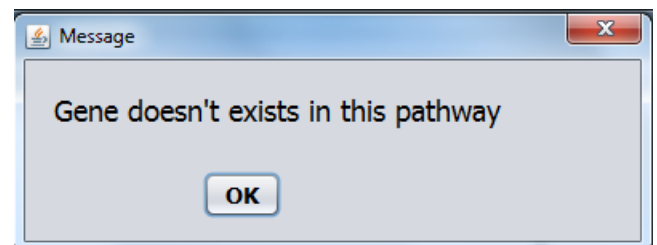
5.4.1.10 Επιλογή Αναζήτησης

Στο κυρίως μενού, κάτω από την επιλογή Edit υπάρχει η επιλογή Find, η οποία όταν επιλεγεί ανοίγει ένα παράθυρο στο χρήστη. Το παράθυρο αυτό, ζητά από το χρήστη το όνομα του γονιδίου που θέλει να αναζητήσει και πατώντας το κουμπί Find, αν υπάρχει το γονίδιο αυτό στο μονοπάτι θα εμφανιστεί με κίτρινο φωσφόρο χρώμα. Αν δεν υπάρχει στο μονοπάτι, θα εμφανιστεί το ανάλογο μήνυμα.

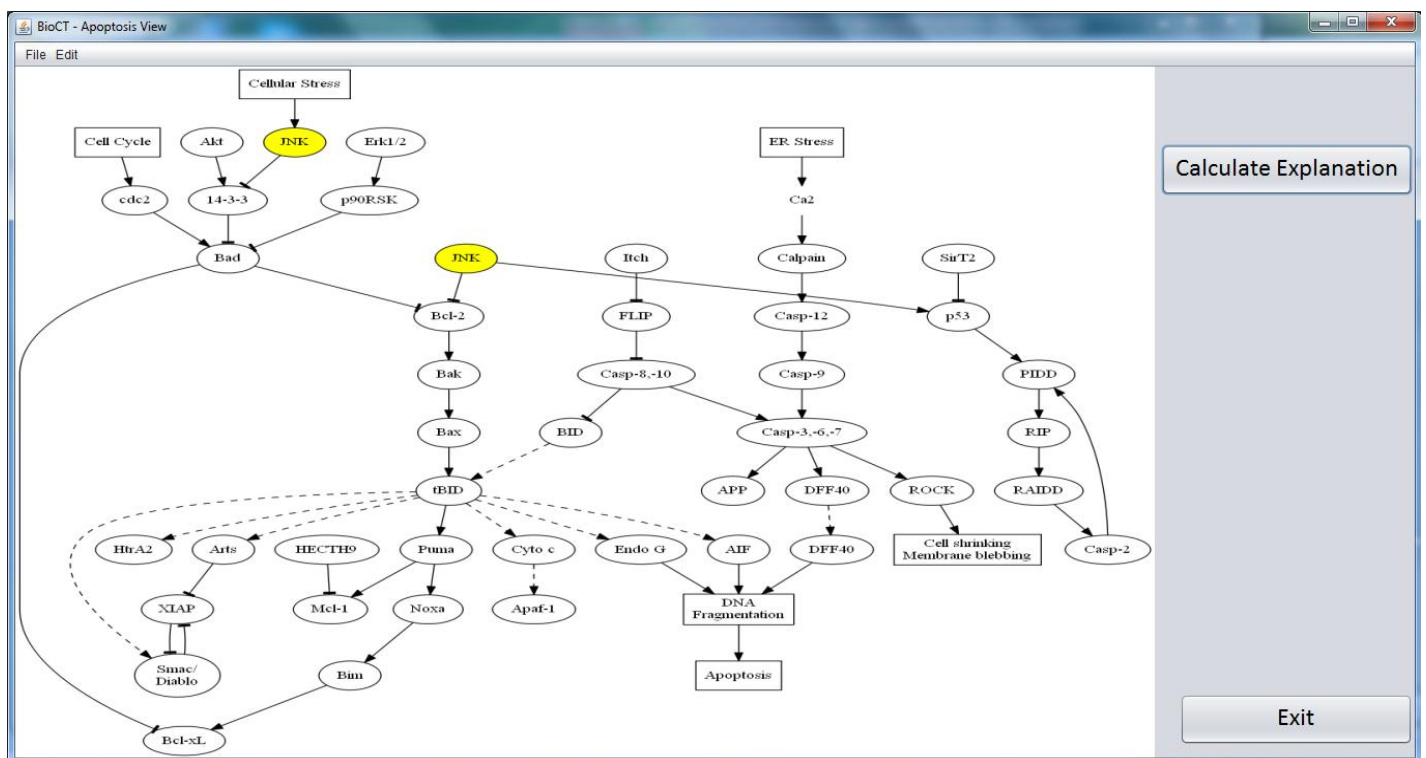
Σημειώνουμε ότι η αναζήτηση είναι case-insensitive, όμως ο χρήστης θα πρέπει να δώσει την ακριβή ονομασία του γονιδίου για να μπορεί αυτό να βρεθεί. Επίσης, ένα γονίδιο μπορεί να εμφανίζεται περισσότερες από μία φορές. Σ' αυτή την περίπτωση, το σύστημα το αναγνωρίζει όσες φορές και να εμφανίζεται.



Εικόνα 31 : Παράθυρο για αναζήτηση γονιδίου



Εικόνα 32 : Μήνυμα όταν το γονίδιο που αναζητήθηκε δεν ανήκει στο μονοπάτι



Εικόνα 33 : Οθόνη Απόπτωσης όταν αναζητήθηκε σε αυτή το γονίδιο με το όνομα : «JNK»

5.4.2 Λειτουργίες που δεν υλοποιήθηκαν

5.4.2.1 Zoom

Η λειτουργία του zoom (μεγέθυνσης και σμίκρυνσης), συμπεριλήφθηκε στις απαιτήσεις του χρήστη, λόγω του ότι στο προηγούμενο σύστημα (ApoCelSys [38]), οι συνδέσεις ανάμεσα στα διάφορα γονίδια δε φαίνονταν ξεκάθαρα. Έτσι, με την επιπλέον λειτουργία του zoom, θα μπορούσε να ξεκαθαρίσει η εικόνα, έτσι ώστε να είναι πιο καλό για το χρήστη.

Η προσέγγιση για τη λειτουργία του zoom, αρχικά ξεκίνησε όπως δηλώθηκε στις απαιτήσεις, δηλαδή με τις δύο επιλογές (+) για zoom-in και (-) για zoom-out. Στη συνέχεια, παρουσιάστηκαν κάποιες δυσκολίες λόγω του με την υλοποίησή του, άλλαζαν οι συντεταγμένες των κόμβων και δεν υπήρχε η δυνατότητα μετακίνησης (pan) πάνω στην εικόνα. Επίσης, χαλούσε η ανάλυση και τα ονόματα των γονιδίων δε φαίνονταν καθαρά. Γ' αυτό το λόγο, έπρεπε να αλλάξουμε την προσέγγιση αυτή.

Μέσα από μια συνομιλία με το χρήστη μας, καταλήξαμε σε μια δεύτερη προσέγγιση, η οποία ήταν να μπορεί να επιλέξει ένα κόμβο, και μεγεθύνοντας πάνω του να τον εμφανίζει μόνο του, μαζί με τους κοντινότερους γείτονές του (γείτονες πρώτου και δεύτερου επιπέδου μόνο).

Όταν καταλήξαμε στην επιλογή του εργαλείου Graphviz [10] και έγινε η τελική απεικόνιση του δικτύου, παρατηρήσαμε ότι η απεικόνιση του δικτύου φαινόταν ξεκάθαρα και στις δύο οθόνες (απόπτωσης και πολλαπλασιασμού). Έτσι, επικοινωνήσαμε ξανά με το χρήστη και βλέποντάς το συμφώνησε πως η λειτουργία του zoom δεν είναι πλέον αναγκαία, γι' αυτό το λόγο και δεν υλοποιήθηκε σε αυτό το στάδιο.

5.4.2.2 Οθόνη Δημιουργίας Νέου Δικτύου και Πρόσθεση Νέων Κόμβων σε Υπάρχον Δίκτυο

Το εργαλείο Graphviz [10] που χρησιμοποιήσαμε, έχει ένα δικό του, ενσωματωμένο τρόπο δυναμικής τοποθέτησης των κόμβων σε κάθε δίκτυο, έτσι ώστε να δημιουργείται η ξεκάθαρη εικόνα που αναφέραμε στο προηγούμενο υποκεφάλαιο. Μ' αυτό τον τρόπο, με κάθε εισαγωγή νέου κόμβου, αλλάζουν οι θέσεις όλων των υπαρχόντων κόμβων. Εμείς

χρειαζόμαστε τις συντεταγμένες του κάθε κόμβου για την αναγνώριση της ταυτότητας του κάθε γονιδίου το οποίο ο χρήστης επιλέγει να επεξεργαστεί.

Οι συντεταγμένες αυτές, βρίσκονται αποθηκευμένες σε ένα αρχείο τύπου .plain (απλού κειμένου). Όμως, αυτές οι συντεταγμένες που βρίσκονται στο αρχείο αυτό, διαφέρουν από τις real-time συντεταγμένες της οθόνης μας (του mouse), λόγω του scaling της εικόνας. Δηλαδή, αν η εικόνα μας έχει μέγεθος π.χ. 50x50 και το frame του συστήματος μας έχει μέγεθος 600x600, η συντεταγμένη (1,1) της εικόνας και η συντεταγμένη (1,1) του frame είναι διαφορετικές. Έτσι, έγινε μια εκτεταμένη και χρονοβόρα προσπάθεια για να βρεθεί μια μαθηματική σχέση ή ένας αλγόριθμος που να συνδέει τα δύο είδη συντεταγμένων.

Έγιναν αρκετές προσπάθειες για εξεύρεση αυτής τη σχέσης, όμως δεν ήταν επιτυχείς και ο λόγος ήταν ότι κάθε φορά που προσθέταμε ένα νέο κόμβο, το μέγεθος της εικόνας άλλαζε, κι έτσι δε μπορούσαμε να βρούμε την κατάλληλη σχέση. Έτσι, καταλήξαμε στο συμπέρασμα ότι θα η μόνη λύση θα ήταν να προχωρήσουμε με στατική εύρεση των συντεταγμένων, μόνο για τους υπάρχοντες κόμβους. Γι' αυτό το λόγο οι λειτουργίες της πρόσθεσης νέων κόμβων και της δημιουργίας νέου δικτύου, δεν ήταν εφικτές.

Κεφάλαιο 6

Έλεγχος – Αξιολόγηση Συστήματος

6.1 Εισαγωγή	48
6.2 Έλεγχος Συστήματος	48
6.3 Αξιολόγηση - Αποτελέσματα	49

6.1 Εισαγωγή

Ο έλεγχος συστήματος και οι αξιολόγηση του συστήματος είναι πάντα απαραίτητοι πριν την τελική έκδοσή του, γιατί πρέπει να δούμε εάν το σύστημα έχει την αναμενόμενη συμπεριφορά και να ανακαλυφθούν τυχόν σφάλματα που δε βρέθηκαν κατά τη διάρκεια της υλοποίησής του. Στο κεφάλαιο αυτό θα αναφερθούμε στις μεθόδους ελέγχου που χρησιμοποιήσαμε, στις μεθόδους αξιολόγησης, καθώς και στα αποτελέσματα που πήραμε μετά από την ολοκλήρωσή τους.

6.1 Έλεγχος Συστήματος

Για να είναι ένας έλεγχος πλήρης, χρειάζονται να γίνουν διάφορες δοκιμές. Το είδος δοκιμών που ακολουθήσαμε εμείς, ονομάζεται Coverage-Based Testing, που έχει ως σκοπό να δείξει σε μας αλλά και στο χρήστη ότι το σύστημα ικανοποιεί τις απαιτήσεις του. Αν η δοκιμή πετύχει, τότε αποδεικνύει ότι η λειτουργικότητα του συστήματος είναι όπως αναμενόταν.

Για να γίνουν οι έλεγχοι ορθότητας και ποιότητας του λογισμικού μας, δημιουργήθηκαν διάφορα τυπικά σενάρια χρήσης από εμάς, τα οποία προτού να πραγματοποιήσουμε την τελική συνάντησή μας με το χρήστη, τα ελέγξαμε ένα προς ένα, έτσι ώστε να είμαστε σίγουροι για την ορθή λειτουργικότητα του συστήματός μας και να ανακαλύψουμε τυχόν σφάλματα σε αυτό.

Στη συνέχεια, πραγματοποιήθηκε η τελική συνάντηση με τη χρήστη του συστήματος, στην οποία δώσαμε το τελικό σύστημα για να το δοκιμάσει. Οι δοκιμές που έκανε, έγιναν βάσει απαιτήσεων και βάσει των σεναρίων. Πιο συγκεκριμένα, εξετάσαμε μαζί αν κάθε απαίτηση που είχε για το σύστημα υλοποιήθηκε σωστά και στη συνέχεια, ελέγξαμε τα τυπικά σενάρια δοκιμών για να ελέγξουμε πλήρως την ορθότητά του. Ένα παράδειγμα τυπικού σεναρίου που δοκιμάσαμε είναι το ακόλουθο : «Ο χρήστης θέλει να ανοίξει ένα υπάρχον πείραμα στην οθόνη της απόπτωσης και να ασχοληθεί με το να εισάγει περισσότερα δεδομένα σε αυτό. Στη συνέχεια, θέλει να αποθηκεύσει το καινούργιο πείραμα, αλλά με διαφορετικό όνομα και να συνεχίσει με τη δημιουργία υποθέσεων. Παράλληλα, θα ήθελε να μπορεί να έχει ανοικτή και την οθόνη του πολλαπλασιασμού.»

Στο τέλος του ελέγχου συστήματος, έπρεπε να γίνει η ανάλογη αξιολόγηση του συστήματος βάσει των αποτελεσμάτων του ελέγχου.

6.2 Αξιολόγηση - Αποτελέσματα

Για την αξιολόγηση του συστήματος, μετά την ολοκλήρωση των κατάλληλων ελέγχων, κάναμε διάφορες ερωτήσεις στη χρήστη σχετικά με το πόσο ικανοποιημένη ήταν με αυτό.

Όταν η χρήστης μας ερωτήθηκε για το πόσο ικανοποιημένη ήταν με τη γενική εικόνα του συστήματος που δημιουργήσαμε, μας έδωσε την απάντηση πως ήταν πολύ ικανοποιημένη. Επίσης, ανέφερε ότι οι απεικονίσεις ήταν ξεκάθαρες και σίγουρα θα μπορούσαν να ερμηνευθούν πολύ εύκολα από οποιοδήποτε Βιολόγο. Επιπρόσθετα, βρήκε τη γραφική διεπαφή εύχρηστη και όπως αναμενόταν να είναι. Ως επιπλέον σχόλιο, επισήμανε ότι της άρεσε ιδιαίτερα ο τρόπος μετάβασης ανάμεσα στις δύο οθόνες.

Όσον αφορά τον έλεγχο ορθότητας του συστήματος, το αποτέλεσμα ήταν ότι το σύστημα λειτουργούσε ορθά σε όλες τις περιπτώσεις – σενάρια που το δοκιμάσαμε. Παρ' όλα αυτά όμως, ενώ το σύστημα είχε σε γενικές γραμμές σωστά απεικονισμένα τα δύο κυτταρικά δίκτυα, υπήρξαν μερικά λάθη σε βιολογικό επίπεδο. Όπως μας επισήμανε η χρήστης μας, κάποια γονίδια ενώ είναι πρωτεΐνες, ήταν απεικονισμένα ως διεργασίες και επίσης, κάποια άλλα είχαν λάθη στις ονομασίες τους. Το πρόβλημα αυτό ήταν το μοναδικό σε επίπεδο ποιότητας και λόγω της ευκολίας της φύσης του, διορθώθηκε αμέσως.

Κεφάλαιο 7

Συμπεράσματα – Μελλοντική Εργασία

7.1 Συμπεράσματα	50
7.2 Μελλοντική Εργασία	51

7.1 Συμπεράσματα

Μετά τη μελέτη των αποτελεσμάτων και τις κατάλληλες διορθώσεις που χρειάζονταν να γίνουν, καταλήξαμε στα ακόλουθα συμπεράσματα σχετικά με το σύστημα BioCT.

7.1.1 Ευχρηστία Συστήματος

Το σύστημα έχει σχετικά γρήγορη και υψηλή απόδοση, είναι εύκολο στη χρήση, εύκολο στην κατανόηση του περιεχομένου του και η μορφή του είναι τέτοια που το κάνει εύκολο στην εκμάθησή του.

7.1.2 Ικανοποίηση Απαιτήσεων Χρήστη

Αν εξαιρέσουμε τα δύο είδη λειτουργιών που δεν καταφέραμε να υλοποιήσουμε, όλες οι υπόλοιπες απαιτήσεις του χρήστη υλοποιήθηκαν επακριβώς, πράγμα που οδήγησε στην ικανοποίησή του. Επιπρόσθετα, συμπεράναμε ότι τα εργαλεία που χρησιμοποιήσαμε ήταν τα κατάλληλα όσον αφορά την απεικόνιση γιατί ήταν ξεκάθαρη και έκανε τη χρήση του συστήματος να είναι ευχάριστη για το χρήστη.

7.1.3 Δυνατότητες Επέκτασης Συστήματος

Το σύστημά μας είναι υλοποιημένο με τέτοιο τρόπο, ώστε να μπορεί να δεχτεί τυχόν μετατροπές και βελτιώσεις, καθώς και οποιεσδήποτε επεκτάσεις που μπορεί να προκύψουν στα πλαίσια μελλοντικής εργασίας.

7.2 Μελλοντική Εργασία

Η έρευνα και η ανάπτυξη του παρόν συστήματος, σίγουρα δε θα σταματήσει εδώ. Υπάρχουν πολλά είδη πιθανών μελλοντικών επεκτάσεων που θα βοηθούσαν στην κάλυψη περισσότερων αναγκών και απαιτήσεων.

7.2.1 Πρόσθεση των λειτουργιών που δεν υλοποιήθηκαν

Οι δύο λειτουργίες που δεν υλοποιήθηκαν είναι αλληλένδετες. Στα επόμενα υποκεφάλαια θα περιγραφούν τρόποι, οι οποίοι ίσως οδηγήσουν μελλοντικά στη λύση των προβλημάτων που αντιμετωπίστηκαν και κατ' επέκταση στην πρόσθεση των λειτουργιών αυτών

7.2.1.1 Πρόσθεση Νέων Κόμβων και Συνδέσεων και Δημιουργία Νέου Κυτταρικού Δικτύου

Η κύρια λειτουργία που δεν καταφέραμε να υλοποιήσουμε, είναι η δυνατότητα πρόσθεσης νέων κόμβων και κατ' επέκταση η δυνατότητα δημιουργίας νέου κυτταρικού δικτύου. Ως μελλοντική εργασία, θα μπορούσε να γίνει μια πιο εκτεταμένη έρευνα έτσι ώστε να ξεπεραστεί το πρόβλημα που αντιμετωπίσα σχετικά με τη λάθος αντιστοιχία των συντεταγμένων, για να βρεθεί η μαθηματική σχέση ή ο αλγόριθμος που να συσχετίζει τις συντεταγμένες της εικόνας με τις συντεταγμένες της οθόνης. Μ' αυτό τον τρόπο, θα μπορούσε να προστεθεί αυτή η λειτουργία στο σύστημα.

Για να γίνει αυτό, θα πρέπει πρώτα να βρεθεί ένας τρόπος για τη δυναμική εύρεση του μεγέθους της εικόνας του κυτταρικού δικτύου που δημιουργείται από το εργαλείο Graphviz, έτσι ώστε να μπορεί να προσαρμοστεί στα πλαίσια της οθόνης του συστήματος και να μπορούν να υπάρχουν οι βάσεις για να μπορέσει να βρεθεί πιο εύκολα η μαθηματική σχέση, ή ο αλγόριθμος συσχέτισης των δύο τύπων συντεταγμένων.

Επιπρόσθετα, θα μπορούσε να γίνει έρευνα, σχετικά με το κατά πόσο μπορεί να αλλάζει το μέγεθος της κάθε οθόνης του συστήματος, για να γίνεται προσαρμογή του μεγέθους της αντίστοιχης εικόνας που θα παραχθεί από το εργαλείο.

7.2.1.2 Λειτουργία του Zoom

Όπως ανέφερα και πιο πριν, η λειτουργία της πρόσθεσης νέων κόμβων και η λειτουργία του zoom, είναι αλληλένδετες. Ο λόγος είναι, γιατί αν βρεθεί τρόπος με τον οποίο να συσχετίζονται οι συντεταγμένες της οθόνης με τις συντεταγμένες της εικόνας που παράγεται από το εργαλείο Graphviz, θα λυνόταν και το πρόβλημα του zoom, αφού κύριο χαρακτηριστικό του είναι ότι παίρνει τις συντεταγμένες στις οποίες επιλέχθηκε να γίνει zoom-in ή zoom-out και μεγεθύνει ή μικραίνει την εικόνα ανάλογα.

Έτσι, αν λυθεί το πρόβλημα αυτό, θα μπορούσε να μελετηθεί και η λειτουργία του zoom, που αν και στο παρόν στάδιο είδαμε ότι δεν είναι αναγκαία, με την πρόσθεση νέων κόμβων ίσως να χρειαστεί. Έτσι, θα ήταν καλό να υλοποιηθεί και να προστεθεί.

7.2.2 Εισαγωγή των στοιχείων του Απαγωγικού Λογικού Προγραμματισμού

Στο παρόν στάδιο, ουσιαστικά το σύστημά μας δεν είναι λειτουργικό. Για να γίνει λειτουργικό και να μπορούν να παραχθούν σωστά η ανάλογες υποθέσεις, θα πρέπει να εισαχθούν τρόποι με τους οποίους να μπορούν να γίνουν οι υπολογισμοί αυτοί. Η σύνδεση του προγράμματός μας με μια γλώσσα λογικού προγραμματισμού, όπως η Sicstus Prolog, και η εισαγωγή στοιχείων απαγωγικού λογικού προγραμματισμού, θα ήταν μια πιθανή προσέγγιση.

Παρ' όλα αυτά, μια άλλη προσέγγιση για τους τρόπους με τους οποίους μπορεί το σύστημα να γίνει λειτουργικό, είναι με το να γίνει μια «προσομοίωση» των στοιχείων του λογικού προγραμματισμού, μέσω της γλώσσας Java, στην οποία είναι ήδη γραμμένο το σύστημα. Δηλαδή, θα μπορούσαμε για παράδειγμα, να δώσουμε σε όλους τους κόμβους αρχική τιμή π.χ. -1, και κάθε φορά που γίνεται κάποιος active μέσω των παρατηρήσεων του Βιολόγου, ο κόμβος θα παίρνει τιμή 1 και κάθε φορά που γίνεται inactive μέσω των παρατηρήσεων του Βιολόγου, ο κόμβος θα παίρνει την τιμή 0. Έτσι θα μπορούσαν να γίνονται οι αντίστοιχοι έλεγχοι για κάθε ένα κόμβο, ανάλογα με το είδος της ακμής που τον συνδέει με άλλους, για να υπολογιστούν οι καταστάσεις και των υπολοίπων κόμβων και να παράγεται η αντίστοιχη απεικόνιση.

Βιβλιογραφία

- [1] Antonis C. Kakas, Bert Van Nuffelen, Marc Denecker : A-System: Problem Solving through Abduction. IJCAI 2001, 591-596
- [2] Arnold, Ken, James Gosling, and David Holmes. The Java programming language. Vol. 2. Reading: Addison-wesley, 1996.
- [3] Auber, David. "Tulip—A huge graph visualization framework." Graph Drawing Software. Springer Berlin Heidelberg, 2004. 105-126.
- [4] Bornstein, Benjamin J., et al. "LibSBML: an API library for SBML." Bioinformatics 24.6 (2008): 880-881.
- [5] BSD License Definition , <http://www.lininfo.org/bsdlicense.html>
- [6] Cell Signaling Technology , <http://www.cellsignal.com/>
- [7] Chapman, Brad, and Jeffrey Chang. "Biopython: Python tools for computational biology." ACM SIGBIO Newsletter 20.2 (2000): 15-19.
- [8] Clewley RH, Sherwood WE, LaMar MD, Guckenheimer JM (2007) PyDSTool, a software environment for dynamical systems modeling. URL <http://pydstool.sourceforge.net>
- [9] Constantinou, C., Neophytou, C. M., Vraka, P., Hyatt, J. A., Papas, K. A., & Constantinou, A. I. (2012). Induction of DNA damage and caspase-independent programmed cell death by vitamin E. Nutrition and cancer, 64(1), 136-152.
- [10] Ellson, John, et al. "Graphviz—open source graph drawing tools." Graph Drawing. Springer Berlin Heidelberg, 2002.
- [11] Elmore, S, Apoptosis: a review of programmed cell death. Toxicologic Pathology 35, 2007

- [12] Funahashi, A., Tanimura, N., Morohashi, M., and Kitano, H., [CellDesigner: a process diagram editor for gene-regulatory and biochemical networks](#), BIOSILICO, 1:159-162, 2003
- [13] Graphviz Java API, Created by Laszlo Szathmary, alias Jabba Laci, 2003—2011, <http://www.loria.fr/~szathmar/off/projects/java/GraphVizAPI/index.php>
- [14] Grayson, John E. Python and Tkinter programming. Vol. 140. Greenwich: Manning, 2000.
- [15] Grivell, L. Συστημική βιολογία στη σχολική τάξη; μετάφραση Davidova, E. και Στασινάκης, Π. , Science in school, issue 11 (2010), scienceinschool.org <http://www.scienceinschool.org/2009/issue11/systemsbiology/greek>
- [16] H. Schmidt, M. Jirstrand: Systems Biology Toolbox for MATLAB: A computational platform for research in Systems Biology, Bioinformatics Advance Access, November 2005, DOI 10.1093/bioinformatics/bti799
- [17] Hagberg, Aric, Dan Schult, and Pieter Swart. "NetworkX Documentation." 2009-9-21]. <http://networkx.lanl.gov> (2009).
- [18] Hucka, Michael, et al. "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models." Bioinformatics 19.4 (2003): 524-531.
- [19] Hunter, John D. "Matplotlib: A 2D graphics environment." Computing in Science & Engineering 9.3 (2007): 0090-95.
- [20] Jones, Thouis R., et al. "CellProfiler Analyst: data exploration and analysis software for complex image-based screens." BMC bioinformatics 9.1 (2008): 482.
- [21] Kakas, A. και Denecker, M., "Abductive Logic Programming", Special issue of Journal of Logic Programming, JLP, Vol. 44(1-3), Elsevier Science, 2000
- [22] Keegan, Patrick, et al. NetBeans (TM) IDE Field Guide: Developing Desktop, Web, Enterprise, and Mobile Applications. Prentice Hall PTR, 2006.

- [23] Koutsofios, Eleftherios, and Stephen North. Drawing graphs with dot. Technical Report 910904-59113-08TM, AT&T Bell Laboratories, Murray Hill, NJ, 1991.
- [24] MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.
- [25] Neophytou, C. M., Constantinou, C., Papageorgis, P., & Constantinou, A. I. (2014). d-alpha-tocopheryl polyethylene glycol succinate (TPGS) induces cell cycle arrest and apoptosis selectively in Survivin-overexpressing breast cancer cells. *Biochemical pharmacology*, 89(1), 31-42.
- [26] Ousterhout, John K. "Scripting: Higher level programming for the 21st century." *Computer* 31.3 (1998): 23-30.
- [27] Ousterhout, John K. "TCL and the TK Toolkit." Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley (1993).
- [28] Ramirez, Alejandro, et al. "ArgoUML User Manual." En ligne.< <http://argouml-stats.tigris.org/documentation/defaulthtmlcookbook/>> Consulté le 15 (2007).
- [29] Rappin, Noel, and Robin Dunn. *wxPython in Action*. Manning, 2006.
- [30] Rumbaugh, James, Ivar Jacobson, and Grady Booch. *Unified Modeling Language Reference Manual*, The. Pearson Higher Education, 2004.
- [31] Sauro, Herbert M., et al. "Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration." *Omics A Journal of Integrative Biology* 7.4 (2003): 355-372.
- [32] Schildt, Herbert. *Java 2: the complete reference*. McGraw-Hill Professional, 2000.
- [33] Systems Biology Toolbox for MATLAB: A computational platform for research in Systems Biology, *Bioinformatics*, 22(4), 514-515, 2006, doi:[10.1093/bioinformatics/bti799](https://doi.org/10.1093/bioinformatics/bti799)

- [34] Systems-Biology : A portal site for Systems Biology, <http://www.systems-biology.org/>
- [35] The SBPOP Package - webpage: www.sbtoolbox2.org
- [36] Van Rossum, Guido, and Fred L. Drake. Python language reference manual. Network Theory, 2003.
- [37] Venners, Bill. Inside the Java virtual machine. McGraw-Hill, Inc., 1996.
- [38] 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
- [39] Κυτταρικός Κύκλος, Βιότοπος, Ιστότοπος από βιολόγους – εκπαιδευτικούς, Θεματική ενότητα : Το Κύτταρο , <http://www.edubiosite.gr/index.php/thematikes-enotites/to-kyttaro/o-kyttarikos-kyklos>
- [40] Σημειώσεις Μαθήματος «ΕΠΛ361 – Τεχνολογία Λογισμικού», Ενότητες 03-04, Γ. Καπιτσάκη, Τμήμα Πληροφορικής Παν. Κύπρου, 2012-2013
- [41] Σημειώσεις Μαθήματος «Τεχνολογία Λογισμικού Ι», Κεφάλαια 1-2, Β. Βεσκούκης, Πρόγραμμα Σπουδών : Πληροφορική, Ελληνικό Ανοιχτό Πανεπιστήμιο.
- [42] ΦΑΣΕΙΣ ΚΥΤΤΑΡΙΚΟΥ ΚΥΚΛΟΥ ΚΑΙ ΡΥΘΜΙΣΗ ΤΟΥ ΚΥΤΤΑΡΙΚΟΥ ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΥ, Σημειώσεις Αν. Καθηγήτριας Βιολογίας Ε. Κιτράκη, Τμήμα Οδοντιατρικής του Πανεπιστημίου Αθηνών

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). Είσατε 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 επιλεγθεί 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

Graphviz Java API

```
// 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;

/**
 * <dl>
 * <dt>Purpose: GraphViz Java API
 * <dd>
 *
 * <dt>Description:
```

```

* <dd> With this Java class you can simply call dot
*     from your Java programs
* <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="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:/temp"; // Windows

    /**
     * Where is your dot program located? It will be called externally.
     */
    // private static String DOT = "/usr/bin/dot"; // Linux
    private static String DOT = "c:/Program Files/Graphviz2.26.3/bin/dot.exe"; //
Windows

    /**
     * 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() {
}

/**
 * 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

```

Παράρτημα Γ

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 BioCT (Biological Cell-Processing Tool).zip
4. Αποσυμπίεση αρχείου BioCT.zip
5. Αντιγραφή του φακέλου BioCT στο directory C:\Users\user
6. Εκκίνηση του συστήματος (αρχείο BioCT.jar)

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=java>
[ee](#)
4. Αποσυμπίεση αρχείου BioCT (Biological Cell-Processing Tool).zip
5. Άνοιγμα του λογισμικού NetBeans->File->Import Project και επιλογή του αρχείου BioCT.zip