Εφαρμογές απεικόνισης και εξόρυξης δεδομένων σε βιολογικές βάσεις δεδομένων

Δ Ι Δ Α Κ Τ Ο Ρ Ι Κ Η Δ Ι Α Τ Ρ Ι Β Η

ΙΩΑΝΝΗΣ Γ. ΧΑΤΖΗΣ

Επιβλέπων: ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΟΥΛΑΣ, Αναπλ. Καθηγητής

Ιούλιος 2018

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Σχολή Επιστημών Υγείας

Τμήμα Φαρμακευτικής

Εφαρμογές απεικόνισης και εξόρυξης δεδομένων σε βιολογικές βάσεις δεδομένων

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΥΠΟΒΛΗΘΗΚΕ ΣΤΟ ΤΜΗΜΑ ΦΑΡΜΑΚΕΥΤΙΚΗΣ

ΤΟΥ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΠΑΤΡΩΝ

Ι Ω Α Ν Ν Η Σ Γ. Χ Α Τ Ζ Η Σ

Επταμελής Εξεταστική Επιτροπή

Κωνσταντίνος Πουλάς, Αναπληρωτής Καθηγητής (Επιβλέπων) [1]

Γεώργιος Πατρινός, Αναπληρωτής Καθηγητής (Μέλος Τριμελούς) [1]

Γιάννης Τζήμας, Αναπληρωτής Καθηγητής (Μέλος Τριμελούς) [2]

Γεώργιος Σπυρούλιας, Καθηγητής [1]

Γρηγόρης Σιβολαπένκο, Αναπληρωτής Καθηγητής [1]

Γεώργιος Παυλίδης, Καθηγητής [3]

Σπυρίδων Σιούτας, Αναπληρωτής Καθηγητής [4]

[1] Τμήμα Φαρμακευτικής, Πανεπιστήμιο Πατρών

[2] Τμήμα Μηχανικών Πληροφορικής Τ.Ε., ΤΕΙ Δυτικής Ελλάδας

[3] Τμήμα Μηχανικών Η/Υ & Πληροφορικής, Πανεπιστήμιο Πατρών

[4] Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την Τετάρτη, 11 Ιουλίου 2018

Πανεπιστήμιο Πατρών

Σχολή Επιστημών Υγείας / Τμήμα Φαρμακευτικής

ISBN 978-618-00-0355-0

Αφιερώνεται στην αγαπημένη μου Αγγελική,

αλλά και στα τρία καταπληκτικά παιδιά μας

ως ένδειξη της αξίας και της ομορφιάς

του "αγωνίζεσθαι"

Ευχαριστίες

Η παρούσα διδακτορική διατριβή που εκπονήθηκε στο Εργαστήριο Μοριακής Βιολογίας και Ανοσολογίας του Τμήματος Φαρμακευτικής του Πανεπιστημίου Πατρών είναι αποτέλεσμα μελέτης, έρευνας αλλά και αρκετής προσπάθειας. Είναι βέβαιο πως η ολοκλήρωσή της δεν θα ήταν δυνατή χωρίς την υποστήριξη αλλά και την ανεκτικότητα κάποιων ανθρώπων τους οποίους και επιθυμώ να μνημονεύσω και να ευχαριστήσω στις παρακάτω γραμμές, οι οποίοι μου έδωσαν την ευκαιρία και τη δυνατότητα να υλοποιήσω τη Διδακτορική μου Διατριβή.

Πρωτίστως θα ήθελα να ευχαριστήσω, τους συντονιστές της προσπάθειας. Τον επιβλέποντα καθηγητή μου, Κωνσταντίνο Πουλά, για τη συνεχή εποικοδομητική συνεργασία μας και επιστημονική υποστήριξη που μου παρείχε καθ' όλη τη διάρκεια εκπόνησης της παρούσας διατριβής. Ομοίως, θα ήθελα να ευχαριστήσω τον καθηγητή μου Γιάννη Τζήμα, για την ηθική, πνευματική στήριξη και την πολύτιμη βοήθειά του. Επίσης, είμαι πραγματικά ευγνώμων, αφενός για την εμπιστοσύνη που έδειξαν απέναντι στο πρόσωπο μου αυτά τα χρόνια και αφετέρου για την ανεκτίμητη συμπαράσταση σε όλες τις δυσκολίες που εμφανίστηκαν κατά τη διάρκεια εκπόνησης της διατριβής.

Ένα ευχαριστώ οφείλω και στους Μανώλη Βιεννά και Άντα Θηραίου, για τη συνεργασία και την υποστήριξη στη δημιουργία της RGDtrip καθώς και στο Μανούσο Καμπούρη, για τη συνεργασία στη δημιουργία της fungibase.

Δε μπορεί να ξεχάσω τους συναδέλφους μου, οι οποίοι στήριξαν την προσπάθειά μου όλα αυτά τα χρόνια, ώστε να μπορώ να καταφέρνω να

προχωρώ τη διατριβή μου αλλά και ταυτόχρονα να επιτελώ τις αρκετές επαγγελματικές μου υποχρεώσεις.

Τέλος, ένα μεγάλο ευχαριστώ οφείλω στους γονείς μου που ήταν και είναι πάντα δίπλα μου. Όμως σίγουρα αξίζει κι ένα τεράστιο ευχαριστώ στη σύζυγό μου Αγγελική, για την αδιάλειπτη υποστήριξη της σε όλες τις επιλογές της ζωής μου καθώς και στα τρία μας παιδιά, για την υπομονή και ανοχή που έδειξαν τα τελευταία χρόνια, απέναντι σε όλες τις ιδιαιτερότητες που παρουσιάζει ένας εργαζόμενος σε θέση Διευθυντή, πατέρας και σύζυγος ως υποψήφιος διδάκτορας.

Περιεχόμενα

Πίνακας εικόνων

Κεφάλαιο 1ο

Βάσεις Βιολογικών Δεδομένων

Τι είναι βάση δεδομένων

Ένας από τους πιο διαδεδομένους τομείς της πληροφορικής είναι η μελέτη και κατασκευή των βάσεων δεδομένων. Με την απλούστερη έννοια, ως βάση δεδομένων μπορεί να οριστεί ένα κατάλληλα δομημένο σύνολο από δεδομένα. Ο ρόλος μιας απλής βάσης δεδομένων είναι να εμφανίζει τα δεδομένα, χωρίς να καταφέρνει να παράγει πληροφορίες από αυτά ή πολύ περισσότερο γνώση. Κάτι τέτοιο απαιτεί την παρέμβαση του χρήστη. Αντίθετα, μία βάση δεδομένων που είτε χρησιμοποιείται για την άμεση καταγραφή ανθρώπινης γνώσης, είτε αξιοποιεί τα δεδομένα που καταχωρεί, ώστε μέσω του συνδυασμού τους να καταφέρνει να τα μετατρέψει σε γνώση (χωρίς απαραίτητα την ανθρώπινη παρέμβαση), τότε αυτή θα μπορούσε να θεωρείται μία βάση γνώσης. Δημιουργείται λοιπόν η ανάγκη για ένα ενιαίο μοντέλο ορισμού των δεδομένων, πληροφοριών και γνώσεων με βάση τους ρόλους τους στην υπολογιστική και γνωστική επεξεργασία πληροφοριών [1].

Μοντέλα βάσεων δεδομένων

Το περιεχόμενο μίας βάσης δεδομένων μπορεί να οργανωθεί με διάφορους τρόπους σύμφωνα με διάφορα μοντέλα που έχουν αναπτυχθεί. Το μοντέλο μιας βάσης δεδομένων καθορίζει το λογικό σχεδιασμό των

δεδομένων. Επίσης, περιγράφει τις σχέσεις μεταξύ των διαφόρων μερών των δεδομένων.

Το μοντέλο του απλού πίνακα, είναι πιθανότατα το πρώτο που χρησιμοποιήθηκε ποτέ. Ξεπεράστηκε μόλις πριν από μερικές δεκαετίες με το ιεραρχικό μοντέλο, το οποίο με τη σειρά του επεκτάθηκε μετά από μερικά χρόνια από το σχεσιακό, το οποίο θεμελιώθηκε μαθηματικά μέσω της σχεσιακής άλγεβρας. Η επιτυχία του σχεσιακού μοντέλου ήταν ιδιαίτερα μεγάλη και είναι το πλέον διαδεδομένο σήμερα. Δυστυχώς δεν είναι ιδιαίτερα συμβατό με τις πιο μοντέρνες μεθόδους περιγραφής οντολογιών.

Ως οντολογία ορίζεται ο τυπικός και σαφής ορισμός μιας κοινής και συμφωνημένης εννοιολογικής μορφοποίησης που αφορά σε ένα πεδίο ενδιαφέροντος. Αυτή η τυπική αναπαράσταση γνώσης ως ένα σύνολο εννοιών, σχέσεων και ιδιοτήτων μπορεί να χρησιμοποιηθεί για συλλογιστική (εξαγωγή συμπερασμάτων/νέας γνώσης) και για την δομημένη περιγραφή γνώσης ενός πεδίου ενδιαφέροντος [2].

Στην πληροφορική, και ιδιαίτερα στους τομείς της τεχνητής νοημοσύνης, του σημασιολογικού δικτύου και της βιοπληροφορικής, η οντολογία χρησιμοποιείται εκτενώς για την δόμηση και οργάνωση της πληροφορίας και την αναπαράσταση της γνώσης.

Το αντικειμενοστραφές μοντέλο δημιουργήθηκε για να προσφέρει λύσεις στα προβλήματα αυτά, χωρίς ακόμη όμως να έχει γίνει ακόμη ιδιαίτερα δημοφιλές. Τα πλεονεκτήματα αυτά όμως έρχονται με ένα σημαντικό υπολογιστικό κόστος και για τον λόγο αυτό κανένα από τα παραπάνω πρότυπα δεν έχει εκλείψει πλήρως.

Μοντέλο επίπεδου πίνακα

Ο πιο απλός τρόπος για να δομηθεί μία βάση δεδομένων είναι με την αποθήκευση των δεδομένων σε έναν πίνακα.

uniprotID	EntryName	proteinName	taxon	Sequence
Q6GZV6	019R_FRG3G	Putative serine	Viruses	MATNYCDEFERNPTRNP…
Q6GZV5	020R_FRG3G	Uncharacterized protein	Viruses	MLQNYAIVLGMAVAVA…
O55753	149L_IIV6	Uncharacterized protein	Viruses	MKETIFEIFVDDLMD…
Q59586	RECF_MYCTU	DNA replication and repair	Bacteria	MYVRHLGLRDFRSWACV…

Εικόνα 1 Καταγραφή δεδομένων πρωτεϊνών σε μορφή πίνακα

Με το μοντέλο αυτό αποθηκεύονται όμοια σύνολα πληροφοριών, με κάθε γραμμή να περιγράφει μία εγγραφή και κάθε στήλη να αναφέρεται σε ένα συγκεκριμένο χαρακτηριστικό της. Το μοντέλο αυτό αν και είναι ίσως το πιο αρχέγονο είναι το πιο εύκολο στην υλοποίηση. Ωστόσο εμφανίζει αρκετά προβλήματα όταν εφαρμόζεται για την περιγραφή σύνθετων δεδομένων στα οποία περιλαμβάνονται σχέσεις μεταξύ τους, μιας και αυτές θα πρέπει να τις περιγράψει μονοδιάστατα.

Ιεραρχικό μοντέλο

Οι βάσεις δεδομένων που ακολουθούν το ιεραρχικό μοντέλο, χρησιμοποιούν μια δενδροειδή δομή για να περιγράψουν δεδομένα. Τα δύο κύρια χαρακτηριστικά που έχουν αυτές οι δομές είναι οι κόμβοι και οι σχέσεις μεταξύ τους. Οι σχέσεις αυτές συνήθως περιγράφονται μονοσήμαντα και σε αντίθεση με τα κοινά δίκτυα η θέση ενός κόμβου ορίζει την ιεράρχηση της πληροφορίας. Σήμερα τα ιεραρχικά μοντέλα είναι αρκετά διαδεδομένα, ιδιαίτερα στο διαδίκτυο, καθώς ένα μεγάλο ποσοστό της ηλεκτρονικής επικοινωνίας πραγματοποιείται μέσω της ιεραρχικής μεταγλώσσας XML.

```
<taxon = “Viruses”>
   <uniprotID = “Q6GZV6” EntryName = “019R_FRG3G” proteinName = “Putative
      serine” Sequence = “MATNYCDEFERNPTRNP…” />
   <uniprotID = “Q6GZV6” EntryName = “020R_FRG3G” proteinName = “Uncharacterized
      protein” Sequence = “MLQNYAIVLGMAVAVA…” />
</taxon>
<taxon = “Bacteria”>
   <uniprotID = “Q59586” EntryName = “RECF_MYCTU” proteinName = “DNA replication
      and repair” Sequence = “MYVRHLGLRDFRSWACV…” />
</taxon>
```

Εικόνα 2 Καταγραφή δεδομένων πρωτεϊνών σε ιεραρχική μορφή XML

Σχεσιακό μοντέλο

Η πρώτη μεγάλη επανάσταση στο χώρο των βάσεων δεδομένων έγινε με την εισαγωγή του σχεσιακού μοντέλου από τον Edgar F. Codd το 1969, το οποίο είναι βασισμένο στη σχεσιακή άλγεβρα. Η φιλοσοφία του μοντέλου αυτού βασίζεται στα τρία κύρια στοιχεία μιας οντολογίας: οντότητες, χαρακτηριστικά και σχέσεις [3].

Η καταγραφή των στοιχείων αυτών γίνεται σε ένα σχεσιακό σχήμα, στο οποίο ορίζεται με μαθηματικό τρόπο η οντολογία της βάσης, η οποία πρακτικά μπορεί να ταυτιστεί απόλυτα με την περιγραφή οποιουδήποτε υποσυνόλου της πραγματικότητας. Το σχεσιακό μοντέλο είναι ιδιαίτερα επιτυχημένο και σήμερα είναι ο πλέον καθιερωμένος τρόπος αποθήκευσης πληροφοριών.

ID_taxon	taxon
1	Viruses
2	Bacteria

uniprotID	EntryName	proteinName	ID_taxon	Sequence
Q6GZV6	019R_FRG3G	Putative serine	1	MATNYCDEFERNPTRNP...
Q6GZV5	020R_FRG3G	Uncharacterized protein	1	MLQNYAIVLGMAVAVA...
O55753	149L_IIV6	Uncharacterized protein	1	MKETIFEIFVDDLMD...
Q59586	RECF_MYCTU	DNA replication and repair	2	MYVRHLGLRDFRSWACV...

Εικόνα 3 Καταγραφή δεδομένων πρωτεϊνών με το σχεσιακό μοντέλο

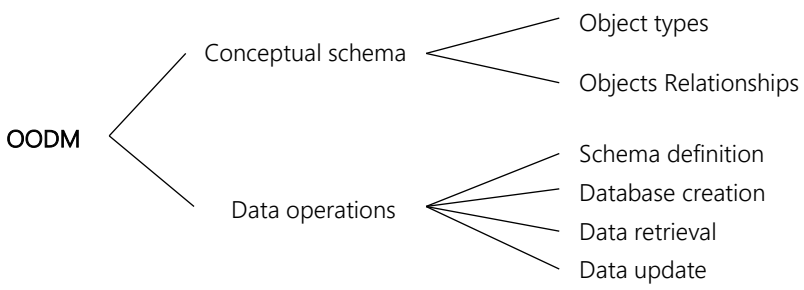
Παρ' όλη την ισχύ του, το μοντέλο αυτό έχει και αρκετές αδυναμίες, οι οποίες συνήθως εκδηλώνονται όταν η περιγραφή ενός τομέα ξεπερνάει κάποιο επίπεδο πολυπλοκότητας. Η κύρια αποτυχία του μοντέλου δεν έχει να κάνει τόσο πολύ με την περιγραφή της πληροφορίας, καθώς μέσω της οντολογίας μπορούμε πρακτικά να περιγράψουμε τα πάντα. Περισσότερο επικεντρώνεται στους μηχανισμούς ανάκτησής της, οι οποίοι μπορεί να γίνουν ιδιαίτερα πολύπλοκοι και δύσκολοι στην ανάγνωση και στη συντήρηση.

Αντικειμενοστραφές μοντέλο

Η περιγραφή δεδομένων που ακολουθεί το μοντέλο του αντικειμενοστραφούς προγραμματισμού έχει αρκετά προβλήματα συμβατότητας με αυτή του σχεσιακού μοντέλου, το οποίο για παράδειγμα δεν υποστηρίζει την κληρονομικότητα (inheritance), τις συμβολοσειρές (strings) αυθαιρέτου μεγέθους, την ενθυλάκωση (encapsulation), τον εύκολο ορισμό αυθαίρετων τελεστών ή τον πολυμορφισμό (polymorphism).

Πρακτικά, το αντικειμενοστραφές μοντέλο (OODM - object oriented data model) βασίζεται και αυτό σε οντολογίες όπως και το σχεσιακό, απλά χρησιμοποιεί πιο ευέλικτα και εύχρηστα εργαλεία για την περιγραφή τους, εργαλεία που δεν είναι άμεσα διαθέσιμα ή πολλές φορές είναι μη συμβατά με αυτά του σχεσιακού μοντέλου. Η ασυμβατότητα αυτή έχει δημιουργήσει ένα σύνηθες πρόβλημα στις εφαρμογές που ακολουθούν το OODM.

Καθώς δεν υπάρχει μια άμεση αντιστοίχιση του τρόπου περιγραφής των δεδομένων ανάμεσα στο επίπεδο εφαρμογής και στο επίπεδο της βάσης δεδομένων, πρέπει πάντα να μεσολαβεί ένα επιπλέον επίπεδο επεξεργασίας για τη μεταφορά τους από το ένα στο άλλο. Για να λυθεί το πρόβλημα αυτό, δημιουργήθηκαν οι αντικειμενοστραφείς βάσεις δεδομένων.



Εικόνα 4 Στοιχεία του αντικειμενοστραφούς μοντέλου δεδομένων

Παρόλα τα πλεονεκτήματα που έχουν, η εξοικείωση του κόσμου με τις σχεσιακές βάσεις δεδομένων είναι τόσο έντονη που δεν έχουν καταφέρει ακόμη να τις αντικαταστήσουν [4].

Συστήματα διαχείρισης βάσεων δεδομένων

Ένα σύστημα διαχείρισης βάσης δεδομένων (Database Management System) είναι ένα ενοποιημένο σύνολο από διεργασίες που επιτρέπει την αποθήκευση, ανάγνωση ή τροποποίηση δεδομένων, χωρίς τη διαταραχή της εσωτερικής ακεραιότητας μιας βάσης δεδομένων. Η χρήση τους επιτρέπει την απόκρυψη της εσωτερικής πολυπλοκότητας μιας βάσης από το χρήστη και καθορίζει την επικοινωνία της με τον εξωτερικό κόσμο μέσω μιας αυστηρά ορισμένης διεπαφής. Τα συστήματα διαχείρισης αντικατοπτρίζουν το μοντέλο των βάσεων δεδομένων που υποστηρίζουν.

Χαρακτηριστικό παράδειγμα σχεσιακών DBMS είναι το λογισμικό Oracle, ο Microsoft SQL Server, η MySQL, κά.

SQL – Structured Query Language

Η γλώσσα SQL (Structured Query Language) σχεδιάστηκε το 1974 ως μία γλώσσα διαχείρισης σχεσιακών βάσεων δεδομένων. Η δημιουργία της ήρθε σε μία εποχή που η μηχανική λογισμικού ήταν ακόμη στα εμβρυϊκά της στάδια και η αυξανόμενη πολυπλοκότητα των συστημάτων που αναπτύσσονταν είχε οδηγήσει τον τομέα σε μια ιδιαίτερα σοβαρή κρίση. Ο σκοπός της SQL ήταν να δημιουργηθεί μία δηλωτική γλώσσα η οποία θα επέτρεπε ακόμη και σε μη ειδικούς να επικοινωνούν με μία βάση δεδομένων με έναν δομημένο τρόπο. Η γλώσσα βασίστηκε στην σχεσιακή άλγεβρα και σχεδιάστηκε για να χρησιμοποιηθεί με βάση το σχεσιακό μοντέλο του Edgar F. Codd [5]. Η πρώτη υλοποίηση της δεν ακολούθησε πιστά το πρότυπο κατά Codd και έτσι σήμερα έχουν δημιουργηθεί διάφορες εκδόσεις, ο οποίες έχουν μικρά προβλήματα συμβατότητας μεταξύ τους.

Ένα από τα βασικότερα στοιχεία της γλώσσας είναι η έννοια του ερωτήματος (query) με το οποίο μπορεί κάνεις να επικοινωνήσει δομημένα και να χειριστεί μία βάση δεδομένων ελέγχοντας τις πληροφορίες που περιέχει. Για παράδειγμα, στη βάση RGDtrip που υλοποιήθηκε, το παρακάτω ερώτημα

επιστρέφει τα ονόματα όλων των πρωτεϊνών που έχουν καταγραφεί στον πίνακα Protein:

```
SELECT Name FROM [protein]
```

Τρόποι μεταφοράς δεδομένων

Ένα σημαντικό πρόβλημα που πρέπει να αναφερθεί είναι ο τρόπος μεταφοράς δεδομένων από μία βάση δεδομένων σε έναν χρήστη ή σε μία άλλη βάση δεδομένων. Τα δεδομένα αυτά πρέπει να μπορούν να παρουσιαστούν με τέτοιο τρόπο ώστε από τη μια να είναι κατανοητά στον αναγνώστη και από την άλλη να μεταφέρουν την πληροφορία τους με όλη της την πολυπλοκότητα, χωρίς απώλειες. Το πρόβλημα αυτό δεν είναι απαραίτητα απλό και υπάρχουν διάφορες λύσεις, η κάθε μία με τα πλεονεκτήματα και τα μειονεκτήματα, όπως παρουσιάζονται στη συνέχεια.

Επίπεδα αρχεία – flat files

Τα επίπεδα αρχεία είναι απλά αρχεία κειμένου δομημένα, συνήθως, σε μορφή comma separated (csv) ή tab delimited values που περιέχουν εγγραφές σε σειρά, κατά παράδοση μία ανά γραμμή. Το πρόβλημα με τα αρχεία αυτά είναι ότι η δομή τους ακολουθεί την αρχέγονη οργάνωση της πληροφορίας σε απλούς πίνακες και πολλές φορές αποδεικνύεται προβληματική όταν μεταφέρει δεδομένα από κανονικοποιημένες βάσεις δεδομένων καθώς είτε πρέπει να μεταφέρει τα δεδομένα κάθε πίνακα σε ξεχωριστά αρχεία ή πρέπει να τα μεταφέρει όλα σε ένα αρχείο αλλά σε αποκανονικοποιημένη μορφή.

```
uniprotID,EntryName,proteinName,taxon,Sequence
Q6GZV6,019R_FRG3G,Putative serine,Viruses,MATNYCDEFERNPTRNP…
Q6GZV5,020R_FRG3G,Uncharacterized protein,Viruses,MLQNYAIVLGMAVAVA…
O55753,149L_IIV6,Uncharacterized protein,Viruses,MKETIFEIFVDDLMD…
```

Εικόνα 5 Τμήμα csv αρχείου με δεδομένα πρωτεϊνών

Η δεύτερη λύση είναι αρκετά διαδεδομένη αλλά δυστυχώς δεν γίνεται πάντα με την απαραίτητη προσοχή, με αποτέλεσμα να προκαλείται απώλεια

πληροφορίας. Το κύριο πλεονέκτημα των flat files είναι ότι έχουν ιδιαίτερα μικρό μέγεθος και είναι σχετικά εύκολα στην ανάγνωση.

Αρχεία XML

Τα αρχεία XML διαφέρουν κατά πολύ από τα flat files, καθώς περιγράφουν ιεραρχικά δεδομένα χρησιμοποιώντας τη μετα-γλώσσα XML. Η γλώσσα που δημιούργησε ο διεθνής οργανισμός προτύπων W3C [6] είναι αυτοπεριγραφόμενη, δηλαδή δεν έχει κάποιο προκαθορισμένο λεξιλόγιο και κατασκευάζεται ανάλογα με τα δεδομένα που περιγράφει. Ο τρόπος περιγραφής είναι δομημένος και μπορεί να διαβαστεί αυτόματα από μηχανές αλλά και ανθρώπους. Η δομή ενός τέτοιου αρχείου μπορεί να περιγραφεί από τρία κύρια χαρακτηριστικά. Το πρώτο και το πιο βασικό είναι ο κόμβος (node) μέσω του οποίου ορίζεται η ιεραρχία και αποθηκεύονται οι πληροφορίες. Κάθε κόμβος (node) μπορεί να έχει ένα σύνολο από χαρακτηριστικά (attributes), μία τιμή (value) ή έναν αριθμό από υπο-κόμβους (sub-nodes). Αν και δεν επιβάλλεται η χρήση του, η οργάνωση ενός αρχείου XML μπορεί να ορίζεται από ένα εξωτερικό σχήμα (schema) το οποίο θέτει κανόνες και περιορισμούς για το πώς μπορεί να δομηθεί.

```
<?xml version="1.0"?>
<proteins>
 <uniprotID="Q6GZV6">
 <EntryName>019R_FRG3G</EntryName>
 <proteinName>Putative serine</proteinName>
 <taxon>Viruses</taxon>
 <Sequence>MATNYCDEFERNPTRNP…</Sequence>
 </uniprotID>
 <uniprotID="Q6GZV5">
 <EntryName>020R_FRG3G</EntryName>
 <proteinName>Uncharacterized protein</proteinName>
 <taxon>Viruses</taxon>
 <Sequence>MLQNYAIVLGMAVAVA…</Sequence>
 </uniprotID>
 .
 .
</proteins>
```

Εικόνα 6 Τμήμα XML αρχείου με δεδομένα πρωτεϊνών

Από το παράδειγμα η επεκτασιμότητα είναι προφανής αφού τα ονόματα των ετικετών επιλέχτηκαν ώστε να περιγράφουν με τον καλύτερο δυνατό τρόπο τα δεδομένα. Τα πραγματικά δεδομένα περιέχονται μεταξύ των ετικετών.

Η γλώσσα XML αποτελεί ακόμη και σήμερα, έναν από τους βασικότερους τρόπους επικοινωνίας στο διαδίκτυο και είναι τόσο διαδεδομένη που έχει αρχίσει να χρησιμοποιείται και ως μέσο αποθήκευσης για βάσεις δεδομένων. Το μόνο της μειονέκτημα σε σχέση με τα flat files είναι το γεγονός ότι για να περιγράψει τα ίδια δεδομένα χρειάζεται πολύ περισσότερο χώρο, καθώς ενώ στα flat files χρειαζόμαστε ένα απλό header για να περιγράψουμε την οργάνωση της πληροφορίας, στην γλώσσα XML η οργάνωση της πληροφορίας αποτελεί μέρος των δεδομένων.

Κώδικας SQL

Πέρα από τους δύο προηγούμενους τρόπους μεταφοράς δεδομένων υπάρχει και η δυνατότητα μεταφοράς πληροφορίας απευθείας σε κώδικα SQL, η οποία αν και όχι ιδανική για ανάγνωση, επιτρέπει την άμεση εισαγωγή της πληροφορίας σε μία βάση δεδομένων.

```
--
-- Δομή πίνακα για τον πίνακα `proteins`
--
CREATE TABLE `proteins` (
 `uniprotID` varchar(10) NOT NULL,
 `EntryName` varchar(50) NOT NULL,
 `proteinName` varchar(255) NOT NULL,
 `taxon` int(2) NOT NULL,
 `Sequence` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
--
-- Εισαγωγή δεδομένων του πίνακα `proteins`
--
INSERT INTO `proteins` (`uniprotID`, `EntryName`, `proteinName`, `taxon`,
`Sequence`) VALUES
('Q6GZV5', '020R_FRG3G', 'Uncharacterized protein', 1, 'MLQNYAIVLGMAVAVA…'),
('Q6GZV6', '019R_FRG3G', 'Putative serine', 1, 'MATNYCDEFERNPTRNP…');
```

Εικόνα 7 Τμήμα SQL αρχείου με δεδομένα πρωτεϊνών

Η χρήση της μεθόδου αυτής είναι πιο σπάνια καθώς έχει πολύ συγκεκριμένες χρήσεις και χρησιμοποιείται κυρίως όταν θέλουμε να κάνουμε ολόκληρα αντίγραφα μιας βάσης με σκοπό να την αναπαράγουμε κάπου αλλού.

Βάσεις βιολογικών δεδομένων

Οι δημόσιες (ελεύθερα προσβάσιμες) βάσεις βιολογικών δεδομένων (ΒΒΔ) είναι ηλεκτρονικές βιβλιοθήκες βιολογικών δεδομένων, τα οποία προέρχονται από διάφορες πηγές (π.χ. πειραματικά εργαστήρια, δημοσιευμένη βιβλιογραφία). Οι βάσεις αυτές αποτελούν ολοκληρωμένα συστήματα καταχώρησης βιολογικών δεδομένων (εγγραφών, entries) και συνοδεύονται από κατάλληλες μηχανές αναζήτησης και ανάκτησης των αποθηκευμένων δεδομένων [7].

Οι τύποι δεδομένων που συναντώνται είναι αυτοί που παράγονται από την βιολογική έρευνα, όπως για παράδειγμα, ακολουθίες πρωτεϊνών και νουκλεοτιδίων, γονιδιώματα, 3D δομές πρωτεϊνών και δεδομένα γενετικής ποικιλότητας. Ως επακόλουθο είναι η ύπαρξη διαφόρων Βάσεων Δεδομένων οι οποίες περιγράφονται παρακάτω [8].

Τύποι βάσεων βιολογικών δεδομένων

Πέρα από το μοντέλο εσωτερικής οργάνωσης, μπορούμε να διαχωρίσουμε τις βάσεις δεδομένων ανάλογα με το περιεχόμενο τους ή το σκοπό λειτουργίας τους.

Ο διαχωρισμός αυτός αποκτά ιδιαίτερο νόημα στη βιοπληροφορική και την συστημική βιολογία καθώς τα δεδομένα που επεξεργάζονται είναι συνήθως δυναμικά, μεγάλου όγκου και περιλαμβάνουν πολύπλοκες συσχετίσεις.

Βιβλιογραφικές Βάσεις Δεδομένων

Μια βιβλιογραφική βάση δεδομένων, καλύπτει τους τομείς της ιατρικής, της υγειονομικής περίθαλψης, των προκλινικών επιστημών, της βιολογίας καθώς και θέματα βιοϊατρικής τεχνολογίας με βιβλιογραφικό υλικό. Περιέχει βιβλιογραφικές παραπομπές και περιλήψεις άρθρων από βιοϊατρικά περιοδικά.

Μερικές βάσεις δεδομένων που χρησιμοποιούνται συνήθως από αξιόπιστες ιατρικές μηχανές αναζήτησης είναι [9]:

- EMBASE η οποία ανήκει στον οίκο Elsevier
- MEDLINE της US National Library of Medicine (NLM)
- PsycINFO
- Cochrane Database of Systematic Reviews (CDSR)
- MedlinePlus

Το περιεχόμενο των παραπάνω βάσεων, παίζει σημαντικό ρόλο στο πεδίο της εξόρυξης δεδομένων κειμένου (text mining) της βιοπληροφορικής.

Στην περίπτωση της MEDLINE, η πρόσβαση γίνεται από την υπηρεσία PubMed. Η **PubMed** περιλαμβάνει περισσότερες από 28 εκατομμύρια αναφορές για Βιοϊατρική βιβλιογραφία, επιστημονικά περιοδικά και online βιβλία, που βρίσκονται καταχωρημένα στη MEDLINE. Οι αναφορές μπορεί να περιλαμβάνουν συνδέσεις σε περιεχόμενο πλήρους κειμένου από τις τοποθεσίες PubMed Central και τους ιστότοπους των εκδοτών [10, 11].

Επίσης, η **Ovid** είναι μια συγκρίσιμη μηχανή αναζήτησης με την PubMed. Το πλεονέκτημά της έναντι της PubMed είναι ότι αναζητά περισσότερες βάσεις δεδομένων εκτός από το MEDLINE, συμπεριλαμβανομένου της EMBASE και της βάσης δεδομένων των συστηματικών ανασκοπήσεων (CDSR). Αυτό σημαίνει ότι η αναζήτηση θα επεκταθεί ώστε να περιλαμβάνει περισσότερα αποτελέσματα δίνοντας περισσότερα αποτελέσματα.

Για τους χρήστες που χρησιμοποιούν τακτικά τη Google για αναζήτηση στο διαδίκτυο, η χρήση του **Google Scholar** είναι ένας δωρεάν και εύκολος τρόπος πλοήγησης και φιλτραρίσματος των αποτελεσμάτων κάθε αναζήτησης. Συνήθως, το Google Scholar είναι μια εναλλακτική λύση αναζήτησης που μπορεί

να χρησιμοποιηθεί μετά την αρχική αναζήτηση. Εάν υπάρχει δυσκολία στην ανεύρεση ενός άρθρου, συχνά μια αναζήτηση στο Google Scholar μπορεί να βοηθήσει στον εντοπισμό του.

Ταξινομικές Βάσεις Δεδομένων

Έχει αναπτυχθεί μια ιδιαίτερη κατηγορία βάσεων δεδομένων, η οποία βασίζεται σε δεδομένα ακολουθίας και έχει ως στόχο την ταξινόμηση των οργανισμών για τους οποίους υπάρχουν δεδομένα νουκλεοτιδικών ή πρωτεϊνικών αλληλουχιών. Μια ενδεικτική ταξινομική βάση δεδομένων είναι ο Taxonomy Browser του NCBI (National Center for Biotechnology Information) [10, 12] περιέχει μόνο εκείνους τους οργανισμούς που αντιπροσωπεύονται στις γενετικές βάσεις δεδομένων με τουλάχιστον μία πρωτεΐνη ή αλληλουχία νουκλεοτιδίων, ενώ χρησιμοποιεί ιεραρχική ταξινόμηση.

Νουκλεοτιδικές Βάσεις Δεδομένων

Στις βάσεις αυτές, τα δεδομένα προέρχονται από την επιστημονική κοινότητα και είναι ελεύθερα, διαθέσιμα. Τα στοιχεία που εισάγονται, συνήθως είναι ετερογενή, ποικίλλουν όσον αφορά την προέλευση του υλικού, την ποιότητά του και την πληρότητα της ακολουθίας σχετικά με το βιολογικό στόχο. Οι τρεις μεγαλύτερες δημόσιες ΒΔ νουκλεοτιδικών αλληλουχιών είναι η GenBank [13], η EMBL_Bank [14] και η DNA Data Bank της Ιαπωνίας (DDBJ) [15] συνεργάζονται και έχουν δημιουργήσει την International Nucleotide Sequence Database Collaboration. Η συνεργασία αυτή περιλαμβάνει τη δημιουργία κοινών κανόνων για τον σχολιασμό των δεδομένων και την καθημερινή ανταλλαγή των εγγραφών που κατατίθενται ανεξάρτητα σε κάθε Βάση [16].

Γονιδιωματικές Βάσεις Δεδομένων

Αν και οι γονιδιωματικές ακολουθίες αποτελούν καταχωρήσεις σε Νουκλεοτιδικές Βάσεις Δεδομένων, για πολλά είδη έχουν αναπτυχθεί ειδικές

βάσεις που συνδυάζουν τα δεδομένα γονιδιωματικών αλληλουχιών και το σχολιασμό τους με άλλα στοιχεία για τα συγκεκριμένα είδη. Οι Βάσεις αυτές παρουσιάζουν μια ποικιλομορφία όσον αφορά στο είδος και στον τρόπο αποθήκευσης δεδομένων. Παραδείγματα τέτοιων Βάσεων είναι η Ensembl [17] και η Entrez Genomes [10].

Το έργο **Ensembl** ξεκίνησε το 1999, μερικά χρόνια πριν από την ολοκλήρωση του σχεδίου ανθρώπινου γονιδιώματος. Ακόμη και σε αυτό το πρώιμο στάδιο ήταν σαφές ότι ο χειροκίνητος σχολιασμός των τριών δισεκατομμυρίων ζευγών βάσεων αλληλουχίας δεν θα ήταν σε θέση να προσφέρει στους ερευνητές την έγκαιρη πρόσβαση στα τελευταία δεδομένα. Ο στόχος του Ensembl ήταν να δίνει τη δυνατότητα του αυτόματου σχολιασμού για το γονιδίωμα, να ενσωματώσει αυτόν το σχολιασμό με άλλα διαθέσιμα βιολογικά στοιχεία και να καταστήσει όλα αυτά δημόσια διαθέσιμα μέσω του ιστού. Από την έναρξη λειτουργίας της, τον Ιούλιο του 2000, πολλά περισσότερα γονιδιώματα έχουν προστεθεί στην Ensembl και το φάσμα των διαθέσιμων δεδομένων έχει επεκταθεί, ώστε να συμπεριλαμβάνει συγκριτικά στοιχεία γονιδιωματικής, παραλλαγή και κανονιστικά δεδομένα [18].

Η βάση δεδομένων **Entrez Genome** περιέχει δεδομένα αλληλουχίας και χάρτη από ολόκληρα γονιδιώματα για περισσότερα από 1000 είδη ή στελέχη. Τα γονιδιώματα αντιπροσωπεύουν τόσο γονιδιώματα με πλήρη αλληλουχία όσο και γονιδιώματα με αλληλουχία σε εξέλιξη. Και οι τρεις κύριοι τομείς της ζωής (βακτήρια, αρχαία και ευκαρυώτες) αντιπροσωπεύονται, καθώς και πολλοί ιοί, φάγοι, πλασμίδια και οργανίδια [19].

Πρωτεϊνικές Βάσεις Δεδομένων

Οι πρωτεϊνικές Βάσεις Δεδομένων είναι η περιεκτικότερη πηγή πληροφοριών για τις πρωτεΐνες. Οι πρωτεϊνικές Βάσεις Δεδομένων περιλαμβάνουν πρωτεϊνικές ακολουθίες που συνάγονται με υπολογιστική μετάφραση αλληλουχιών αποθηκευμένων στις νουκλεοτιδικές βάσεις δεδομένων ή από πειραματική αλληλούχηση πρωτεϊνών.

Οι πρωτεϊνικές βάσεις διαχωρίζονται στις **πρωτογενείς** και τις **δευτερογενείς** [7].

Οι πρωτογενείς βάσεις δεδομένων περιλαμβάνουν βιολογικά δεδομένα στην πρωτογενή τους μορφή, όπως αυτά προσδιορίζονται από τους πειραματικούς επιστήμονες, ενώ συνήθως περιέχουν επιπλέον ταξινόμηση και σχολιασμό. Συνήθως λειτουργούν ως καταθετήρια δεδομένων, καταγράφοντας μη επεξεργασμένα τα δεδομένα. Αυτό το χαρακτηριστικό είναι ιδιαίτερα σημαντικό, καθώς οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν για διάφορους λόγους και συνήθως είναι απαραίτητο να είναι άμεσα προσβάσιμες και διαθέσιμες στην αρχική μορφή με την οποία καταχωρήθηκαν από την πηγή πληροφορίας.

Οι δευτερογενείς βάσεις δεδομένων περιέχουν αποτελέσματα της επεξεργασίας βιολογικών δεδομένων που έχουν προέλθει από πρωτογενείς βάσεις. Το αποτέλεσμα της επεξεργασίας αυτής είναι ο εμπλουτισμός των αρχικών δεδομένων με επιπλέον χαρακτηριστικά, επισημάνσεις και σχέσεις που δεν ήταν διαθέσιμα.

Δομικές Βάσεις Δεδομένων

Αυτές οι Βάσεις περιέχουν δομική πληροφορία για μόρια πρωτεϊνών, νουκλεϊνικών οξέων και υδατανθράκων. Περισσότερο γνωστές είναι η Protein Data Bank (PDB) και η Nucleic Acid Database (NDB). Η PDB (Protein Data Bank) [20] αποτελεί μια διεθνή αποθήκη πειραματικά προσδιορισμένων τριτοταγών δομών πρωτεϊνών, νουκλεϊνικών οξέων και συμπλοκών βιομακρομορίων, ενώ παρέχει μια συλλογή εργαλείων λογισμικού για την ανάλυση των μακρομοριακών δομών.

Βάσεις Δεδομένων Ταξινόμησης Πρωτεϊνικών Δομών

Οι Βάσεις αυτές αποτελούν ταξινομίες πρωτεϊνικής δομής. Με άλλα λόγια, οι πρωτεΐνες που μοιάζουν από άποψη μορφής και τοπολογίας, είναι

ταξινομημένες ως πιο στενά συνδεδεμένες σε σχέση με πρωτεΐνες που φαίνονται ουσιαστικά διαφορετικές. Στη βάση δεδομένων CATH [21], οι δομές ταξινομούνται σύμφωνα με τις κατηγορίες: Class (τάξη), Architecture (αρχιτεκτονική), Topology (τοπολογία), και Homology (ομολογία). Αντίστοιχα, στη βάση SCOP (Structural Classification Of Proteins) [22] οι πρωτεϊνικές δομές ταξινομούνται ιεραρχικά σε οικογένειες, υπεροικογένειες, πτυχώσεις, και τάξεις.

Εξειδικευμένες Βάσεις Δεδομένων

Πρόκειται για Βάσεις Δεδομένων που περιέχουν στοιχεία από συγκεκριμένους οργανισμούς, συγκεκριμένες κατηγορίες και λειτουργίες αλληλουχιών, ή δεδομένα που παράγονται από συγκεκριμένες τεχνολογίες αλληλούχισης (sequencing technologies). Χαρακτηριστική είναι η βάση Ολοκληρωμένων Μικροβιακών Γονιδιωμάτων και Μικροβιομών (IMG/M), η οποία υποστηρίζει το σχολιασμό, την ανάλυση και τη διανομή σειρών δεδομένων μικροβιακού γονιδιώματος και μικροβίων, που ταξινομήθηκαν στο Joint Genome Institute (JGI) [23].

Πρωτογενείς βάσεις δεδομένων

Όπως έχει ήδη αναφερθεί προηγούμενα, οι πρωτογενείς Βάσεις Δεδομένων περιέχουν πληροφορία για την ακολουθία των πρωτεϊνών.

Η **PIR** αποτελεί μια περιεκτική βάση δεδομένων πρωτεϊνικών αλληλουχιών με λειτουργικό και βιβλιογραφικό σχολιασμό των καταχωρήσεων [24].

Η **SwissProt** (Bairoch and Apweiler, 2000) αποτελεί μια ΒΔ πρωτεϊνικών αλληλουχιών η οποία παρέχει υψηλή ποιότητα μη αυτοματοποιημένου σχολιασμού (π.χ. περιγραφή της λειτουργίας μιας πρωτεΐνης, βιβλιογραφικές αναφορές κ.ο.κ), περιορισμένο βαθμό πλεονασμού (δηλαδή χωρίς αλληλεπικαλύψεις) και διασύνδεση με άλλες ΒΔ. Ωστόσο, η σχετικά αργή

διαδικασία σχολιασμού της SwissProt οδήγησε στη δημιουργία της συμπληρωματικής TrEMBL [25].

Η **TrEMBL** περιέχει καταχωρήσεις που δεν έχουν καταχωρηθεί στη βάση δεδομένων SwissProt αλλά προέκυψαν από τη μετάφραση των καταχωρημένων νουκλεοτιδικών αλληλουχιών της EMBL. Ο σχολιασμός των καταχωρήσεων γίνεται αυτοματοποιημένα.

Η **UniProt** αποτελεί μια παγκόσμια, πλήρη και αναλυτική βάση δεδομένων πρωτεϊνικών αλληλουχιών. Είναι το αποτέλεσμα της συνεργασίας των SwissProt, TrEMBL και PIR. Το κυρίως τμήμα της UniProt, η UniProtKB (UniProt knowledgebase), αποτελεί μια ταξινομημένη βάση, με ακριβή σχολιασμό των πρωτεϊνικών αλληλουχιών και εκτεταμένη διασύνδεση με άλλες βάσεις [26].

Τέλος, η **RefSeq** [27] του NCBI αποτελεί, επίσης, μια περιεκτική βάση δεδομένων.

Δευτερογενείς ή παραγόμενες βάσεις δεδομένων

Οι πρωτεΐνες συνήθως απαρτίζονται από μια ή περισσότερες πρωτεϊνικές περιοχές. Η πρωτεϊνική περιοχή σχηματίζεται από κάθε τμήμα μιας πολυπεπτιδικής αλυσίδας που μπορεί να διπλωθεί ανεξάρτητα σε μια συμπαγή, σταθερή δομή. Οι πρωτεΐνες με πολύπλοκη λειτουργία συνήθως αποτελούνται από ένα συνδυασμό πρωτεϊνικών περιοχών που αλληλεπιδρούν μεταξύ τους [28, 29].

Οι βάσεις δεδομένων των δομικών περιοχών ή των οικογενειών πρωτεϊνών είναι χρήσιμες για την ανάλυση των πρωτεϊνών και συγκεκριμένα για τον χαρακτηρισμό της λειτουργίας τους. Αυτές επιτρέπουν την αναγνώριση των δομικών περιοχών ή των οικογενειών πρωτεϊνών, η οποία έχει προκύψει από την επεξεργασία πολλαπλών στοιχίσεων-συγκρίσεων ενός συνόλου ομόλογων αλληλουχιών πρωτεϊνών χρησιμοποιώντας ποικίλες μεθόδους [30]. Για το λόγο αυτό, οι βάσεις αυτές συνήθως αποκαλούνται "βάσεις δεδομένων πρωτεϊνικών υπογραφών" (protein signature databases).

Πρέπει να σημειωθεί ότι οι δευτερογενείς βάσεις χρησιμοποιούν συγκεκριμένα κομμάτια από τις πρωτογενείς, (ανάλογα με το σκοπό λειτουργίας τους) και διαφορετικές προσεγγίσεις ή συνδυασμό προσεγγίσεων για την εξαγωγή της υπογραφής μιας πρωτεϊνικής οικογένειας.

Παραδείγματα τέτοιων βάσεων είναι:

- BLOCKS, http://blocks.fhcrc.org/blocks
- ODD, http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml
- Pfam, http://pfam.sanger.ac.uk
- PRINTS, http://www.bioinf.manchester.ac.uk/dbbrowser/PRINTS/index.php
- PROSITE, http://au.expasy.org/prosite
- SMART, http://smart.embl.de

Μετα-βάσεις δεδομένων

Μια μετα-βάση δεδομένων (μερικές φορές ονομάζεται μετα-βάση) είναι μια βάση δεδομένων για την αποθήκευση μεταδεδομένων (δεδομένα που περιγράφουν δεδομένα) για συγκεκριμένο σκοπό. Μια φυσική μετα-βάση είναι αυτή στην οποία τα μεταδεδομένα συλλέγονται στην πραγματικότητα σε ένα μόνο σημείο προτού προσπελαστεί. Μια εικονική μετα-βάση είναι αυτή στην οποία συγκεντρώνονται τα μεταδεδομένα όταν χρειαστεί, πιθανώς όταν εκτελείται ένα πρόγραμμα.

Επίσης, ο όρος μετα-βάση δεδομένων χρησιμοποιείται ορθά και για να περιγράψει βάσεις δεδομένων που αποθηκεύουν άλλες βάσεις δεδομένων. Ο σκοπός αυτών των μετα-βάσεων είναι να ενσωματώσουν πληροφορίες (ενδεχομένως ετερογενείς) από διαφορετικές πηγές κάτω από ένα ενιαίο σχήμα και όχι να τις εμπλουτίσουν (αν και αρκετά συχνά αυτό αποτελεί μια θετική παρενέργεια της διαδικασίας ενοποίησης).

Μετα-βάσεις που έχουν αναπτυχθεί για να συλλέγουν βιολογικά δεδομένα από διαφορετικές πηγές και συνήθως τα καθιστούν διαθέσιμα σε νέα και πιο βολική μορφή ή με έμφαση σε μια συγκεκριμένη ασθένεια ή οργανισμό, ενδεικτικά είναι:

- Entrez. Εκτελεί αναζήτηση σε όλες τις βάσεις δεδομένων του NCBI, https://www.ncbi.nlm.nih.gov/sites/gquery
- euGenes. Αναζητά γονιδιωματικές πληροφορίες για ευκαρυωτικούς οργανισμούς [32], http://eugenes.org
- GeneCards. Είναι μια μετα-βάση δεδομένων που μπορεί να αναζητήσει, να ενσωματώσει και να παρέχει ολοκληρωμένες και φιλικές προς το χρήστη πληροφορίες, για όλα τα σχολιασμένα και προβλεπόμενα ανθρώπινα γονίδια. Αυτόματα ενσωματώνει γονιδιακά δεδομένα από περίπου 125 πηγές ιστού, συμπεριλαμβανομένων γονιδιωματικών, μεταγραφικών, πρωτεϊνωματικών, γενετικών, κλινικών και λειτουργικών πληροφοριών [32], http://www.genecards.org

Η οντολογία της γενετικής πληροφορίας

Η διαρκώς αυξανόμενη αποκρυπτογράφηση μοριακών αλληλουχιών, ιδιαίτερα αλληλουχιών ολόκληρων γονιδιωμάτων, άλλαξε την οπτική υπό την οποία εξετάζεται η θεωρία και η πράξη στην πειραματική βιολογία. Παλαιότερα, οι βιοχημικοί συνήθιζαν να χαρακτηρίζουν τις πρωτεΐνες από τις διαφορές στις δραστηριότητές τους και οι γενετιστές τα γονίδια από το φαινότυπο των μεταλλάξεων τους. Τώρα πια έχει γίνει κοινά αποδεκτό ότι υπάρχουν χαρακτηριστικές ομάδες γονιδίων και πρωτεϊνών που έχουν διατηρηθεί (για τη χρησιμότητά τους) σε κύτταρα κάθε είδους οργανισμού. Αυτή η αποδοχή έχει ωθήσει στην ενοποίηση της βιολογίας. Με άλλα λόγια, η γνώση του βιολογικού ρόλου μίας πρωτεΐνης σε ένα οργανισμό που εμφανίζεται να έχει κοινά στοιχεία με αντίστοιχες πρωτεΐνες σε άλλους οργανισμούς μπορεί να διαφωτίσει το ρόλο των αντίστοιχων γονιδιακών προϊόντων και στους άλλους οργανισμούς.

Κατά συνέπεια, η κατανόηση της οργάνωσης της γενετικής πληροφορίας είναι απαραίτητη για να είναι δυνατή η λειτουργία των πρωτεϊνικών αλληλεπιδράσεων. Η πολυπλοκότητα δε, της γενετικής οργάνωσης είναι τέτοια που συχνά τέτοιες περιγραφές δεν μπορούν να χαρακτηρίσουν τις πρωτεΐνες μοναδικά και εισάγουν αβεβαιότητα.

Καθώς λοιπόν η μελέτη των επιπέδων της γενετικής οργάνωσης (και ιδίως του πρωτεϊνώματος) βρίσκεται σε εξέλιξη, η οντολογία της γενετικής πληροφορίας δεν είναι ακόμη πλήρως γνωστή ή τουλάχιστον βρίσκεται υπό εξέταση. Το σύνολο των πρωτεϊνικών αλληλουχιών που προκύπτουν από την μεταγραφή και μετάφραση όλων των κωδικών αλληλουχιών ενός γονιδιώματος ονομάζεται πλήρες πρωτέωμα (complete proteome). Πολλές από τις αλληλουχίες αυτές έχουν προκύψει από αυτόματους μηχανισμούς πρόβλεψης και δεν έχουν επιβεβαιωθεί πειραματικά ακόμη.

Η περιγραφή και ο ορισμός τέτοιων κοινών βιολογικών στοιχείων και λεπτομερούς χαρακτηρισμού των γονιδίων και των προϊόντων (καθώς και των ιδιοτήτων τους) δεν ακολουθεί κάποιο κοινό πρότυπο με αποτέλεσμα την έλλειψη ενός ενιαίου μοντέλου που θα συγκέντρωνε όλα τα απαραίτητα στοιχεία για το χαρακτηρισμό γονιδίων (και των προϊόντων τους) υπό μία κοινώς αποδεκτή και τυποποιημένη φόρμα.

Αυτό οδήγησε στην ανάληψη πρωτοβουλίας για την ενοποίηση και σύνδεση όλων των γονιδιωματικών βάσεων δεδομένων, η οποία αποτέλεσε την κυριότερη αιτία για τη δημιουργία της Γονιδιακής Οντολογίας [33].

Γονιδιακή Οντολογία - Gene Ontology (GO)

Το έργο της Γονιδιακής Οντολογίας ξεκίνησε το 1988 στα πλαίσια της συνεργασίας που αναπτύχθηκε μεταξύ των ερευνητικών ομάδων τριών οργανισμών της FlyBase (Drosophila), της Saccharomyces Genome Database (SGD) and της Mouse Genome Database (MGD) που μελετούσαν τα γονιδιώματα της φρουτόμυγας, του ζαχαρομύκητα, και του ποντικού αντίστοιχα.

Από τότε στην κοινοπραξία της Γονιδιακής Οντολογίας έχουν εισχωρήσει πολλές άλλες βάσεις δεδομένων, εκ των οποίων και μερικές από τις πιο σημαντικές πηγές πληροφορίας φυτικών, ζωικών και μικροβιακών γονιδιωμάτων [33].

Κύριος σκοπός τους ήταν να κατασκευάσουν έναν οικουμενικό τρόπο περιγραφής των χαρακτηριστικών και των λειτουργιών των γονιδίων καθώς

και των παραγώγων τους [34]. Παράγωγο της οντολογίας αυτής είναι ένα ελεγχόμενο λεξιλόγιο που περιγράφει τα διάφορα κυτταρικά διαμερίσματα, τις μοριακές λειτουργίες καθώς και τις διάφορες βιολογικές διαδικασίες [35].

Τα βιολογικά συστήματα είναι τόσο περίπλοκα που πρέπει να βασιζόμαστε στους υπολογιστές για να αναπαραστήσουμε αυτή τη γνώση. Το έργο έχει αναπτύξει επίσημες οντολογίες που αντιπροσωπεύουν πάνω από 40.000 βιολογικές έννοιες και αναθεωρούνται συνεχώς ώστε να αντικατοπτρίζουν νέες ανακαλύψεις. Μέχρι σήμερα, αυτές οι έννοιες έχουν χρησιμοποιηθεί για να "σχολιάσουν" τις λειτουργίες των γονιδίων με βάση τα πειράματα που αναφέρθηκαν σε πάνω από 100.000 επιστημονικές μελέτες.

Αναλυτικότερα, η Γονιδιακή Οντολογία αποτελεί ένα ελεγχόμενο λεξιλόγιο που είναι δομημένο και περιέχονται όροι, οι οποίοι είναι γνωστοί ως GO όροι (GO-terms). Η Γονιδιακή Οντολογία διαιρείται σε τρεις επιμέρους οντολογίες-απόψεις (aspects) οι οποίες παρέχουν πληροφορίες κοινές για όλα τα είδη οργανισμών. Αυτές οι τρεις οντολογίες-απόψεις είναι οι:

- μοριακή λειτουργία (Molecular Function, MF)
- κυτταρική σύσταση (Cellular Component, CC)
- βιολογική διαδικασία (Biological Process, BP)

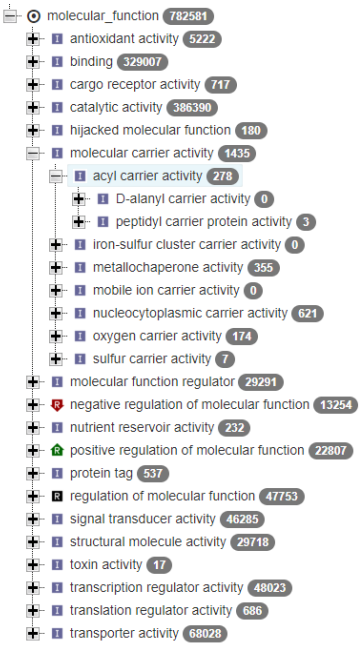
Το πιο σημαντικό στοιχείο αυτών των οντολογιών-απόψεων είναι η ορθογωνιότητα, η διασφάλιση δηλαδή ύπαρξης κάθε όρου μόνο σε μία από τις τρεις απόψεις. Η ιδιότητα αυτή εγγυάται τη μοναδικότητα ενός γνωρίσματος που αποδίδεται σε ένα γονίδιο. Με απλά λόγια, όταν ένας όρος Γονιδιακής Οντολογίας (GO-term) αποδίδεται σε ένα γονίδιο αυτόματα αποκλείεται η ύπαρξη ενός αντίστοιχου όρου από άλλη οντολογία-άποψη που να φέρει πανομοιότυπες ιδιότητες.

Τελικά, οι MF και CC οντολογίες απαντούν στο ερώτημα του τι κάνει ένα γονιδιακό προϊόν και σε ποια μέρη βρίσκεται ενεργό ενώ η BP οντολογία αποσαφηνίζει το βιολογικό σκοπό που επιτελεί ένα γονιδιακό προϊόν [36].

Μοριακή λειτουργία

Οι όροι μοριακής λειτουργίας περιγράφουν δραστηριότητες που συμβαίνουν σε μοριακό επίπεδο, όπως "καταλυτική δραστηριότητα" ή "δραστηριότητα δέσμευσης". Οι όροι της μοριακής συνάρτησης (GO-terms) αντιπροσωπεύουν τις δραστηριότητες και όχι τις οντότητες (μόρια ή σύμπλοκα) που εκτελούν τις ενέργειες χωρίς να καθορίζει το χρόνο ή το χώρο όπου αυτή (η δραστηριότητα) συνέβη. Οι μοριακές λειτουργίες γενικά αντιστοιχούν σε δραστηριότητες που μπορούν να εκτελεστούν από μεμονωμένα γονιδιακά προϊόντα, αλλά μερικές δραστηριότητες εκτελούνται από συναρμολογημένα σύμπλοκα γονιδιακών προϊόντων. Παραδείγματα ευρέων λειτουργικών όρων είναι η "καταλυτική δραστηριότητα" και η "δραστηριότητα μεταφορέων". Παραδείγματα στενότερων λειτουργικών όρων είναι η "δραστικότητα αδενυλικής κυκλάσης" ή η "δέσμευση του υποδοχέα Toll".

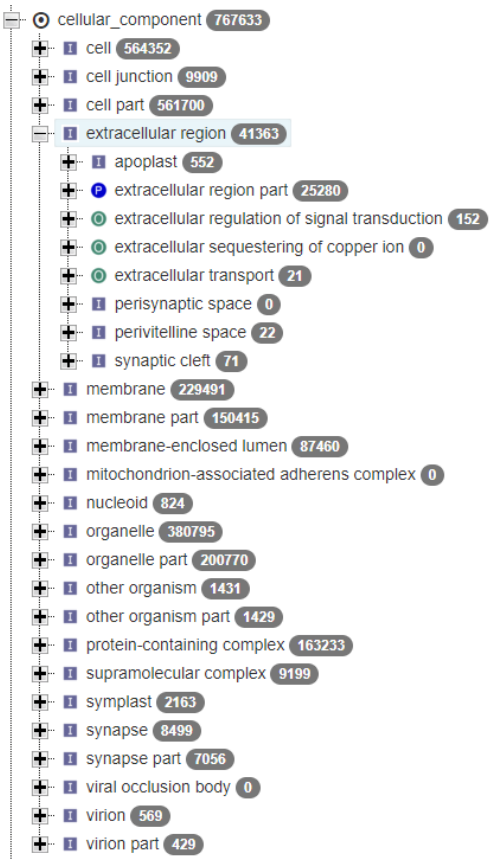
Είναι εύκολο να συγχέουμε ένα όνομα προϊόντος γονιδίου με τη μοριακή λειτουργία του. Για το λόγο αυτό οι μοριακές λειτουργίες GO συχνά ακολουθούνται από τη λέξη "δραστηριότητα" (activity).



Εικόνα 8 Στιγμιότυπο της Μοριακής Λειτουργίας σε δενδρική δομή

Κυτταρική σύσταση

Αυτοί οι όροι περιγράφουν μια θέση, σε σχέση με τα κυτταρικά διαμερίσματα και τις δομές, που καταλαμβάνονται από μια μακρομοριακή μηχανή όταν εκτελεί μια μοριακή λειτουργία. Υπάρχουν δύο τρόποι με τους οποίους οι βιολόγοι περιγράφουν τις θέσεις των γονιδιακών προϊόντων: (1) σε σχέση με τις κυτταρικές δομές (π.χ. κυτταροπλασματική πλευρά της μεμβράνης του πλάσματος) ή τα διαμερίσματα (π.χ. μιτοχόνδριο) και (2) τα σταθερά μακρομοριακά σύμπλοκα των οποίων είναι μέρη (π.χ. το ριβόσωμα). Σε αντίθεση με τις άλλες πτυχές του GO, οι έννοιες των κυτταρικών συστατικών δεν αναφέρονται σε διεργασίες, αλλά στην κυτταρική ανατομία.

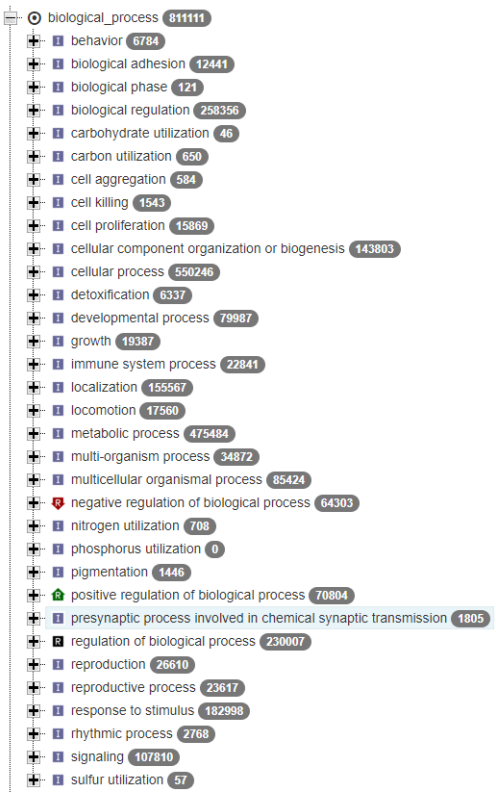


Εικόνα 9 Στιγμιότυπο της Κυτταρικής σύστασης σε δενδρική δομή

Βιολογική διαδικασία

Ένας όρος βιολογικής διεργασίας περιγράφει μια σειρά γεγονότων που πραγματοποιούνται από ένα ή περισσότερα οργανωμένα συγκροτήματα μοριακών λειτουργιών. Παραδείγματα ευρέων όρων βιολογικής διεργασίας είναι "κυτταρική φυσιολογική διαδικασία" ή "μεταγωγή σήματος". Παραδείγματα πιο συγκεκριμένων όρων είναι η «μεταβολική διαδικασία πυριμιδίνης» ή η «μεταφορά α-γλυκοσίδης». Ο γενικός κανόνας που βοηθά στη διάκριση μεταξύ μιας βιολογικής διεργασίας και μιας μοριακής συνάρτησης είναι ότι μια διαδικασία πρέπει να έχει περισσότερα από ένα ξεχωριστά βήματα.

Μια βιολογική διαδικασία δεν είναι ισοδύναμη με μια οδό. Προς το παρόν, ο GO δεν επιχειρεί να αντιπροσωπεύσει τη δυναμική ή τις εξαρτήσεις που θα απαιτούσε για να περιγράψει πλήρως μια πορεία.



Εικόνα 10 Στιγμιότυπο της Βιολογικής διαδικασίας σε δενδρική δομή

Ανάκτηση δεδομένων από βάσεις βιολογικών δεδομένων

Συνήθως, η ανάκτηση βιολογικών δεδομένων από τις ΒΒΔ γίνεται με «λέξεις-κλειδιά», τα οποία μπορεί να είναι το όνομα ενός οργανισμού, το όνομα μιας ακολουθίας ή ο αριθμός καταχώρησης μιας εγγραφής, ανάλογα με το είδος της πληροφορίας που θέλει να εξάγει ο χρήστης [7].

Επίσης, η μετα-βάση Entrez του NCBI (http://www.ncbi.nlm.nih.gov) αποτελεί ένα διακεκριμένο ολοκληρωμένο σύστημα ανάκτησης βιολογικών δεδομένων, το οποίο παρέχει στον χρήστη τη δυνατότητα για ταυτόχρονη αναζήτηση σε όλες τις ΒΔ που περιλαμβάνονται στο NCBI [10].

Μορφή Αρχείου Καταχώρησης

Η μορφή αρχείου καταχώρησης είναι μια συγκεκριμένη μορφή αποθήκευσης των καταχωρήσεων στις βάσεις βιολογικών δεδομένων. Ένα τυπικό παράδειγμα μορφής αρχείου είναι το αποκαλούμενο "FASTA format". Η συγκεκριμένη μορφή αποτελείται από μια επικεφαλίδα (header), η οποία αρχίζει με το σύμβολο «>». Η επικεφαλίδα περιλαμβάνει τον αριθμό καταχώρησης ή αριθμό πρόσβασης (accession number), και μια προαιρετική σύντομη περιγραφή της καταχώρησης, ακολουθούμενη από την κυρίως ακολουθία. Το σύμβολο «>» διαχωρίζει την επικεφαλίδα από την ακολουθία [37].

αριθμός πρόσβασης περιγραφή καταχώρησης

```
>gi|4503351|ref|NP_001370.1| DNA (cytosine-5)-methyltransferase 1
 isoform b [Homo sapiens]
MPARTAPARVPTLAVPAISLPDDVRRRLKDLERDSLTEKECVKEKLNLLHEFLQTEIKNQLCDLETKLRK
EELSEEGYLAKVKSLLNKDLSLENGAHAYNREVNGRLENGNQARSEARRVGMADANSPPKPLSKPRTPRR
SKSDGEAKPEPSPSPRITRKSTRQTTITSHFAKGPAKRKPQEESERAKSDESIKEEDK............
```

Εικόνα 11 Παράδειγμα αρχείου σε μορφή FASTA

Υπολογιστικές μέθοδοι στοίχισης - BLAST

Η αναζήτηση ομοιότητας στις βάσεις δεδομένων επιδιώκει τη βέλτιστη πιθανή στοίχιση των αλληλουχιών ανά ζεύγη. Συνήθως, για την αναζήτηση ομοιότητας στις βάσεις δεδομένων χρησιμοποιούνται ευρετικές μέθοδοι, οι οποίες επιδιώκουν τη γρήγορη και ευαίσθητη αναζήτηση των πιθανότερων στοιχίσεων χωρίς όμως να εγγυώνται την εύρεση της βέλτιστης στοίχισης.

Το πακέτο λογισμικού BLAST (Basic Local Alignment Search Tool, http://blast.ncbi.nlm.nih.gov/Blast.cgi) [38] αποτελεί το πλέον δημοφιλές εργαλείο για την εύρεση τοπικών ομοιοτήτων μεταξύ μιας ακολουθίας επερώτησης και όλων των αλληλουχιών που τη συγκεκριμένη στιγμή υπάρχουν στη βάση. Αρχικά, ο αλγόριθμος BLAST διαιρεί την ακολουθία επερώτησης σε επιμέρους ακολουθίες ή «λέξεις». Το τυπικό μήκος των λέξεων είναι 3 για τις πρωτεϊνικές και 11 για τις νουκλεοτιδικές ακολουθίες. Στη συνέχεια, ο αλγόριθμος εντοπίζει παρόμοιες λέξεις στη ΒΔ με βαθμολογία στοίχισης μεγαλύτερη από μια τιμή κατωφλιού. Τότε, η ευθυγράμμιση επεκτείνεται και προς τις δυο κατευθύνσεις. Η επέκταση συνεχίζεται όσο η βαθμολογία στοίχισης είναι μεγαλύτερη της τιμής κατωφλιού [38,39]. Κατά αυτόν τον τρόπο, προκύπτουν στοιχίσεις με τη μεγαλύτερη βαθμολογία για μια ακολουθία (HSPs, High Scoring Pairs). Τα αποτελέσματα της ανάλυσης με το πρόγραμμα BLAST περιλαμβάνουν μια λίστα με τις παρόμοιες ακολουθίες που βρέθηκαν στη βάση, τις στοιχιζόμενες περιοχές των αλληλουχιών και στατιστικά στοιχεία. Η τελική βαθμολογία της κάθε τοπικής στοίχισης είναι το άθροισμα των βαθμολογιών των HSPs. Η στατιστική σημαντικότητα μιας στοίχισης καθορίζεται από την E-value (Expectation value), η οποία υποδεικνύει την πιθανότητα οι στοιχίσεις να έχουν προκύψει τυχαία. Επομένως, μικρότερη E-value συνεπάγεται μεγαλύτερη πιθανότητα οι στοιχίσεις να μην έχουν προκύψει τυχαία. Η χρήση υψηλότερων E-value εξυπηρετεί την εύρεση στοιχίσεων μεταξύ εξελικτικά απομακρυσμένων αλληλουχιών [38,39].

Υπάρχουν πολλαπλά ειδικά προγράμματα BLAST (βλ. Πίνακα 1). Κάθε πρόγραμμα παρουσιάζει ιδιαίτερα πλεονεκτήματα ανάλογα με τον επιδιωκόμενο στόχο κάθε ανάλυσης.

Πίνακας 1 Ειδικά προγράμματα BLAST

Εργαλείο	Ακολουθία επερώτησης	Ακολουθία στη ΒΔ	Σχόλια
BLASTn	νουκλεοτίδια	νουκλεοτίδια	εύρεση συγγενικών αλληλουχιών
BLASTp	πρωτεΐνη	πρωτεΐνη	εύρεση απομακρυσμένων αλληλουχιών
tBLASTn	πρωτεΐνη	νουκλεοτίδια1	εύρεση εξωνίων
tBLASTx	νουκλεοτίδια1	νουκλεοτίδια1	εύρεση ESTs2

Το πρόγραμμα PSI-BLAST (Position Specific Iterated-BLAST) αποτελεί αναβαθμισμένη έκδοση του προγράμματος BLAST [40]. Το PSI-BLAST, εφαρμόζει μια επαναληπτική διαδικασία στοίχισης, κατά την οποία ένα προφίλ που δημιουργείται από σημαντικές στοιχίσεις στο πρώτο στάδιο της ανάλυσης χρησιμοποιείται ως επερώτηση στο δεύτερο στάδιο της ανάλυσης. Η διαδικασία αυτή επαναλαμβάνεται. Το πρόγραμμα PSI-BLAST είναι κατάλληλο για τον εντοπισμό ασθενών στοιχίσεων (π.χ. αυτών που βρίσκονται στη «ζώνη του λυκόφωτος»). Το πρόγραμμα αμφίδρομο BLAST (reciprocal BLAST), που είναι μια παραλλαγή του βασικού προγράμματος BLAST (Altschul et al., 1997), αναζητά τις αμφίδρομες βέλτιστες στοιχίσεις. Πρώτα, μια ακολουθία επερώτησης A χρησιμοποιείται για τον εντοπισμό της ακολουθίας B σε μια βάση δεδομένων. Στη συνέχεια, η B χρησιμοποιείται για αναζήτηση έναντι της ΒΔ που εντοπίστηκε η A. Εάν, το βέλτιστο HSP είναι η ακολουθία A, τότε οι ακολουθίες A και B θεωρούνται ομόλογες.

Βάσεις βιολογικών δεδομένων που χρησιμοποιήθηκαν

Η βιοπληροφορική συμβάλλει ουσιαστικά στην ανάπτυξη υπολογιστικών μεθόδων και εργαλείων για την οργάνωση και τη διαχείριση της ολοένα αυξανόμενης βιολογικής πληροφορίας καθώς και των νέων τύπων βιολογικών δεδομένων που διαρκώς παράγονται. Μέσα από αυτόν τον τεράστιο όγκο δεδομένων προβάλλει επιτακτική η ανάγκη για την αποδοτική αποθήκευση τους έτσι ώστε να είναι δυνατή η μελέτη τους με σκοπό την ερμηνεία τους και την εξαγωγή πολύτιμης πληροφορίας για την καλύτερη κατανόηση των βιολογικών φαινομένων, ιδίως από τους επιστήμονες υγείας.

Μία από τις σημαντικότερες ερευνητικές περιοχές της επιστήμης της βιοπληροφορικής είναι η ανάπτυξη των βάσεων βιολογικών δεδομένων που ήρθαν στο προσκήνιο για να καλύψουν ακριβώς αυτήν την ανάγκη. Οι βάσεις βιολογικών δεδομένων έχουν σήμερα εξαιρετική δυναμική και περιθώρια εφαρμογών. Κατά κύριο λόγο περιέχουν δεδομένα τα οποία παράγονται από διάφορα ερευνητικά προγράμματα, από επιστημονικά πειράματα ή ακόμη και δεδομένα που συλλέγουν οι ερευνητές από δημοσιευμένα επιστημονικά άρθρα, βιβλία και πρακτικά συνεδρίων. Η συλλογή αυτών των δεδομένων, η αξιολόγησή τους μέσω των τεχνικών εξόρυξης και η οπτικοποίηση (αναπαράσταση) της παραγόμενης γνώσης αποτελούν την κύρια πηγή πληροφορίας των επιστημόνων στην προσπάθεια τους να ερμηνεύσουν τις διάφορες βιολογικές διαδικασίες.

Σήμερα υπάρχει μια πληθώρα δημόσιων βάσεων βιολογικών δεδομένων οι οποίες είναι δωρεάν διαθέσιμες και περιέχουν έναν τεράστιο και διαφοροποιημένο όγκο δεδομένων. Συνήθως παρέχουν την δυνατότητα στους χρήστες να εκτελέσουν ένα ερώτημα (query) στην υποκείμενη συλλογή δεδομένων εφαρμόζοντας κάποια κριτήρια και επιστρέφουν το αντίστοιχο αποτέλεσμα.

Οι υπάρχουσες πρωτογενείς βάσεις βιολογικών δεδομένων παρέχουν σημαντικές πληροφορίες για τη γενετική οντολογία. Χρησιμοποιούνται για να κατανοηθεί η ροή της γενετικής πληροφορίας, τα χαρακτηριστικά των βιολογικών οντοτήτων και πιο πρακτικά για να συνδεθούν μεταξύ τους οι διάφοροι κωδικοί αναγνώρισης (identifiers) που έχουν δημιουργηθεί για να χαρακτηρίσουν τις βιολογικές οντότητες και τα βιολογικών δεδομένα των βάσεων. Οι βάσεις δεδομένων που χρησιμοποιήθηκαν και ενσωματώθηκαν στη μετα-βάση μας αναλύονται παρακάτω.

UniProt

Η βάση δεδομένων UniProt είναι μια ολοκληρωμένη πηγή που περιέχει πληροφορίες για την αλληλουχία πρωτεϊνών. Οι βάσεις δεδομένων της UniProt αποτελούνται από τη βάση UniProt Knowledgebase (UniProtKB), τη βάση UniProt Reference Clusters (UniRef) και τη βάση UniProt Archive (UniParc).

Η **UniProt** (Universal Protein Resource [41, 42], δημιουργήθηκε με κύριο στόχο να αποτελέσει την πηγή αναφοράς για την περιγραφή των πρωτεϊνών. Δημιουργήθηκε το 2002 από την ένωση των βάσεων Swiss-Prot, TrEMBL και PIR-PSD (Translated EMBL Nucleotide Sequence Data Library και Protein Information Resource Protein Sequence Database). Ένα από το πιο σημαντικά της χαρακτηριστικά είναι ότι διασταυρώνει τις πληροφορίες που καταγράφει με έναν μεγάλο αριθμό από άλλες βάσεις βιολογικών δεδομένων και έτσι μπορεί και προσφέρει πρακτικά όλη την επίσημη γνώση που έχουμε πάνω στις πρωτεΐνες.

Προκειμένου να περιορίσει τον πλεονασμό που υπάρχει σε νουκλεοτιδικές αλληλουχίες ή κωδικούς αναγνώρισης που αναφέρονται στην ίδια πρωτεΐνη, έχει προσπαθήσει να εντάξει κάτω από τον ίδιο UniProt κωδικό αναγνώρισης (UniProt ID) όλα τα παρόμοια πρωτεϊνικά παράγωγα. Κάθε ενεργός κωδικός ονομάζεται primary accession identifier, αλλά στην περίπτωση που δύο ομάδες πρωτεϊνικών παραγώγων χρειαστεί να ενωθούν, ένας από τους δύο υποβιβάζεται σε secondary. Μέσω της διαδικασίας αυτής, για κάθε γονίδιο ορίζεται μία τουλάχιστον χαρακτηριστική πρωτεϊνική αλληλουχία, η canonical sequence (κανονική ή αντιπροσωπευτική αλληλουχία), η οποία χρησιμοποιείται για να περιγράψει το σύνολο των πρωτεϊνικών παραγώγων του. Σε περίπτωση που υπάρχει αβεβαιότητα για τις πρωτεϊνικές ισομορφές ενός γονιδίου ή οι διαφορές ανάμεσα τους είναι ιδιαίτερα σημαντικές και δεν μπορεί να αποδοθεί ένα μοναδικό canonical sequence και Uniprot ID, τότε

ορίζονται περισσότερα του ενός. Είναι επίσης δυνατό να συμβεί και το αντίστροφο.

Τα δεδομένα της Uniprot χωρίζονται γενικά σε δύο κατηγορίες, τα reviewed (επιβεβαιωμένα) – αυτά που προέρχονται από την βάση Swiss-Prot και περιέχουν περισσότερη πληροφορία καθώς έχουν καταγραφεί με ανθρώπινη επιμέλεια και τα unreviewed (μη επιβεβαιωμένα) – αυτά που δίνονται από την TrEMBL που συλλέγει πληροφορίες αυτόματα. Η σχέση ανάμεσα στις δύο βάσεις είναι δυναμική και πληροφορίες που αρχικά περιέχονται στην TrEMBL, αφού περάσουν από τον έλεγχο ενός επιμελητή, μεταφέρονται στην Swiss-Prot ή και το αντίστροφο καθώς κάποια εγγραφή μπορεί να χρειάζεται να ξαναπεράσει από έλεγχο. Στη βάση RGDtrip, χρησιμοποιήθηκαν αποκλειστικά στοιχεία της Swiss-Prot, αν και λειτουργικά η επέκταση σε οποιοδήποτε σύνολο πρωτεϊνών είναι εύκολη, καθώς επίσης χρησιμοποιήθηκε το reviewed σύνολο από UniProt IDs.

Η UniProt ενημερώνεται κάθε τέσσερις εβδομάδες. Υπάρχει δυνατότητα λήψης μικρών συνόλων δεδομένων και υποσυνόλων απευθείας από τον ιστότοπο ακολουθώντας τη σύνδεση λήψης σε οποιαδήποτε σελίδα αποτελεσμάτων αναζήτησης.

Protein Data Bank (PDB)

Η Protein Data Bank (PDB) είναι το μοναδικό παγκόσμιο αποθετήριο πληροφοριών για τις τρισδιάστατες δομές μεγάλων βιολογικών μορίων, συμπεριλαμβανομένων των πρωτεϊνών και των νουκλεϊνικών οξέων [20, 43]. Αυτά είναι τα μόρια της ζωής που βρίσκονται σε όλους τους οργανισμούς, συμπεριλαμβανομένων των βακτηριδίων, ζυμών, φυτών, άλλων ζώων και των ανθρώπων. Η κατανόηση του σχήματος ενός μορίου συνάγει το ρόλο μιας δομής στην ανθρώπινη υγεία και ασθένεια και στην ανάπτυξη φαρμάκων. Οι

δομές στο φάσμα των αρχείων κυμαίνονται από μικροσκοπικές πρωτεΐνες και κομμάτια DNA έως πολύπλοκες μοριακές μηχανές όπως το ριβόσωμα.

Τα δεδομένα της βάσης διατίθενται δωρεάν στους χρήστες, ενώ ενημερώνεται κάθε εβδομάδα.

Η PDB δημιουργήθηκε το 1971 στο Εθνικό Εργαστήριο Brookhaven υπό την ηγεσία του Walter Hamilton και αρχικά περιείχε 7 δομές. Μετά τον πρόωρο θάνατο του Χάμιλτον, ο Tom Koetzle άρχισε να ηγείται της PDB το 1973 και έπειτα ο Joel Sussman το 1994. Με επικεφαλής την Helen M. Berman, το Research Collaboratory for Structural Bioinformatics (RCSB) ανέλαβε ην ευθύνη για τη διαχείριση της PDB το 1998. Το 2003, το wwPDB δημιουργήθηκε για να διατηρήσει ένα ενιαίο αρχείο PDB μακρομοριακών δομικών δεδομένων που είναι ελεύθερα και δημοσίως διαθέσιμο στην παγκόσμια κοινότητα. Αποτελείται από οργανισμούς που δρουν ως αποθετήρια, επεξεργασίας δεδομένων και κέντρα διανομής δεδομένων της PDB.

Επιπλέον, η RCSB-PDB υποστηρίζει μια ιστοσελίδα όπου οι επισκέπτες μπορούν να εκτελούν απλά και περίπλοκα ερωτήματα στα δεδομένα και να αναλύουν και να απεικονίζουν τα αποτελέσματα.

CATH-Gene3D

Η βάση δεδομένων CATH είναι ένας δωρεάν δημόσιος διαθέσιμος ηλεκτρονικός πόρος που παρέχει πληροφορίες σχετικά με τις εξελικτικές σχέσεις των τομέων πρωτεϊνών. Δημιουργήθηκε στα μέσα της δεκαετίας του 1990 από την καθηγήτρια Christine Orengo και συνεργάτες της και συνεχίζει να αναπτύσσεται από τον όμιλο Orengo στο University College του Λονδίνου [44].

Πειραματικά καθορισμένες πρωτεϊνικές τρισδιάστατες δομές λαμβάνονται από τη βάση δεδομένων PDB και διασπώνται στις διαδοχικές πολυπεπτιδικές τους αλυσίδες, όπου υπάρχει δυνατότητα. Οι περιοχές

πρωτεΐνης ταυτοποιούνται εντός αυτών των αλυσίδων με τη χρήση μίγματος αυτόματων μεθόδων και χειρωνακτικής επεξεργασίας. Οι τομείς στη συνέχεια ταξινομούνται στην δομική ιεραρχία CATH:

α. στο επίπεδο Class (C), οι τομείς τοποθετούνται σύμφωνα με το περιεχόμενο της δευτερογενούς δομής τους (δηλαδή, όλα τα άλφα, όλα τα βήτα, ένα μείγμα άλφα και βήτα ή μικρή δευτερεύουσα δομή),

β. στο επίπεδο της Αρχιτεκτονικής (A), χρησιμοποιούνται πληροφορίες από το γενικό σχήμα τους όπως καθορίζεται από τους προσανατολισμούς των δευτερογενών δομών στον τρισδιάστατο χώρο, αγνοώντας τη συνδεσιμότητα μεταξύ τους,

γ. στο επίπεδο τοπολογίας / πτυχώσεων (T) οι δομές ομαδοποιούνται σε ομάδες δίπλωσης, ανάλογα με το συνολικό σχήμα και τη συνδεσιμότητα των δευτερογενών δομών και

δ. στο επίπεδο ομόλογης υπεροικογένειας (H) οι αναθέσεις γίνονται αν υπάρχουν καλά αποδεικτικά στοιχεία ότι οι τομείς σχετίζονται με εξέλιξη, δηλαδή αν είναι ομόλογοι.

Επιπρόσθετα δεδομένα αλληλουχίας για τομείς χωρίς πειραματικά καθορισμένες δομές παρέχονται από τον πόρο της Gene3D, που χρησιμοποιούνται για την κάλυψη των ομόλογων υπεροικογενειών. Οι πρωτεϊνικές αλληλουχίες από UniProtKB και Ensembl σαρώνονται έναντι CATH HMMs για να προβλέψουν τα όρια αλληλουχίας περιοχών και να κάνουν ομόλογες αναθέσεις υπεροικογένειας.

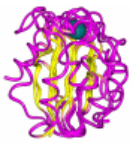
Για κάθε δεδομένη δομή που ταξινομείται στη βάση δεδομένων, η CATH δίνει πληροφορίες σχετικά με τη δομή και τη λειτουργία αυτής της πρωτεΐνης. Οι εξελικτικές σχέσεις που αφορούν τη δομή ενδιαφέροντος και άλλες πρωτεΐνες στη βάση δεδομένων μπορούν επίσης να προσδιοριστούν.

Η CATH δίνει επίσης μια συνολική άποψη του γνωστού συμπλέγματος δομής πρωτεΐνης μέχρι σήμερα. Για παράδειγμα, μπορείτε να βρείτε ποιες πτυχές και υπεροικογένειες είναι οι πιο δημοφιλείς και ποιες δομές είναι σπάνιες στη φύση.

SUPfam

Superfamily 1.75

HMM library and genome assignments server



Το SUPERFAMILY είναι μια βάση δεδομένων με διαρθρωτικό και λειτουργικό σχολιασμό για όλες τις πρωτεΐνες και τα γονιδιώματα [45].

Ο σχολιασμός SUPERFAMILY βασίζεται σε μια συλλογή κρυφών μοντέλων Markov, τα οποία αντιπροσωπεύουν δομικές πρωτεϊνικές περιοχές στο επίπεδο της υπεροικογένειας (Structural Classification of Proteins, SCOP). Μια υπεροικογένεια συγκεντρώνει τομείς που έχουν μια εξελικτική σχέση. Ο σχολιασμός παράγεται με σάρωση αλληλουχιών πρωτεϊνών από πάνω από 2.478 γονιδιώματα με πλήρη αλληλουχία έναντι των κρυφών μοντέλων Markov.

Για κάθε πρωτεΐνη μπορεί να:

- Υποβληθούν ακολουθίες για ταξινόμηση SCOP
- Προβληθεί η οργάνωση τομέα, ευθυγράμμιση αλληλουχιών και λεπτομερειών ακολουθίας πρωτεϊνών

Για κάθε γονιδίωμα μπορεί να:

- Εξεταστούν οι αναθέσεις υπερ-οικογένειας, φυλογενετικά δέντρα, λίστες οργάνωσης τομέα και δίκτυα
- Ελεγχθούν υπερεμφανίσεις υπερ- και υπο- εκπροσωπούμενες εντός ενός γονιδιώματος

Για κάθε υπεροικογένεια μπορεί να:

- Επιθεωρηθεί η ταξινόμηση SCOP, ο λειτουργικός σχολιασμός, η σχολιασμένη οντολογία γονιδίων, οι αναθέσεις αφαίρεσης και γονιδιώματος InterPro

- Εξερευνηθεί η ταξινομική κατανομή μιας υπεροικογένειας στο δέντρο της ζωής

Όλοι οι σχολιασμοί, τα μοντέλα και η χωρητικότητα βάσης δεδομένων είναι ελεύθερα διαθέσιμα για λήψη από όλους.

Το SUPERFAMILY είναι μέλος της κοινοπραξίας των βάσεων δεδομένων σχολιασμών πρωτεϊνών της InterPro και έχει ενσωματωθεί στο έργο του ευκαρυωτικού γονιδιώματος του Ensembl και στον πόρο πληροφοριών του Arabidopsis. Μέχρι σήμερα, οι δημοσιεύσεις SUPERFAMILY έχουν αναφερθεί πάνω από 1.000 φορές. Το SUPERFAMILY έχει χρησιμοποιηθεί σε δομικά, λειτουργικά, εξελικτικά και φυλογενετικά ερευνητικά έργα.

Pfam

Η έκδοση Pfam 31.0 δημιουργήθηκε από το Ευρωπαϊκό Ινστιτούτο Βιοπληροφορικής χρησιμοποιώντας μια βάση δεδομένων ακολουθίας που ονομάζεται Pfamseq, η οποία βασίζεται στην έκδοση UniProt 2016_10 [47].

Η βάση δεδομένων Pfam είναι ελεύθερα διαθέσιμη υπό την άδεια Creative Commons Zero ("CC0"). Τροφοδοτείται από το πακέτο HMMER3, το οποίο έγραψε ο Sean Eddy και η ομάδα του στο HHMI / Πανεπιστήμιο του Χάρβαρντ και χτίστηκε από την κοινοπραξία Xfam.

Οι πρωτεΐνες γενικά αποτελούνται από μία ή περισσότερες λειτουργικές περιοχές, κοινώς ονομαζόμενες περιοχές. Η παρουσία διαφορετικών περιοχών σε διάφορους συνδυασμούς σε διαφορετικές πρωτεΐνες δημιουργεί το ποικίλο ρεπερτόριο πρωτεϊνών που βρίσκονται στη φύση. Ο εντοπισμός των περιοχών που υπάρχουν σε μια πρωτεΐνη μπορεί να δώσει πληροφορίες για τη λειτουργία αυτής της πρωτεΐνης.

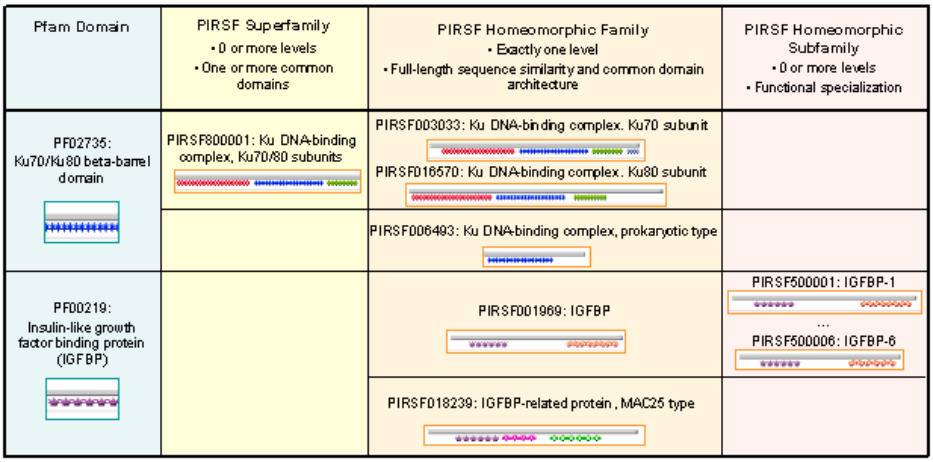
Η βάση δεδομένων Pfam είναι μια μεγάλη συλλογή από οικογένειες τομέων πρωτεϊνών. Κάθε οικογένεια αντιπροσωπεύεται από πολλαπλές ευθυγραμμίσεις αλληλουχιών και ένα κρυφό μοντέλο Markov (HMMs). Κάθε οικογένεια Pfam, που συχνά αναφέρεται ως είσοδος Pfam-A, αποτελείται από μια επιμελημένη ευθυγράμμιση των σπόρων που περιέχει ένα μικρό σύνολο αντιπροσωπευτικών μελών της οικογένειας, προφίλ κρυμμένα μοντέλα Markov (προφίλ HMM) που κατασκευάστηκαν από την ευθυγράμμιση των σπόρων και ένα αυτόματα παραγόμενο πλήρες ευθυγράμμιση, η οποία περιέχει όλες τις ανιχνεύσιμες αλληλουχίες πρωτεΐνης που ανήκουν στην οικογένεια, όπως ορίζεται από τις αναζητήσεις προφίλ HMM των βάσεων δεδομένων πρωτεύουσας αλληλουχίας. Οι καταχωρίσεις Pfam ταξινομούνται με έναν από τους έξι τρόπους:

- Οικογένεια: Μια συλλογή συναφών πρωτεϊνικών περιοχών
- Τομέας: Μια δομική μονάδα
- Επανάληψη: Μια σύντομη μονάδα που είναι ασταθής μεμονωμένα, αλλά σχηματίζει σταθερή δομή όταν υπάρχουν πολλαπλά αντίγραφα
- Μοτίβα: Μια σύντομη μονάδα που βρίσκεται έξω από τους σφαιρικούς τομείς
- Σπειροειδής πηνίο: Περιφέρειες που περιέχουν κατά κύριο λόγο μοτίβα σπειροειδούς πηνίου, περιοχές που τυπικά περιέχουν αλφα-έλικες που είναι συσπειρωμένες μαζί σε δεσμίδες 2-7.
- Ανακατεμένη: Περιοχές που είναι διατηρημένες, όμως είτε παρουσιάζονται είτε προβλέπεται να περιέχουν σύνθεση αλληλουχίας μεροληψίας και / ή είναι ενδογενώς διαταραγμένες (μη σφαιρικοί).

Οι σχετικές καταχωρήσεις του Pfam ομαδοποιούνται σε ομάδες. η σχέση μπορεί να οριστεί από την ομοιότητα της ακολουθίας, της δομής ή του προφίλ-HMM.

PIRSF

Η ιδέα PIRSF χρησιμοποιείται ως κατευθυντήρια αρχή για την παροχή ολοκληρωμένης και μη επικαλυπτόμενης ομαδοποίησης των αλληλουχιών UniProtKB σε μια ιεραρχική σειρά που να αντικατοπτρίζει τις εξελικτικές σχέσεις τους. Το σύστημα ταξινόμησης PIRSF βασίζεται σε ολόκληρες πρωτεΐνες και όχι σε τομείς συνιστωσών. Επομένως, επιτρέπει τη σχολιασμό γενικών βιοχημικών και ειδικών βιολογικών λειτουργιών, καθώς και την ταξινόμηση πρωτεϊνών χωρίς σαφώς καθορισμένους τομείς [47, 48].



Εικόνα 12 Παραδείγματα των επιπέδων ταξινόμησης PIRSF

Το πρωτεύον επίπεδο είναι η ομοιομορφική οικογένεια, τα μέλη της οποίας είναι τόσο ομόλογα (εξελίχθηκαν από έναν κοινό πρόγονο) όσο και ομοιομορφικά (μοιράζοντας ομοιότητα αλληλουχιών πλήρους μήκους και μια κοινή αρχιτεκτονική τομέων). Σε χαμηλότερο επίπεδο είναι οι υποοικογένειες που είναι ομάδες που αντιπροσωπεύουν λειτουργική εξειδίκευση ή / και διακύμανση αρχιτεκτονικής τομέα εντός της οικογένειας. Πάνω από το ομοιομορφικό επίπεδο μπορεί να υπάρχουν γονικές υπεροικογένειες που συνδέουν απομακρυσμένες συγγενείς οικογένειες και ορφανές πρωτεΐνες με

βάση κοινές περιοχές. Επειδή οι πρωτεΐνες μπορούν να ανήκουν σε περισσότερες από μία υπεροικογενείς περιοχές, η δομή PIRSF είναι τυπικά ένα δίκτυο [49].

Ως μέρος της κοινοπραξίας UniProt, η PIR ανέπτυξε αυτή τη στρατηγική ταξινόμησης, με κανόνες για το λειτουργικό σημείο και το όνομα της πρωτεΐνης, για να βοηθήσει στη διάδοση και τυποποίηση της πρωτεϊνικής σχολιασμού και στη συστηματική ανίχνευση σφαλμάτων σχολιασμού. Με αυτό τον τρόπο, το PIRSF βελτιώνει την ευαισθησία της ταυτοποίησης πρωτεϊνών και των λειτουργικών συμπερασμάτων και παρέχει επίσης τη βάση για την εξελικτική και συγκριτική έρευνα γονιδιωματικής.

Οι οικογένειες PIRSF επιμελούνται χρησιμοποιώντας μια υποδομή βιοπληροφορικής που υλοποιείται σε πλαίσιο J2EE.

InterPro

Το InterPro είναι ένας πόρος που παρέχει λειτουργική ανάλυση αλληλουχιών πρωτεϊνών, ταξινομώντας τα σε οικογένειες και προβλέποντας την παρουσία τομέων και σημαντικών περιοχών. Για να ταξινομεί τις πρωτεΐνες με αυτόν τον τρόπο, η InterPro χρησιμοποιεί προγνωστικά μοντέλα, γνωστά ως υπογραφές, που παρέχονται από διάφορες διαφορετικές βάσεις δεδομένων (που αναφέρονται ως βάσεις δεδομένων μελών) που αποτελούν την κοινοπραξία InterPro [50, 51].

Το InterProScan είναι το πακέτο λογισμικού που επιτρέπει τη σάρωση αλληλουχιών έναντι των υπογραφών της InterPro [52].

Το InterPro συνδυάζει τις υπογραφές από πολλαπλές διαφορετικές βάσεις δεδομένων σε έναν μοναδικό πόρο αναζήτησης, μειώνοντας την απόλυση και βοηθώντας τους χρήστες να ερμηνεύσουν τα αποτελέσματα της ανάλυσης

αλληλουχίας τους. Συνδυάζοντας τις βάσεις δεδομένων μελών, η InterPro αξιοποιεί τα μεμονωμένα πλεονεκτήματα, δημιουργώντας ένα ισχυρό εργαλείο διάγνωσης και έναν ολοκληρωμένο πόρο.

Χρησιμοποιείται από ερευνητές που ενδιαφέρονται για την ανάλυση σε μεγάλη κλίμακα ολόκληρων πρωτεόνομων, γονιδιωμάτων και μεταγονιδιωμάτων, καθώς και ερευνητές που επιδιώκουν να χαρακτηρίσουν μεμονωμένες αλληλουχίες πρωτεϊνών. Μέσα στο EBI, το InterPro χρησιμοποιείται για να βοηθά στο σχολιασμό των αλληλουχιών πρωτεϊνών στο UniProtKB. Χρησιμοποιείται επίσης από την ομάδα σχολιασμού οντολογίας γονιδίων για την αυτόματη εκχώρηση των όρων γονιδιακής οντολογίας σε αλληλουχίες πρωτεϊνών.

Το InterPro ενημερώνεται περίπου κάθε 8 εβδομάδες. Οι σελίδες των διαφορετικών εκδόσεων περιέχουν πληροφορίες σχετικά με το τι έχει αλλάξει σε κάθε ενημέρωση.

Αναφορές – Βιβλιογραφία κεφαλαίου

1. Agnar Aamodt and Mads Nygård. (1995). Different roles and mutual dependencies of data, information, and knowledge-an AI perspective on their integration. Data Knowl. Eng. 16, 3 (October 1995), 191-222. doi:10.1016/0169-023X(95)00017-M
2. Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. The Knowledge Engineering Review, 11(2), 93-136. doi:10.1017/S0269888900007797
3. E. F. Codd. (1970). A relational model of data for large shared data banks. Commun. ACM 13, 6 (June 1970), 377-387. doi:10.1145/362384.362685
4. L. Zhao, S. A. Roberts. (1988). An Object-Oriented Data Model for Database Modelling, Implementation and Access, The Computer Journal, Volume 31, Issue 2, 1 January 1988, Pages 116-124, doi:10.1093/comjnl/31.2.116
5. Donald D. Chamberlin and Raymond F. Boyce. (1974). SEQUEL: A structured English query language. In Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control (SIGFIDET '74). ACM, New York, NY, USA, 249-264. doi:10.1145/800296.811515
6. https://www.w3schools.com/xml
7. N. Gautham. (2006). Bioinformatics: Databases and Algorithms. Alpha Science International, Ltd. ISBN:1842653008
8. Atlamazoglou, Vassilis & Thireou, Trias & Hamodrakas, Yannis & Spyrou, George. (2006). MetaBasis: A web-based database containing metadata on software tools and databases in the field of bioinformatics. Applied bioinformatics. 5. 187-92. doi:10.2165/00822942-200605030-00007
9. Chapman, D. (2009). Health-Related Databases. Journal of the Canadian Academy of Child and Adolescent Psychiatry, 18(2), 148-149.
10. Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Chetvernin, V., ... Yaschenko, E. (2008). Database resources of the National Center for Biotechnology Information. Nucleic Acids Research, 36(Database issue), D13-D21. doi:10.1093/nar/gkm1000
11. https://www.ncbi.nlm.nih.gov/pubmed
12. https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi
13. Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., & Sayers, E. W. (2009). GenBank. Nucleic Acids Research, 37(Database issue), D26-D31. doi:10.1093/nar/gkn723

14. Kulikova, T., Akhtar, R., Aldebert, P., Althorpe, N., Andersson, M., Baldwin, A., ... Apweiler, R. (2007). EMBL Nucleotide Sequence Database in 2006. Nucleic Acids Research, 35(Database issue), D16-D20. doi:10.1093/nar/gkl913
15. Sugawara, H., Ogasawara, O., Okubo, K., Gojobori, T., & Tateno, Y. (2008). DDBJ with new system and face. Nucleic Acids Research, 36(Database issue), D22-D24. doi:10.1093/nar/gkm889
16. http://www.insdc.org
17. Hubbard, T. J. P., Aken, B. L., Ayling, S., Ballester, B., Beal, K., Bragin, E., ... Flicek, P. (2009). Ensembl 2009. Nucleic Acids Research, 37(Database issue), D690-D697. doi:10.1093/nar/gkn828
18. https://www.ensembl.org/info/about/index.html
19. https://www.ncbi.nlm.nih.gov/genome
20. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. (2000) The Protein Data Bank. Nucleic Acids Research, 28, 235-242. doi:10.1093/nar/28.1.235
21. Pearl, F., Todd, A., Sillitoe, I., Dibley, M., Redfern, O., Lewis, T., ... Orengo, C. (2005). The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. Nucleic Acids Research, 33(Database Issue), D247-D251. doi:10.1093/nar/gki024
22. Andreeva, A., Howorth, D., Brenner, S. E., Hubbard, T. J. P., Chothia, C., & Murzin, A. G. (2004). SCOP database in 2004: refinements integrate structure and sequence family data. Nucleic Acids Research, 32(Database issue), D226-D229. doi:10.1093/nar/gkh039
23. Markowitz, V. M., Chen, I.-M. A., Palaniappan, K., Chu, K., Szeto, E., Grechkin, Y., ... Kyrpides, N. C. (2010). The integrated microbial genomes system: an expanding comparative analysis resource. Nucleic Acids Research, 38(Database issue), D382-D390. doi:10.1093/nar/gkp887
24. Wu, C. H., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z.-Z., ... Barker, W. C. (2002). The Protein Information Resource: an integrated public resource of functional annotation of proteins. Nucleic Acids Research, 30(1), 35-37. doi:10.1093/nar/30.1.35
25. Bairoch, A., & Apweiler, R. (2000). The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Research, 28(1), 45-48. doi:10.1093/nar/28.1.45
26. Bairoch, A., Apweiler, R., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., ... Yeh, L.-S. L. (2005). The Universal Protein Resource (UniProt). Nucleic Acids Research, 33(Database Issue), D154-D159. doi:10.1093/nar/gki070

27. Pruitt, K. D., Tatusova, T., & Maglott, D. R. (2007). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Research, 35(Database issue), D61-D65. doi:10.1093/nar/gkl842
28. Holland, T. A., Veretnik, S., Shindyalov, I. N., Bourne P. E. (2006). Partitioning Protein Structures into Domains: Why Is it so Difficult?. Journal of Molecular Biology, Volume 361, Issue 3, 562-590, ISSN 0022-2836. doi:10.1016/j.jmb.2006.05.060.
29. Ponting, C. P., Russell, R. R. (2002). The natural history of protein domains. Annual Review of Biophysics and Biomolecular Structure, Vol.31, 45-71. doi:10.1146/annurev.biophys.31.082901.134314
30. Wu, C. H., Huang, H., Yeh, L.-S. L., Barker W. C. (2003). Protein family classification and functional annotation. Computational Biology and Chemistry, Vol.27(1), 37-47. ISSN 1476-9271. doi:10.1016/S1476-9271(02)00098-1.
31. Gilbert, D. G. (2002). euGenes: a eukaryote genome information system. Nucleic Acids Research, 30(1), 145-148. doi:10.1093/nar/30.1.145
32. Safran, M., Dalah, I., Alexander, J., Rosen, N., Iny Stein, T., Shmoish, M., ... Lancet, D. (2010). GeneCards Version 3: the human gene integrator. Database: The Journal of Biological Databases and Curation, 2010, baq020. doi:10.1093/database/baq020
33. http://www.geneontology.org
34. The Gene Ontology Consortium, Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., ... Sherlock, G. (2000). Gene Ontology: tool for the unification of biology. Nature Genetics, 25(1), 25-29. doi:10.1038/75556
35. The Gene Ontology Consortium. (2008). The Gene Ontology project in 2008. Nucleic Acids Research, 36(Database issue), D440-D444. doi:10.1093/nar/gkm883
36. Consortium, T. G. O. (2001). Creating the Gene Ontology Resource: Design and Implementation. Genome Research, 11(8), 1425-1433. doi:10.1101/gr.180801
37. Westhead, D., Oarish, J. H., Twyman, R. M. (2002) Bioinformatics: BIOS Scientific Publishers Ltd., ISBN 13: 9781859962725.
38. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. (1990) Basic local alignment search tool. J Mol Biol 215: 403-410. doi:10.1016/S0022-2836(05)80360-2
39. Pertsemlidis, A., & Fondon, J. W. (2001). Having a BLAST with bioinformatics (and avoiding BLASTphemy). Genome Biology, 2(10), reviews2002.1-reviews2002.10. doi:10.1186/gb-2001-2-10-reviews2002
40. Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 25(17), 3389-3402. doi:10.1093/nar/25.17.3389

41. http://www.uniprot.org
42. Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., ... Yeh, L.-S. L. (2004). UniProt: the Universal Protein knowledgebase. Nucleic Acids Research, 32(Database issue), D115-D119. doi:10.1093/nar/gkh131
43. https://www.rcsb.org
44. http://www.cathdb.info
45. http://supfam.org/SUPERFAMILY
46. Wilson, D., Pethica, R., Zhou, Y., Talbot, C., Vogel, C., Madera, M., ... Gough, J. (2009). SUPERFAMILY—sophisticated comparative genomics, data mining, visualization and phylogeny. Nucleic Acids Research, 37(Database issue), D380-D386. doi:10.1093/nar/gkn762
47. https://pfam.xfam.org
48. Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., ... Bateman, A. (2016). The Pfam protein families database: towards a more sustainable future. Nucleic Acids Research, 44(Database issue), D279-D285. doi:10.1093/nar/gkv1344
49. Wu, C. H., Nikolskaya, A., Huang, H., Yeh, L.-S. L., Natale, D. A., Vinayaka, C. R., ... Barker, W. C. (2004). PIRSF: family classification system at the Protein Information Resource. Nucleic Acids Research, 32(Database issue), D112-D114. doi:10.1093/nar/gkh097
50. https://www.ebi.ac.uk/interpro
51. Finn, R. D., Attwood, T. K., Babbitt, P. C., Bateman, A., Bork, P., Bridge, A. J., ... Mitchell, A. L. (2017). InterPro in 2017—beyond protein family and domain annotations. Nucleic Acids Research, 45(Database issue), D190-D199. doi:10.1093/nar/gkw1107
52. Jones, P., Binns, D., Chang, H.-Y., Fraser, M., Li, W., McAnulla, C., ... Hunter, S. (2014). InterProScan 5: genome-scale protein function classification. Bioinformatics, 30(9), 1236-1240. doi:10.1093/bioinformatics/btu031

Η πρόσβαση στις ηλεκτρονικές πηγές - αναφορές επικαιροποιήθηκε τον Μάρτιο 2018.

Κεφάλαιο 2ο
Οπτικοποίηση δεδομένων

Γενικά

Η οπτικοποίηση δεδομένων είναι η διαδικασία της αναπαράστασης δεδομένων με έναν οπτικό και ουσιαστικό τρόπο, έτσι ώστε ο χρήστης να μπορεί να τα κατανοήσει καλύτερα. Η απεικόνιση των δεδομένων σε ένα γράφημα, επιτρέπει στους χρήστες να αντλούν στοιχεία από αφηρημένα δεδομένα με ικανοποιητικό και αποτελεσματικό τρόπο.

Η διαδικασία οπτικοποίησης δεδομένων αρχίζει συνήθως με την κατανόηση των αναγκών της ομάδας ενδιαφερομένων. Η ποιοτική έρευνα (π.χ. συνεντεύξεις χρηστών-ερευνητών) μπορεί να αποκαλύψει πώς, πότε και πού θα χρησιμοποιηθεί η απεικόνιση. Λαμβάνοντας αυτές τις πληροφορίες, μπορεί να καθοριστεί ποια μορφή οργάνωσης δεδομένων είναι απαραίτητη για την επίτευξη των στόχων των χρηστών. Μόλις οι πληροφορίες οργανωθούν με τρόπο που βοηθά τους χρήστες να τις κατανοήσουν καλύτερα - και τους βοηθά να επιτύχουν τους στόχους τους - οι τεχνικές απεικόνισης είναι τα επόμενα εργαλεία που θα εφαρμοστούν στα δεδομένα. Πλήθος από οπτικά στοιχεία (π.χ. χάρτες και γραφήματα) μαζί με τις κατάλληλες ετικέτες καθώς και άλλες οπτικές παράμετροι, όπως το χρώμα, η αντίθεση, η απόσταση και το μέγεθος, χρησιμοποιούνται για τη δημιουργία μιας κατάλληλης οπτικής ιεραρχίας και μιας οπτικής διαδρομής μέσα στα δεδομένα.

Η οπτικοποίηση των δεδομένων γίνεται όλο και πιο διαδραστική, ειδικά όταν χρησιμοποιείται σε έναν ιστότοπο ή μια εφαρμογή. Η διαδραστικότητα

επιτρέπει τη χειραγώγηση της οπτικοποίησης από τους χρήστες, καθιστώντας την εξαιρετικά αποτελεσματική στην κάλυψη των αναγκών τους. Με τη διαδραστική απεικόνιση των πληροφοριών, οι χρήστες έχουν τη δυνατότητα να προβάλλουν τα δεδομένα από διαφορετικές οπτικές γωνίες και να χειρίζονται τις απεικονίσεις τους μέχρι να φτάσουν στις επιθυμητές απαντήσεις [1].

Γιατί είναι απαραίτητη η απεικόνιση δεδομένων

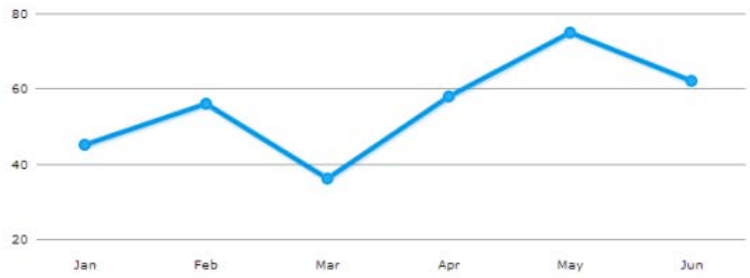
Η απεικόνιση δεδομένων, είναι μία από τις πιο πρωτόγονες μορφές επικοινωνίας που είναι γνωστή στον άνθρωπο. Έχει την προέλευσή της σε σχέδια σπηλαίου που χρονολογούνται ήδη από το 30.000 π.Χ., ακόμη και πριν από τη γραπτή επικοινωνία, η οποία εμφανίστηκε το 3.000 π.Χ. Η εικόνα είναι ένας από τους σημαντικότερους τρόπους για την επικοινωνία και μετάδοση των πληροφοριών.

Με την πάροδο του χρόνου, βρέθηκαν νέοι τρόποι για την απεικόνιση των πληροφοριών. Σήμερα, υπάρχει εξοικείωση και είναι πολύ διαδεδομένοι, βασικοί τύποι οπτικής αναπαράστασης δεδομένων, όπως το γραμμικό διάγραμμα, το ραβδόγραμμα και το διάγραμμα πίτας.

Μια οπτική αναπαράσταση, μπορεί να μεταδώσει περισσότερες πληροφορίες από έναν πίνακα με αριθμούς σε πολύ μικρότερο χώρο. Αυτό το χαρακτηριστικό των γραφικών τα καθιστά πιο αποτελεσματικά από τους πίνακες για την παρουσίαση δεδομένων.

Στο παρακάτω παράδειγμα είναι ξεκάθαρο, ποιος από τους δύο τρόπους (πίνακας ή διάγραμμα) είναι περισσότερο εύκολος και σύντομος για να εντοπιστεί ο μήνας με τις υψηλότερες πωλήσεις.

Month	Jan	Feb	Mar	Apr	May	Jun
Sales	45	56	36	58	75	62



Εικόνα 13 Αναπαράσταση δεδομένων με πίνακα και διάγραμμα

Μέχρι σήμερα, η οπτικοποίηση δεδομένων έχει χρησιμοποιηθεί ευρέως από τους επιστήμονες για την αναπαράσταση των δεδομένων χρησιμοποιώντας απλά γραφήματα (όπως τα ραβδογράμματα, οι πίτες κλπ.). Σήμερα, με την ανάπτυξη νέων τεχνολογιών οπτικοποίησης, αυτά μπορούν να συνδυαστούν με δυναμικές εφαρμογές και να επεξεργαστούν μεγάλο όγκο δεδομένων με αποτέλεσμα σύγχρονες εφαρμογές που ενσωματώνουν δυνατότητες κίνησης και επιτρέπουν την αλληλεπίδραση με το χρήστη [2].

Η οπτική αντίληψη στην οπτικοποίηση δεδομένων

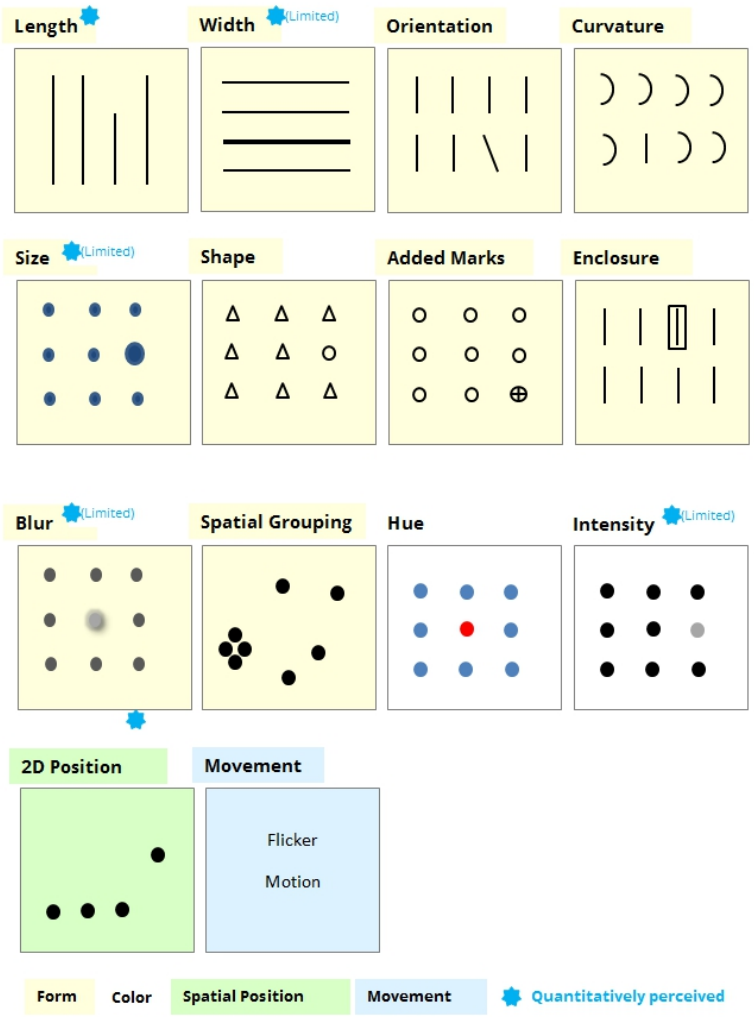
Ο John Tukey, Αμερικανός μαθηματικός και στατιστικός, είπε: "Η μεγαλύτερη αξία μιας εικόνας είναι όταν μας αναγκάζει να παρατηρήσουμε αυτό που ποτέ δεν περίμενε κανείς να βλέπει". Με την πρόσβαση στις προφανείς ιδιότητες των απεικονίσεων μπορεί να δοθεί δυνατότητα στους χρήστες να βρουν αυτό που ποτέ δεν περίμεναν να δουν.

Ο Colin Ware, Διευθυντής του Εργαστηρίου Ερευνών Οπτικοποίησης Δεδομένων στο Πανεπιστήμιο του Νιου Χαμσάιρ, στο βιβλίο του "Οπτικοποίηση πληροφοριών: Αντίληψη για το σχεδιασμό" ονομάζει τα βασικά δομικά στοιχεία κατά τη λειτουργία της οπτικής αντίληψης ως **Preattentive attributes** (χαρακτηριστικά συνέγερσης) [3]. Αυτά τα χαρακτηριστικά είναι αυτά που αμέσως "τραβούν την προσοχή", όταν ο χρήστης βλέπει μια εικόνα. Μπορούν

να γίνουν αντιληπτά σε λιγότερο από 10 χιλιοστά του δευτερολέπτου, ακόμη και πριν καταβάλει συνειδητή προσπάθεια να τα παρατηρήσει.

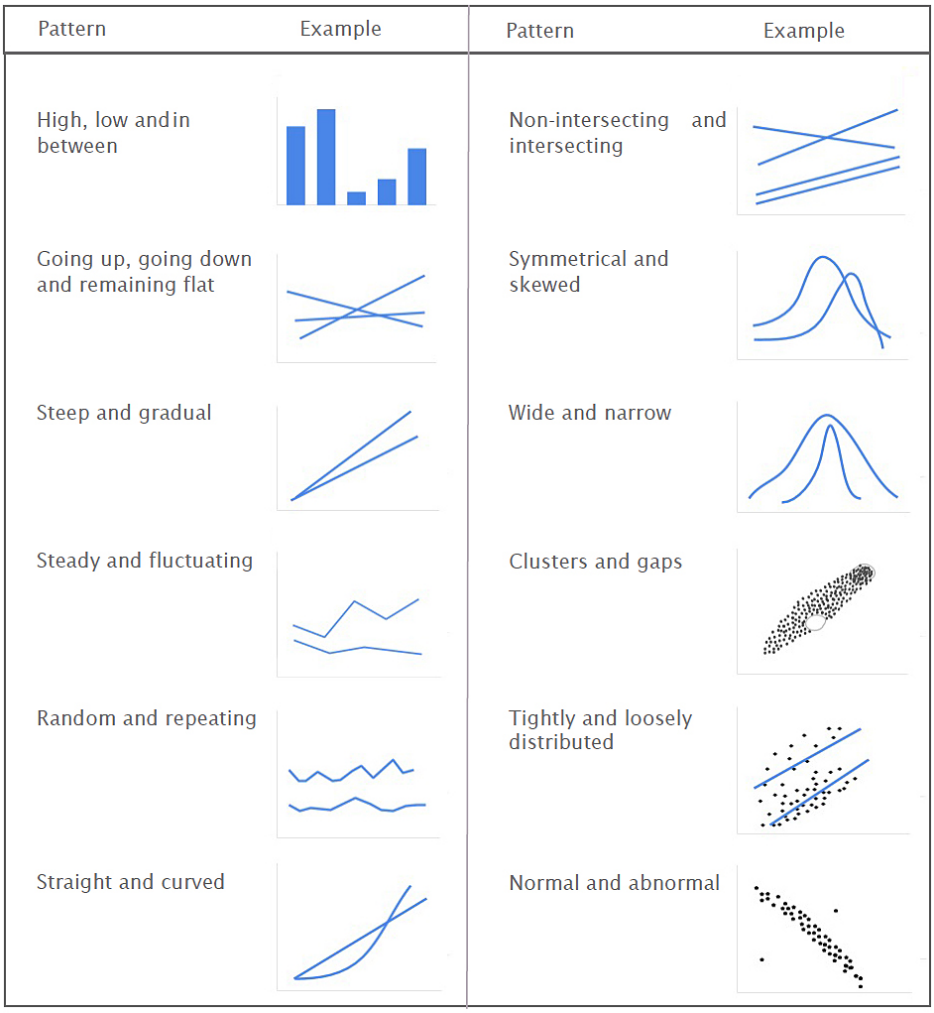
Οι τέσσερις συνεγερτικές οπτικές ιδιότητες είναι:

- Μορφή
- Χρώμα
- Θέση στο χώρο
- Κίνηση



Εικόνα 14 Τα συνεγερτικά χαρακτηριστικά (Preattentive attributes)

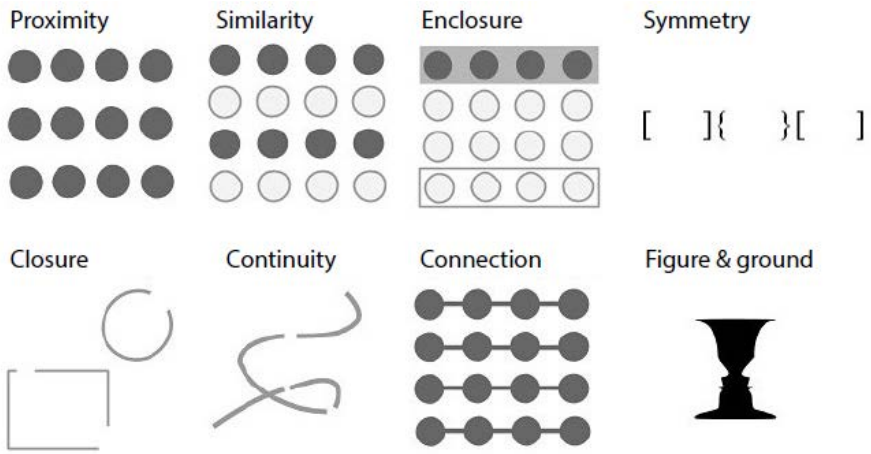
Εάν τα συνεγερτικά χαρακτηριστικά είναι το αλφάβητο της οπτικής γλώσσας τότε τα πρότυπα/μοτίβα (patterns) είναι οι λέξεις που διαμορφώνουμε με τη χρήση τους. Παρατηρώντας μια απεικόνιση, είναι άμεση η αναγνώριση των συνεγερτικών χαρακτηριστικών της. Στη συνέχεια, αυτά συνδυάζονται με σκοπό την αναζήτηση αναλυτικών μοτίβων στην απεικόνιση. Στην επόμενη εικόνα εμφανίζονται βασικά αναλυτικά μοτίβα.



Εικόνα 15 Βασικά αναλυτικά μοτίβα

Από τα παραπάνω φαίνεται πως τα συνεγερτικά χαρακτηριστικά και τα μοτίβα επιτρέπουν την επεξεργασία οπτικών πληροφοριών. Ωστόσο, όταν δημιουργούνται οπτικοποιήσεις δεδομένων, συχνά είναι αναγκαίο να επισημαίνονται ορισμένα μοτίβα σε σχέση με άλλα. Σε αυτές τις περιπτώσεις, οι αρχές Gestalt είναι χρήσιμες.

Οι αρχές **Gestalt** περιγράφουν πώς ο νους οργανώνει μεμονωμένα στοιχεία σε ομάδες. Με αυτή τη διαδικασία μπορεί να επισημάνει τα μοτίβα που είναι σημαντικά και να υποβαθμίσει άλλα μοτίβα. Η παρακάτω εικόνα απεικονίζει τις αρχές Gestalt που σχετίζονται με την απεικόνιση.



Εικόνα 16 Αρχές Gestalt που σχετίζονται με την απεικόνιση

Οι αρχές Gestalt που εμφανίζονται στην παραπάνω εικόνα είναι:

Εγγύτητα (Proximity): Οι κουκίδες δίνουν την εντύπωση ότι είναι σε τρεις σειρές αντί για τέσσερις στήλες, επειδή είναι πιο κοντά οριζόντια από όσο είναι κάθετα.

Ομοιότητα (Similarity): Στο σήμα η πρώτη εντύπωση δεν είναι οι οκτώ κουκίδες. Οι σκουρόχρωμες και οι ανοικτόχρωμες κουκίδες φαίνονται σαν μέρος της ίδιας ομάδας.

Περίβλημα (Enclosure): Οι πρώτες και οι τελευταίες τέσσερις κουκίδες ομαδοποιούνται. Έτσι ξεχωρίζουν ως δύο σειρές, και υποβιβάζεται στην εντύπωση η συνολική εικόνα με τις οκτώ κουκίδες.

Συμμετρία (Symmetry): Η αντίληψη της εικόνας από τρία ζεύγη συμμετρικών σχημάτων, υπερισχύει αυτής των έξι ατομικών παρενθέσεων.

Κλείσιμο (Closure): Η οπτική αντίληψη κλείνει αυτόματα το τετράγωνο και τον κύκλο και δεν εντοπίζει τρεις γραμμές που δεν έχουν συνδεθεί.

Συνέχεια (Continuity): Όμοια με το προηγούμενο, είναι εμφανέστερη η συνεχής πορεία αντί των τεσσάρων αυθαίρετων γραμμών.

Σύνδεση (Connection): Η οπτική αντίληψη ομαδοποιεί τις συνδεδεμένες κουκίδες ως μέλη της ίδιας ομάδας.

Σχήμα και υπόβαθρο (Figure & ground): Στην ίδια εικόνα εμφανίζονται είτε δύο πρόσωπα είτε ένα βάζο. Ανάλογα με την εστίαση του θεατή κάθε φορά, μπορεί να γίνει εναλλαγή των ρόλων του σχήματος και του υπόβαθρου.

Αυτές οι αρχές μπορούν να βοηθήσουν για να εκτελεστούν πολλές εργασίες κατά την οπτικοποίηση δεδομένων, όπως η μείωση του θορύβου από τα γραφήματα, η επιλογή του ιδανικού λόγου διαστάσεων και η πιο ξεκάθαρη εμφάνιση των σχέσεων μεταξύ των δεδομένων, κά.

Οπτικοποίηση και εξόρυξη βιολογικών δεδομένων

Η εκθετικά αυξανόμενη παραγωγή βιολογικών δεδομένων τα τελευταία χρόνια οδήγησε στην ανάπτυξη των βάσεων βιολογικών δεδομένων όπως περιγράφηκε στην προηγούμενη ενότητα. Ωστόσο, για να γίνει εφικτή η πλήρης εκμετάλλευση και η αξιοποίηση της πληροφορίας που βρίσκεται κρυμμένη σε αυτόν τον τεράστιο όγκο δεδομένων δεν αρκεί μόνο η συλλογή τους. Τα δεδομένα που συλλέγονται πρέπει να αναλυθούν με σκοπό την αποκάλυψη της πληροφορίας που ενδεχομένως κρύβουν.

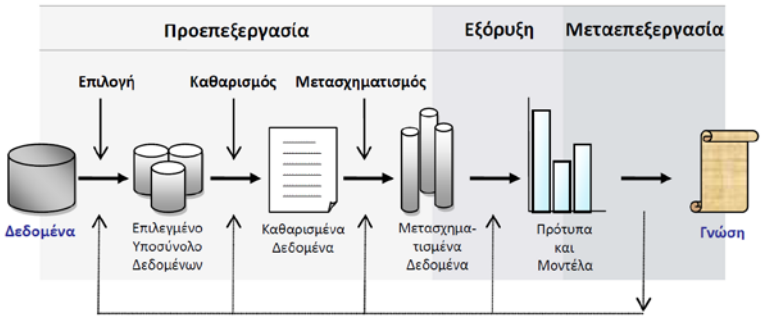
Για την επίτευξη αυτού του σκοπού, σήμερα χρησιμοποιούνται οι τεχνικές εξόρυξης και οπτικοποίησης σε συλλογές βιολογικών δεδομένων. Οι

πρώτες χρησιμοποιούνται για τον εντοπισμό των αξιόλογων συσχετίσεων μέσα σε μια μεγάλη μάζα βιολογικών δεδομένων ενώ οι δεύτερες χρησιμοποιούνται για την παρουσίαση και την οπτική αναπαράσταση της χρήσιμης γνώσης που έχει ήδη εξαχθεί σε μορφή που να είναι όσο γίνεται πιο κατανοητή από το χρήστη. Αυτό γίνεται ακόμη πιο αναγκαίο, όταν το σύστημα ανακάλυψης γνώσης είναι αλληλεπιδραστικό.

Ένας γενικός ορισμός της εξόρυξης δεδομένων (data mining) είναι ο εξής: *«Ως εξόρυξη δεδομένων εννοείται η εξαγωγή άδηλης, προηγουμένως άγνωστης και συχνά ιδιαιτέρως χρήσιμης πληροφορίας από δεδομένα»*.

Πρόκειται δηλαδή για μια διαδικασία ανακάλυψης προτύπων και σχέσεων που πιθανόν υπάρχουν μέσα σε μια συλλογή δεδομένων με απώτερο σκοπό την εξαγωγή χρήσιμη πληροφορίας. Συνήθως η συλλογή δεδομένων αναφέρεται σε κάποια μεγάλη βάση δεδομένων. Η διαδικασία αυτή μπορεί να είναι εξολοκλήρου αυτόματη ή ημι-αυτόματη όπου ο ανθρώπινος παράγοντας καλείται να αξιολογήσει και να αποτιμήσει τις σχέσεις που έχουν παραχθεί.

Στην βιοπληροφορική, η εξόρυξη γνώσης από τα βιολογικά δεδομένα παρουσιάζει ιδιαίτερο ενδιαφέρον και έχει εξαιρετικά περιθώρια εφαρμογών. Οι κυριότεροι λόγοι για αυτό είναι φυσικά ο μέχρι πρότινος ανεκμετάλλευτος τεράστιος όγκος βιολογικών δεδομένων καθώς και τα εξαιρετικά χρήσιμα και ευεργετικά συμπεράσματα που θα μπορούσαν να εξαχθούν από τέτοιου είδους δεδομένα. Τα επιμέρους στάδια της διαδικασίας ανακάλυψης γνώσης απεικονίζονται στην επόμενη εικόνα και περιγράφονται παρακάτω [4, 5].



Εικόνα 17 Τα βασικά στάδια της διαδικασίας ανακάλυψης γνώσης

Στάδιο 1: Επιλογή

Τα δεδομένα μπορούν να ληφθούν από διαφορετικές και ετερογενείς πηγές. Η επιλογή των πηγών ίσως αποτελεί και το πιο κρίσιμο στάδιο καθώς η απόφαση για την πηγή προέλευσης των δεδομένων είναι καθοριστική. Συγκεκριμένα, η ποιότητα των δεδομένων είναι αυτή που θα επηρεάσει τα αποτελέσματα. Κακής ποιότητας δεδομένα είθισται να οδηγούν σε μη αξιόπιστα συμπεράσματα. Συνήθως προέρχονται από σχεσιακές βάσεις βιολογικών δεδομένων. Οι περισσότερες βάσεις βιολογικών δεδομένων σήμερα είναι σχεσιακές ενώ η νέα τάση είναι οι αντικειμενοστραφείς βάσεις δεδομένων που αποθηκεύουν τα δεδομένα με πιο οργανωμένο τρόπο.

Στάδιο 2: Καθαρισμός

Το στάδιο αυτό, λόγω της φύσης των εργασιών που λαμβάνουν χώρα, ονομάζεται στάδιο καθαρισμού των δεδομένων (data cleaning). Επειδή η ποιότητα των δεδομένων είναι το δεύτερο καθοριστικό στοιχείο για τα αποτελέσματα της εξόρυξης, είναι επιτακτική η ανάγκη της επεξεργασίας των δεδομένων έτσι ώστε να επιτευχθεί η υψηλότερη δυνατή ποιότητα. Δεδομένα χαμηλής ποιότητας, ελλιπή ή λανθασμένα υπόκεινται σε επεξεργασία προκειμένου να απαλειφθούν οι "προβληματικές" τιμές. Η διαδικασία αυτή γίνεται συνήθως ημι-αυτόματα, χρησιμοποιώντας κάποιο πρόγραμμα λογισμικού για τον εντοπισμό των προβληματικών δεδομένων υπό την εποπτεία του ανθρώπινου παράγοντα που θα αποφασίσει για τις σωστές τιμές που θα τις αντικαταστήσουν ή ακόμη ίσως προβλέψει τις τιμές που ενδεχομένως λείπουν.

Στάδιο 3: Μετασχηματισμός

Μετά τον καθαρισμό των δεδομένων, σειρά έχει ο μετασχηματισμός τους έτσι ώστε να διευκολύνουν τη διαδικασία της εξόρυξης. Ένας συνηθισμένος λόγος που οδηγεί στον μετασχηματισμό τους, είναι η

διαφορετική πηγή προέλευσης τους. Με άλλα λόγια, η ύπαρξη δεδομένων που προέρχονται από διαφορετικές μεταξύ τους βάσεις δεδομένων. Αυτό συνεπάγεται την ανάγκη για μετατροπή αυτών των δεδομένων σε μια κοινή μορφή η οποία να επιτρέπει την επεξεργασία τους. Ένας ακόμη λόγος είναι πως ορισμένοι αλγόριθμοι προϋποθέτουν πως τα δεδομένα πάνω στα οποία θα εφαρμοστούν πρέπει να είναι οργανωμένα σε συγκεκριμένες δομές καταχώρησης δεδομένων. Επομένως και πάλι επιβάλλεται η προσαρμογή των αρχικών δεδομένων σε αυτές τις δομές.

Στάδιο 4: Εξόρυξη

Σε αυτό το στάδιο, εφαρμόζεται ο κατάλληλος αλγόριθμος πάνω στα επεξεργασμένα δεδομένα. Βασική προϋπόθεση είναι να επιλεχθεί ο κατάλληλος αλγόριθμος εξόρυξης για να εξάγει τα κατάλληλα συμπεράσματα. Η επιλογή του αλγορίθμου θα πρέπει να γίνει με προσοχή καθώς εξαρτάται από το είδος του προβλήματος που πρέπει να επιλυθεί αλλά και τους στόχους που έχουν αρχικά τεθεί.

Στάδιο 5: Μεταεπεξεργασία και Οπτικοποίηση δεδομένων

Σε αυτό το στάδιο, εφαρμόζονται διάφορες τεχνικές οπτικής αναπαράστασης (visualization). Η κατανόηση της χρησιμότητας των αποτελεσμάτων εξαρτάται σε μεγάλο βαθμό από τον τρόπο παρουσίασής τους. Για την όσο το δυνατόν πιο κατανοητή παρουσίαση των δεδομένων στο χρήστη θα πρέπει να επιλεγούν οι κατάλληλες τεχνικές. Στη συνέχεια, ο χρήστης θα μελετήσει πιθανές συσχετίσεις μεταξύ των δεδομένων και θα αναζητήσει απαντήσεις σε πιθανά ερωτήματα που μπορεί να προκύψουν από τα δεδομένα.

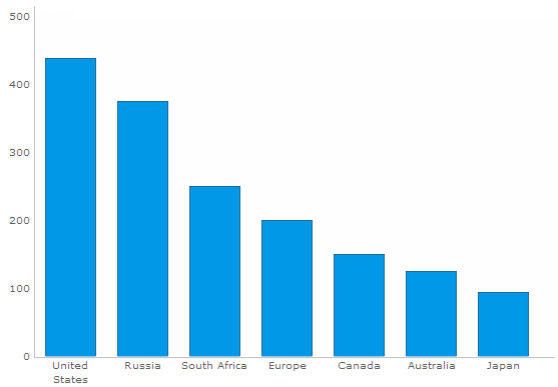
Στόχοι οπτικοποίησης δεδομένων

Όπως ήδη αναφέρθηκε προηγούμενα, οι τεχνικές οπτικοποίησης (visualization)/αναπαράστασης μπορεί να έχουν δύο ευρείς στόχους, οι οποίοι μερικές φορές αλληλεπικαλύπτονται. Θα μπορούσαν αντίστοιχα οι απεικονίσεις να διαχωριστούν σε δύο κατηγορίες, ανάλογα με το είδος των απαντήσεων που μπορεί να αναζητήσει από τα δεδομένα ο χρήστης.

- Επεξηγηματικές
- Διερευνητικές

Οι απεικονίσεις που προορίζονται να κατευθύνουν το χρήστη κατά μήκος μιας καθορισμένης διαδρομής για να αναζητήσει απαντήσεις, είναι **επεξηγηματικές**. Το μεγαλύτερο μέρος των γραφημάτων που συναντώνται καθημερινά, εμπίπτουν σε αυτήν την κατηγορία [2].

Στη συνέχεια περιγράφεται η λειτουργία μιας επεξηγηματικής απεικόνισης δεδομένων. Συγκεκριμένα, από ένα πλήθος δεδομένων σχεδιάζεται το παρακάτω ιστόγραμμα.



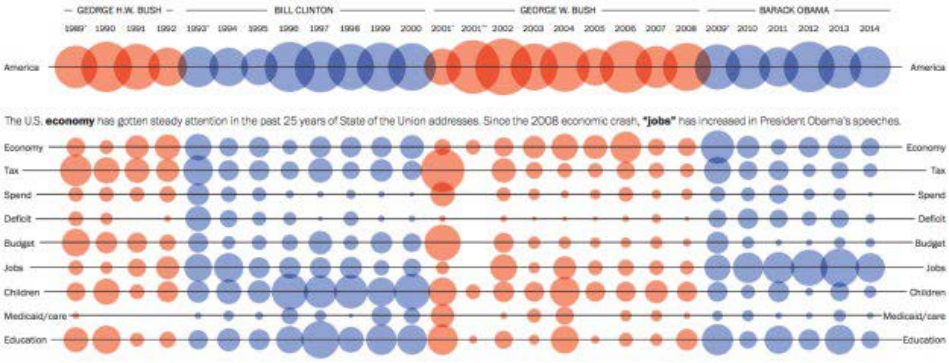
Εικόνα 18 Απλό ιστόγραμμα δεδομένων

Ο χρήστης αναζητά απάντηση στην ερώτηση «ποια χώρα έχει την υψηλότερη τιμή». Στη συνέχεια παρατηρεί την οπτικοποίηση των δεδομένων στο γράφημα και από την απεικόνιση βρίσκει άμεσα και σωστά την απάντηση στην ερώτηση.

Εικόνα 19 Λειτουργία επεξηγηματικών απεικονίσεων δεδομένων

Οι **διερευνητικές** απεικονίσεις προσφέρουν στο χρήστη περισσότερες διαστάσεις σε ένα σύνολο δεδομένων ή συγκρίνουν πολλαπλά σύνολα δεδομένων. Καλούν το χρήστη να διερευνήσει την απεικόνιση, να θέσει ερωτήσεις στην πορεία και να βρει τις απαντήσεις σε αυτές τις ερωτήσεις.

Στο παρακάτω παράδειγμα, παρουσιάζεται μια απεικόνιση από δεδομένα που προέκυψαν μελετώντας την επιλογή των λέξεων Προέδρων των ΗΠΑ στις ομιλίες τους[1].

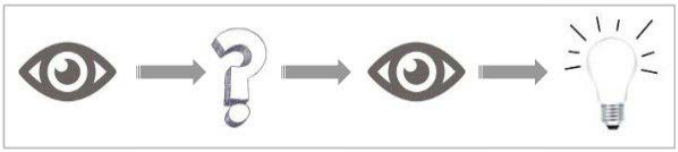


Εικόνα 20 Γράφημα μελέτης "Ιστορία μέσα από τα λόγια του Προέδρου"

Στην παραπάνω απεικόνιση ο χρήστης αρχικά προσπαθεί να εξοικειωθεί και να κατανοήσει τη λειτουργία της απεικόνισης. Στη συνέχεια προσδιορίζει την περιοχή ενδιαφέροντος, ανάλογα με το ερώτημα που διερευνά να απαντήσει. Για παράδειγμα, «ποιος πρόεδρος μίλησε περισσότερο για τις θέσεις εργασίας». Στη συνέχεια διερευνά την ενότητα "Εργασία" (Jobs) της απεικόνισης

[1] https://www.washingtonpost.com/graphics/politics/2016-sotu/language

και βρίσκει την απάντηση. Έχει επίσης τη δυνατότητα να προχωρήσει στη διερεύνηση επιπλέον ενοτήτων της απεικόνισης και να εξάγει συμπεράσματα σύγκρισης ή συσχετισμού.



Εικόνα 21 Λειτουργία διερευνητικών απεικονίσεων δεδομένων

Η παραπάνω λειτουργία μπορεί να είναι κυκλική χωρίς ένα συγκεκριμένο τελικό σημείο. Οι χρήστες μπορούν να βρουν πολλές πληροφορίες από μια ενιαία διερευνητική οπτικοποίηση και να αλληλεπιδρούν μαζί της, περισσότερο για να κατανοήσουν και να ανακαλύψουν κρυμμένες πληροφορίες, παρά να λάβουν συγκεκριμένη απάντηση.

Παρόλο που δεν είναι τόσο δημοφιλείς όσο η προηγούμενη κατηγορία, οι διερευνητικές απεικονίσεις έχουν αποκτήσει μεγάλη σημασία τα τελευταία χρόνια, ιδιαίτερα με την αύξηση του πλήθους των δεδομένων. Ο μεγάλος όγκος δεδομένων αλλά και ποικίλα σύνολα δεδομένων που μπορούν να συνδυαστούν είναι μια ιδανική επιλογή για διερευνητική ανάλυση.

Επιλογή τεχνικής για την οπτικοποίηση δεδομένων

Βασικό συστατικό στη διαδικασία επιλογής της τεχνικής οπτικοποίησης είναι να υπάρχει μια σαφής ιδέα για τα δεδομένα που θα αναπαρασταθούν γραφικά και με ποιο τρόπο. Οι απαντήσεις στις ακόλουθες ερωτήσεις μπορεί να βοηθήσουν στη διαδικασία επιλογής.

Ξεκινώντας θα πρέπει να έχει προσδιοριστεί ότι:

- Ποια δεδομένα θα χρειαστεί να οπτικοποιηθούν; Η οπτικοποίηση τελικά θα παράσχει κάτι που τα ακατέργαστα δεδομένα δεν μπορούν;

- Οι οπτικοποιήσεις θα τροφοδοτούνται από ήδη αποθηκευμένα δεδομένα για να οδηγήσουν σε στρατηγικές αποφάσεις ή θα υπάρχουν δεδομένα που παράγονται σε πραγματικό χρόνο και χρειάζεται η λήψη άμεσου αποτελέσματος;
- Ποιες λειτουργίες και ρόλους θα επιτελούν αυτές οι απεικονίσεις; Τι συμπεράσματα αναμένονται από την απεικόνιση;
- Ποιος θα ήταν ο σωστός τύπος γραφήματος για την αναπαράσταση αυτών των δεδομένων;
- Ποιο επίπεδο δεδομένων θα χρησιμοποιηθεί; Συγκεκριμένα, οι αναφορές θα αναπαριστούν μόνο συγκεντρωτικά δεδομένα ή θα επιτρέπεται στους χρήστες να τα αναλύουν λεπτομερώς;
- Υπάρχουν οντότητες δεδομένων που σχετίζονται μεταξύ τους και μπορούν να συγκεντρωθούν για να δώσουν πιο ουσιαστικές πληροφορίες;
- Είναι απαραίτητο να προσφέρεται στους χρήστες η ευελιξία επιλογής προβολής διαφορετικών προοπτικών των δεδομένων;

Μετά τις απαντήσεις στις παραπάνω ερωτήσεις, το επόμενο βήμα είναι να αναζητηθεί η τεχνική οπτικοποίησης που θα τις ικανοποιεί. Το στάδιο επιλογής της τεχνικής συνιστάται να γίνεται στην αρχική φάση της σχεδίασης της εφαρμογής, έτσι ώστε να μπορεί να προγραμματιστεί τόσο η διεπαφή χρήστη όσο και η γενικότερη υλοποίηση της εφαρμογής [6].

Διαγράμματα για οπτικοποίηση δεδομένων

Η λήψη ορισμένων αποφάσεων είναι απαραίτητη για να γίνει μία σωστή οπτικοποίηση. Αρχικά πρέπει να καθοριστούν τα ερωτήματα που πρέπει να απαντηθούν εύκολα από την επικείμενη οπτικοποίηση, έπειτα να αναγνωριστούν τα κατάλληλα δεδομένα και τέλος να επιλεγούν αποδοτικές κωδικοποιήσεις για να αντιστοιχιστούν με εύχρηστο τρόπο οι τιμές των δεδομένων σε γραφικές λειτουργίες (θέση, μέγεθος, σχήμα, χρώμα). Στην πορεία της διαδικασίας υπάρχουν πολλές προκλήσεις, όπως το πλήθος των

οπτικών κωδικοποιήσεων, ο αποθηκευτικός χώρος απαιτείται ή η επεξεργαστική ισχύς που είναι απαραίτητη.

Υπάρχουν τέσσερις βασικοί τύποι παρουσίασης δεδομένων με τους 2 πρώτους να είναι οι πιο δημοφιλείς:

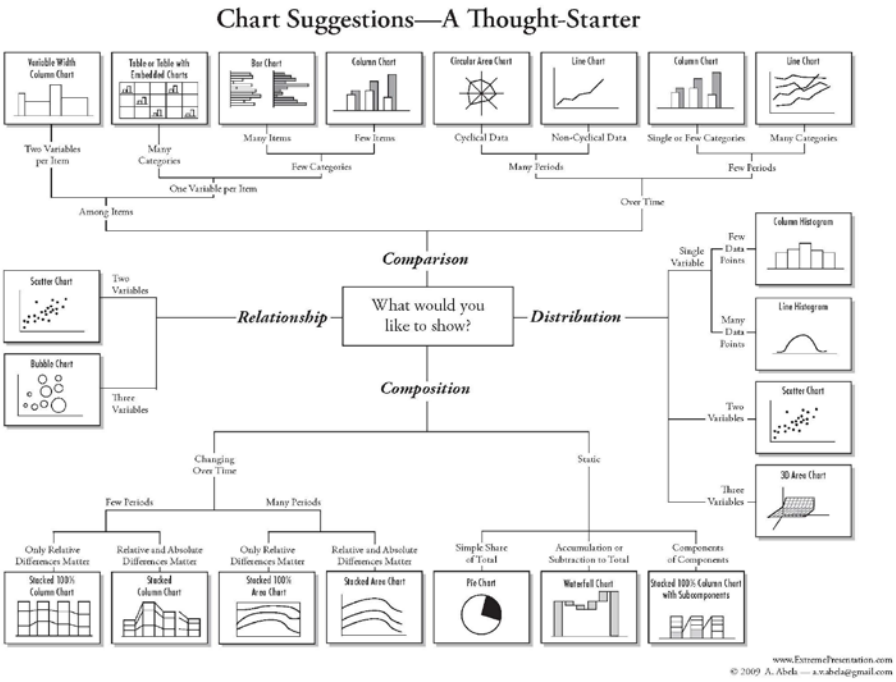
- Σύγκριση (Comparison)
- Σύνθεση (Composition)
- Κατανομή (Distribution)
- Συσχέτιση (Relationship)

Για να καθοριστεί ποιο διάγραμμα ταιριάζει καλύτερα σε κάθε έναν από αυτούς τους τύπους παρουσίασης, πρώτα πρέπει να απαντηθούν μερικές σημαντικές ερωτήσεις:

- Πόσες μεταβλητές να εμφανίζονται σε ένα γράφημα;
- Πόσα σημεία δεδομένων να εμφανίζονται για κάθε μεταβλητή;
- Θα προβληθούν τιμές για μια χρονική περίοδο ή μεταξύ σημείων δεδομένων ή ομάδων;

Τα ραβδογράμματα (bar charts) είναι καλά για Σύγκριση (Comparison) ενώ τα διαγράμματα (line charts) είναι καλύτερα για να εμφανίζεται η τάση (trend). Τα διαγράμματα διασποράς (scatter plot charts) είναι καλύτερα για Κατανομή (Distribution) και Συσχέτιση (Relationship) και τα διαγράμματα πίτας (pie charts) είναι καλύτερα για Σύνθεση (Composition).

Στη συνέχεια παρουσιάζεται ένα διάγραμμα καθοδήγησης για την επιλογή γραφήματος που δημιουργήθηκε από τον Δρ. Andrew Abela [7].



Εικόνα 22 Διάγραμμα καθοδήγησης για την επιλογή γραφήματος που δημιουργήθηκε από τον Δρ. Andrew Abela

Επιλογή εργαλείων οπτικοποίησης δεδομένων

Για την οπτικοποίηση δεδομένων υπάρχει πληθώρα προσφερόμενων εργαλείων. Συγκεκριμένα, η δημιουργία ενός συγκεκριμένου γραφήματος (πχ γράφημα πίτας), είναι δυνατό να προσφέρεται από διαφορετικά εργαλεία οπτικοποίησης. Συνεπώς, είναι απαραίτητο να υπάρξει μια διαδικασία επιλογής του εργαλείου ανάμεσα από αυτά που είναι διαθέσιμα στην αγορά [6].

Στη συνέχεια αναφέρονται παράγοντες που προτείνεται να εξεταστούν για την τελική επιλογή του εργαλείου οπτικοποίησης δεδομένων.

- Τι είδους συσκευές, προγράμματα περιήγησης και πλατφόρμες θα πρέπει να υποστηρίζονται.
- Πόση εύκολη είναι η υλοποίηση και η ενσωμάτωση στην εφαρμογή.

- Σε τι βαθμό παρέχονται οι τύποι γραφημάτων που χρειάζονται για την ήδη σχεδιασμένη αναπαράσταση των δεδομένων.
- Τι δυνατότητες τροποποιήσεων των γραφημάτων υπάρχουν, ώστε να δίνεται η δυνατότητα εμπλουτισμού του τελικού αποτελέσματος.
- Αν υπάρχει και σε ποιο βαθμό η απαραίτητη τεχνική υποστήριξη.

Εργαλεία οπτικοποίησης δεδομένων

Υπάρχουν πάρα πολλά εργαλεία οπτικοποίησης δεδομένων και στις μέρες μας αυτά γίνονται όλο και πιο αξιόπιστα και εύχρηστα. Στη συνέχεια αυτής της ενότητας παρουσιάζονται τα πιο σημαντικά από αυτά καθώς και αυτά που επελέγησαν κατά τη διάρκεια της ερευνητικής αυτής εργασίας.

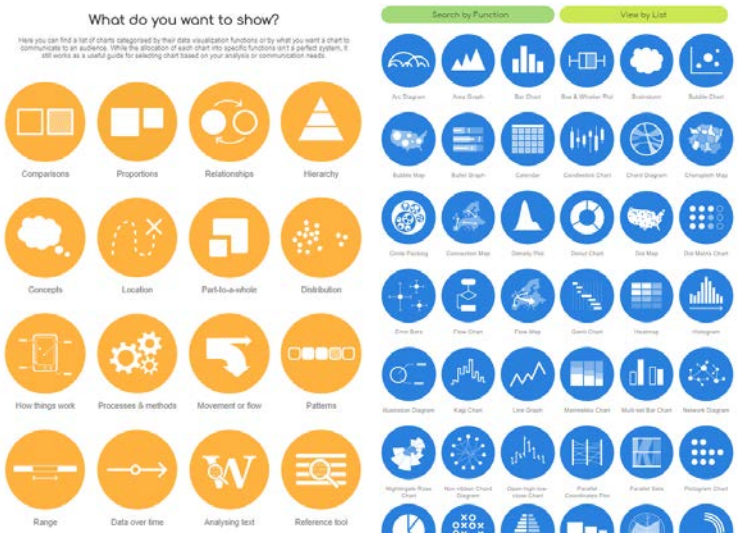
Αν και έχουν υπάρξει προσπάθειες στο παρελθόν να δημιουργηθούν κατάλογοι μερικών από τις καθιερωμένες μεθόδους οπτικοποίησης δεδομένων, δεν υπήρχε κανένας ιστοχώρος που να είναι πραγματικά περιεκτικός, λεπτομερής ή να βοηθά το χρήστη να αποφασίσει την καταλληλότερη μέθοδο για τις εκάστοτε ανάγκες.

Ένας ιδιαίτερα ενδιαφέρων κατάλογος οπτικοποίησης δεδομένων αναπτύχθηκε από τον Severino Ribecca, ο οποίος συνέλεξε σε έναν ιστοχώρο (datavizcatalogue.com) και δημιούργησε μια βιβλιοθήκη διαφορετικών τύπων οπτικοποίησης δεδομένων [8].

Αρχικά, το έργο αυτό ήταν ένας τρόπος ώστε ο δημιουργός του να αναπτύξει τις δικές του γνώσεις για την οπτικοποίηση των δεδομένων και να δημιουργήσει ένα εργαλείο αναφοράς. Είναι παράλληλα επωφελής τόσο για τους σχεδιαστές όσο και για όλους όσους εργάζονται σε ένα πεδίο που απαιτεί τη χρήση οπτικοποίησης δεδομένων.

Παρόλο που υπήρξαν κάποιες απόπειρες κατά το παρελθόν για την καταγραφή μερικών από τις καθιερωμένες μεθόδους απεικόνισης δεδομένων, ο συγκεκριμένος ιστότοπος είναι πραγματικά πλήρης, λεπτομερής και βοηθά να εντοπίσει κανείς τη σωστή μέθοδο για τις ανάγκες του.

Τα περισσότερα από τα δεδομένα που απεικονίζονται στα παραδείγματα εικόνων του ιστότοπου είναι εικονικά.



Εικόνα 23 Βιβλιοθήκη γραφημάτων datavizcatalogue.com

Τεχνολογίες εργαλείων οπτικοποίησης δεδομένων

Στις αρχές της δεκαετίας του 1990, όταν ο Ιστός ήταν ακόμα σε εκκολαπτόμενο στάδιο, ήταν πολύ κοινό στις ιστοσελίδες να διευκρινίζεται το σύνολο προγραμμάτων περιήγησης που θα ήταν καλύτερα να χρησιμοποιήσει ο χρήστης για τη σωστή εμφάνισή του.

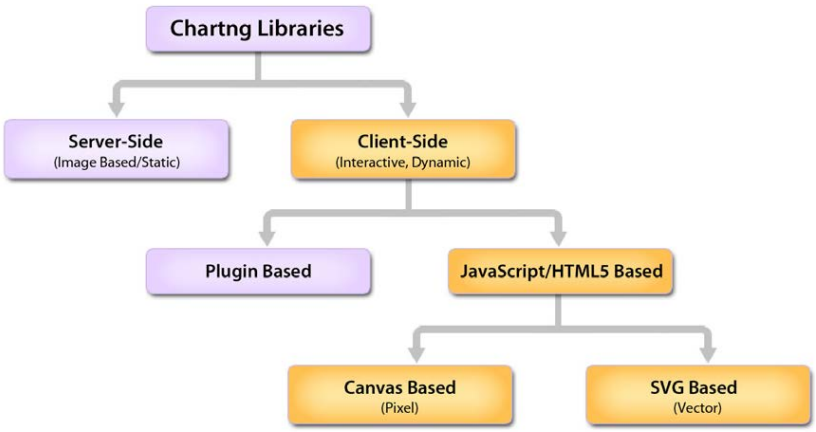
Σήμερα, η ραγδαία τεχνολογική εξέλιξη στον τομέα της πληροφορικής, οδήγησε τις δικτυακές εφαρμογές να αντικαθιστούν τις εφαρμογές υπολογιστών γραφείου και μάλιστα χωρίς τους παραπάνω περιορισμούς που υπήρχαν παλαιότερα. Επιπλέον, οι χρήστες θέλουν πρόσβαση στις εφαρμογές τους από τους υπολογιστές τους στην εργασία, στα tablet και στα smartphone τους εν κινήσει. Ως εκ τούτου το λογισμικό και όλα τα χαρακτηριστικά του πρέπει να υποστηρίζονται, αν όχι όλα, αλλά μια πλειοψηφία, σε διάφορες συσκευές και προγράμματα περιήγησης.

Οι διαθέσιμες τεχνολογίες εργαλείων οπτικοποίησης δεδομένων μπορεί να διαχωριστούν σε δύο βασικές γενικές κατηγορίες.

1. Η πρώτη κατηγορία, στηρίζεται στη δημιουργία γραφημάτων στην πλευρά του διακομιστή (server side). Οι βιβλιοθήκες που δημιουργούν τα γραφήματα είναι βασισμένα σε εικόνες.

2. Στη δεύτερη κατηγορία, ανήκουν οι βιβλιοθήκες γραφημάτων οι οποίες λειτουργούν στην πλευρά του χρήστη (client side) που (συνήθως) δημιουργούν αλληλεπιδραστικά γραφήματα σε πραγματικό χρόνο.

Στο παρακάτω γράφημα παρουσιάζονται οι διαθέσιμες τεχνολογίες εργαλείων οπτικοποίησης δεδομένων [6].



Εικόνα 24 Τεχνολογίες οπτικοποίησης

Βιβλιοθήκες γραφημάτων στην πλευρά του διακομιστή

Αυτές οι βιβλιοθήκες της συγκεκριμένης πλατφόρμας δέχονται δεδομένα στο διακομιστή μέσω των API τους, δημιουργούν εικόνες για να σχεδιάσουν το γράφημα και στη συνέχεια αποστέλλουν αυτές τις εικόνες ως αποτέλεσμα στο χρήστη. Το πλεονέκτημα της χρήσης αυτών των βιβλιοθηκών είναι ότι παρέχουν τα ίδια αποτελέσματα σε όλες τις συσκευές, λόγω της ύπαρξης

εικόνων. Επίσης για μικρά και ένα ή δύο γραφήματα δεν απαιτείται μεγάλο εύρος ζώνης σύνδεσης για τη μεταφορά τους στο χρήστη.

Παραδείγματα τέτοιων στοιχείων είναι Telerik (.NET), Infragistics (.NET), ComponentArt (.NET), ChartFX (Java και .NET), Steema (.NET), pChart (PHP) και jPGraph (PHP). Τα περισσότερα από αυτά είναι βιβλιοθήκες που έχουν ωριμάσει αρκετά τεχνολογικά, ώστε να προσφέρουν ένα ευρύ φάσμα τύπων γραφημάτων.

Ωστόσο, ένα μειονέκτημα της χρήσης αυτών των βιβλιοθηκών είναι ότι όταν υπάρχουν πολλά διαγράμματα για παρουσίαση, καταναλώνονται σημαντικοί πόροι του διακομιστή κατά τη δημιουργία των γραφημάτων ως εικόνες, πολύ περισσότερο δε όταν υπάρχουν αρκετοί ταυτόχρονοι χρήστες. Δεύτερο μειονέκτημα που παρουσιάζουν είναι ότι επειδή είναι βασισμένες σε εικόνες, έχουν μικρή ή καμία διαδραστικότητα για το χρήστη. Τέλος, επειδή αυτές οι βιβλιοθήκες εξαρτώνται από την τεχνολογία του διακομιστή, σε περίπτωση αλλαγής στην τεχνολογία του διακομιστή, θα πρέπει να γίνει αλλαγή και στη βιβλιοθήκη γραφημάτων. Μοιραία αυτό θα οδηγήσει σε μια διαφορετική εμφάνιση και αίσθηση του χρήστη για τα διαγράμματα.

Βιβλιοθήκες γραφημάτων στην πλευρά του χρήστη

Η δεύτερη επιλογή είναι να χρησιμοποιηθούν βιβλιοθήκες γραφημάτων που λειτουργούν στη συσκευή του χρήστη χρησιμοποιώντας τα πρόσθετα JavaScript, CSS, flash, Silverlight ή Java applets. Λαμβάνοντας υπόψη ότι οι συσκευές iOS δεν υποστηρίζουν κάποια από τα παραπάνω πρόσθετα (flash, Java και Silverlight) είναι ασφαλέστερο, αν είναι δυνατό, να αποφεύγονται μιας και ένας τεράστιος αριθμός χρηστών χρησιμοποιεί iOS.

Αντίθετα η χρήση των CSS είναι μια ασφαλής λύση και μπορεί να χρησιμοποιηθεί για πολύ βασικές απεικονίσεις. Αυτό μας φέρνει αναγκαία στην τεχνολογική λύση των JavaScript/HTML5. Είναι τεχνολογίες που έχουν υιοθετηθεί από την πλειοψηφία των συσκευών και browsers και τις έχουν

αγκαλιάσει οι μεγαλύτεροι ηγέτες της βιομηχανίας του διαδικτύου, συμπεριλαμβανομένων των Google, Apple και Microsoft.

Τα εργαλεία οπτικοποίησης JavaScript/HTML5 μπορούν να χωριστούν και πάλι στις δύο παρακάτω κατηγορίες:

- Στοιχεία που βασίζονται στο Canvas API
- Στοιχεία βασισμένα σε Scalable Vector Graphics (SVG)

Εργαλεία οπτικοποίησης JavaScript/HTML5

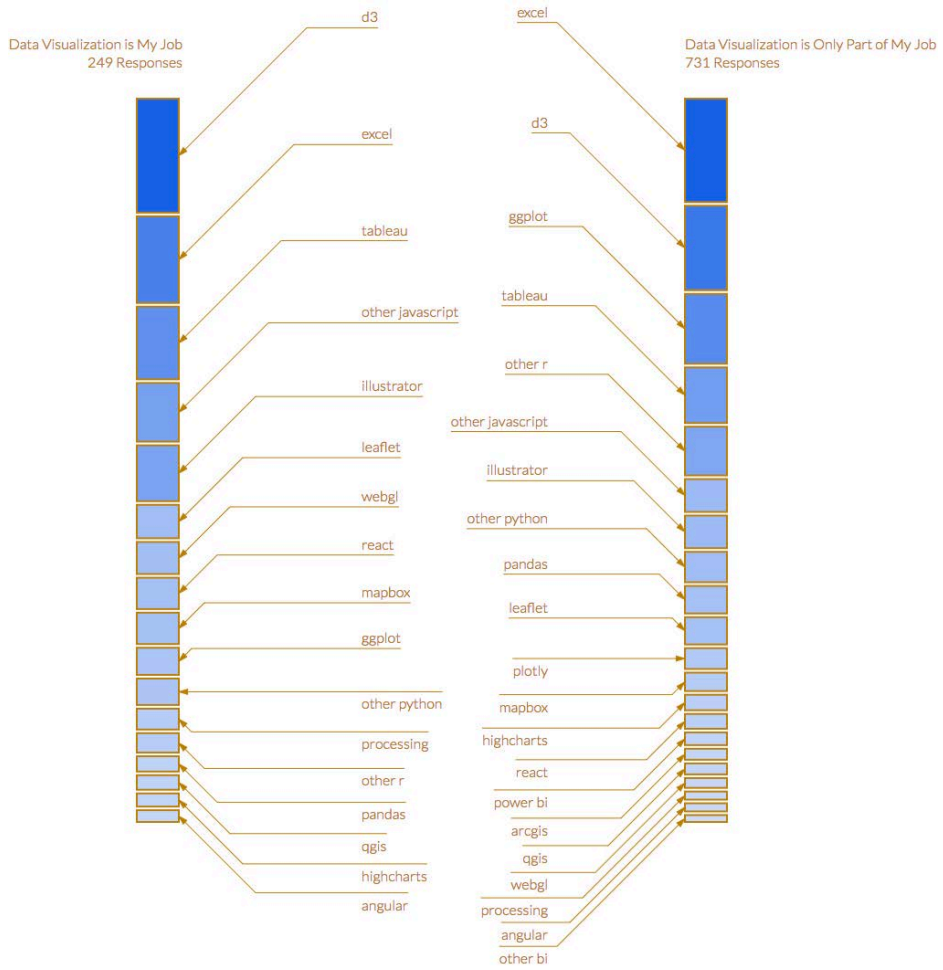
Γενικά, το Canvas API και η SVG είναι και οι δύο τεχνολογίες web που έχουν σχεδιαστεί ώστε να αποδίδουν πλούσια γραφικά μέσα στο πρόγραμμα περιήγησης.

Η τεχνολογία SVG εμφανίστηκε περίπου το 1999 ενώ το Canvas API υπάρχει από το 2005 περίπου. Ενώ φαίνεται να αρκετά παρόμοια, υπάρχει μια πολύ σημαντική βασική διαφορά μεταξύ τους. Η τεχνολογία SVG δημιουργεί διανυσματικά γραφικά, ενώ αντίθετα τα γραφικά που δημιουργεί το Canvas API βασίζονται σε εικονοστοιχεία (pixel). Έτσι, τα SVG γραφικά είναι καταλληλότερα για εφαρμογές Web, δεδομένου ότι μπορεί να είναι διαδραστικά και δυναμικά. Σε αντίθεση τα γραφικά με το Canvas API, για να υπάρξει κάθε είδους διαδραστικότητα, θα πρέπει να επανασχεδιαστεί το πλήρες γραφικό. Αυτό κάνει τα SVG γραφικά, μια σωστή επιλογή για τη δημιουργία αλληλεπιδραστικών γραφημάτων.

Δημοτικότητα εργαλείων οπτικοποίησης δεδομένων

Σχετική έρευνα σχετικά με τα εργαλεία οπτικοποίησης δεδομένων [9], που πραγματοποιήθηκε από 27 Φεβρουαρίου έως τις 8 Μαρτίου 2017, είχε ως στόχο να ερευνήσει την κατάσταση που επικρατεί στην επαγγελματική απεικόνιση δεδομένων.

Μέσα από μια σειρά 45 ερωτήσεων, οι 981 ερωτηθέντες απάντησαν, μεταξύ άλλων, ποιοι ήταν οι τίτλοι εργασίας που σχετίζονται με την οπτικοποίηση των δεδομένων τους, τα εργαλεία που χρησιμοποιήθηκαν, τους ηγέτες της σκέψης, τα προβλήματα που αντιμετώπισαν αλλά και δημογραφικά στοιχεία. Η έρευνα εστιάζει στην επαγγελματική οπτικοποίηση δεδομένων και την επιλογή των εργαλείων και πώς αυτό ποικίλλει αν κάποιος επικεντρώνεται κυρίως στην οπτικοποίηση δεδομένων ή όχι.



Εικόνα 25 Η διάδοση των εργαλείων οπτικοποίηση ανάλογα με το ρόλο τους στις εργασίες οπτικοποίησης δεδομένων

Επίσης, μία από τις πιο ενδιαφέρουσες απαντήσεις ήταν αν οι ερωτηθέντες είναι ειδικοί στην απεικόνιση δεδομένων. Το αποτέλεσμα σαφώς έδειξε ότι αυτό δεν συμβαίνει (το 75% απάντησε "Όχι"). Η απάντηση αυτή, σε συσχέτιση με την απάντηση στο ερώτημα «Πόσα χρόνια ασχολείστε με την οπτικοποίηση δεδομένων;», όπου περίπου το 20% των ερωτηθέντων ασχολείται γύρω στα 4 χρόνια, αντικατοπτρίζει ξεκάθαρα την αυξανόμενη δημοτικότητα του τομέα.

Εργαλεία οπτικοποίησης δεδομένων που μελετήθηκαν

Όπως προκύπτει και από την προηγούμενη έρευνα, στις μέρες μας υπάρχουν πάρα πολλά εργαλεία οπτικοποίησης δεδομένων, και μάλιστα αρκετά αξιόπιστα και εύχρηστα. Στη συνέχεια αυτής της ενότητας θα παρουσιαστούν αυτά τα οποία μελετήθηκαν στην παρούσα ερευνητική εργασία και αυτά που τελικά επιλέχθηκαν για υλοποίηση.

Microsoft Pivot Viewer

Το Microsoft Pivot Viewer [10] είναι ένα εργαλείο οπτικοποιήσεων βασισμένο στο Silverlight [11] το οποίο παράγει ένα διαδραστικό περιβάλλον φιλικό προς το χρήστη και στοχεύει στη μαζική αναπαράσταση δεδομένων μεγάλου όγκου (big data), στο φιλτράρισμα και στην ταξινόμησή τους.

Εικόνα 26 Παράδειγμα οπτικοποίησης με το Miscrosoft Pivot Viewer

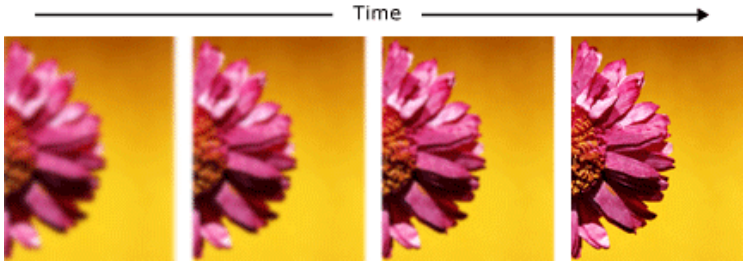
Στις μέρες μας, με την έλευση των τεχνολογιών της HTML5 οι περιηγητές (browsers) αποτρέπουν τη χρήση plugins όπως το Silverlight και το Flash, και έγινε μία προσπάθεια το Microsoft Pivot Viewer να μετατραπεί να λειτουργεί χωρίς Silverlight αλλά μόνο με jQuery [12] (μία JavaScript βιβλιοθήκη) και το αποτέλεσμα εμφανίζεται με το όνομα html5pivotviewer [13].

Το Microsoft Pivot Viewer βασίστηκε σε μία τεχνολογία με όνομα Deep Zoom [14], η οποία είναι μια τεχνολογία που αναπτύχθηκε από τη Microsoft για την αποτελεσματική μετάδοση και προβολή εικόνων. Επιτρέπει στους χρήστες να περιηγηθούν και να μεγεθύνουν μια μεγάλη εικόνα υψηλής ανάλυσης ή μια μεγάλη συλλογή εικόνων. Μειώνει τον απαιτούμενο χρόνο στην έναρξη της διαδικασίας, μεταφορτώντας μόνο την περιοχή που είναι κάθε φορά ορατή και την απαραίτητη ανάλυση στην οποία εμφανίζεται. Οι μεταγενέστερες περιοχές κατεβαίνουν στον υπολογιστή καθώς ο χρήστης περιηγείται (ή μεγεθύνει).

Η μορφή αρχείου Deep Zoom είναι παρόμοια με τη μορφή της εικόνας των Χαρτών Google, όπου οι εικόνες σπάζουν σε πλακίδια (tiles) και στη συνέχεια εμφανίζονται όταν απαιτείται. Η κύρια διαφορά είναι ότι με τους Χάρτες Google οι πραγματικές λεπτομέρειες της εικόνας αλλάζουν από το ένα επίπεδο ζουμ σε ένα άλλο, ενώ με το Deep Zoom εμφανίζεται η ίδια εικόνα σε κάθε επίπεδο ζουμ.

Το Deep Zoom παρέχει επίσης, τη δυνατότητα διαδραστικής προβολής εικόνων υψηλής ανάλυσης. Μπορεί να γίνει μεγέθυνση και σμίκρυνση των εικόνων γρήγορα χωρίς να επηρεαστεί η απόδοση της εφαρμογής. Για να τα καταφέρει το Deep Zoom χρησιμοποιεί εικόνες πολλαπλών αναλύσεων για να επιτύχει υψηλή αναλογία καρέ και μικρό χρόνο φόρτωσης στην έναρξη ακόμα και για πολύ μεγάλες εικόνες.

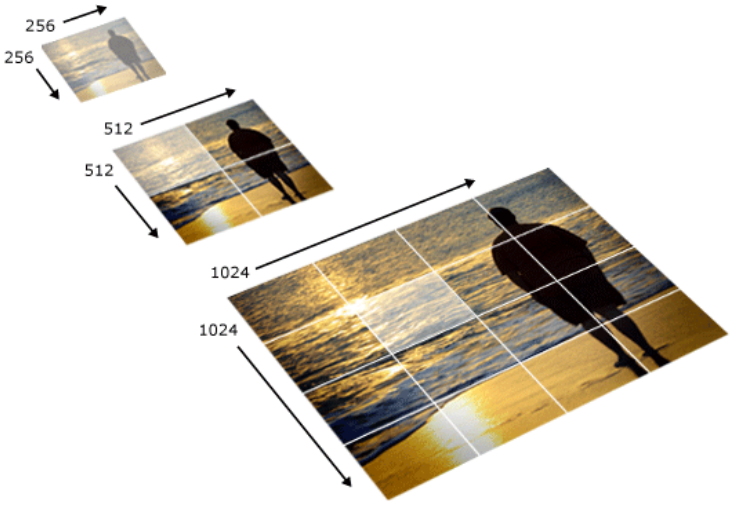
Με την έναρξη χρειάζεται μόνο μια μικρή ποσότητα δεδομένων για να εμφανιστεί γρήγορα κάτι στην οθόνη. Το πρώτο στάδιο κατά τη φόρτωση της εικόνας είναι να εμφανιστεί μια έκδοση χαμηλής ανάλυσης της εικόνας και σιγά σιγά να εμφανίζεται η εικόνα σε υψηλότερη ανάλυση καθώς οι πληροφορίες κατεβαίνουν στον υπολογιστή του χρήστη.



Εικόνα 27 Με την πάροδο του χρόνου η εικόνα μετατρέπεται σταδιακά από θολή σε καθαρή

Όπως φαίνεται στην παραπάνω εικόνα, η εικόνα εμφανίζεται αρχικά θολή και με την πάροδο του χρόνο (καθώς πληροφορίες κατεβαίνουν στον υπολογιστή) μετατρέπεται σε καθαρή. Εκτός από την αρχική φόρτωση, αυτή η ίδια συμπεριφορά επαναλαμβάνεται καθώς ο χρήστης αλληλοεπιδρά με την εφαρμογή.

Η παρακάτω εικόνα δείχνει πώς λειτουργεί η τεχνολογία Deep zoom. Η ίδια η εικόνα είναι διαθέσιμη σε πλήρη ανάλυση στο κάτω μέρος της πυραμίδας αλλά και εκδόσεις χαμηλότερης ανάλυσης σε pixel 4x4 αποθηκεύονται παράλληλα με την εικόνα πλήρους ανάλυσης. Οι εικόνες σε κάθε επίπεδο της πυραμίδας αποθηκεύονται σε 256x256 pixel πλακίδια (υποδεικνύεται από τις λευκές γραμμές στις εικόνες).



Εικόνα 28 Εφαρμογή της τεχνολογίας Deep Zoom στις εικόνες

Το εργαλείο αυτό υλοποιήθηκε στην ανάπτυξη της **RGDtrip** η οποία περιγράφεται στο επόμενο κεφάλαιο.

D3.js

Η βιβλιοθήκη D3.js [15] είναι μια μία από τις πιο δημοφιλείς ανοιχτού κώδικα βιβλιοθήκες JavaScript για το χειρισμό εγγράφων με βάση τα δεδομένα. Η βιβλιοθήκη D3.js βοηθά τους προγραμματιστές να «ζωντανέψουν» τα δεδομένα χρησιμοποιώντας τεχνολογίες όπως HTML, SVG και CSS. Η βιβλιοθήκη D3.js δίνει έμφαση στα πρότυπα του ιστού (web standards) και είναι συμβατή με όλους τους περιηγητές.

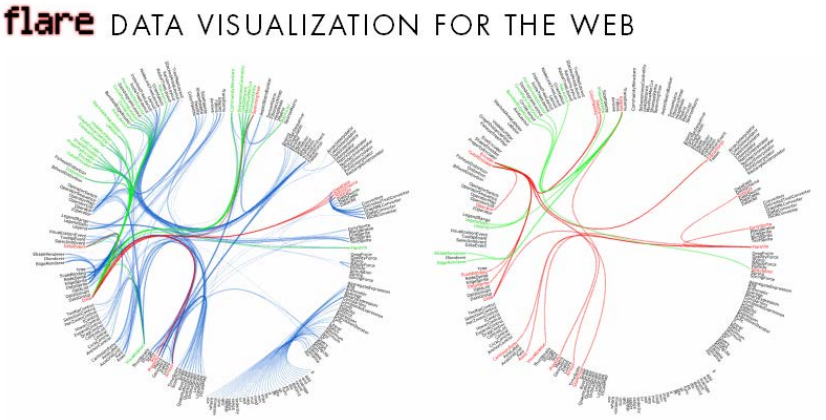
Η D3 είναι ένα εξαιρετικά γρήγορο εργαλείο οπτικοποίησης δεδομένων, υποστηρίζοντας μεγάλα σύνολα δεδομένων και δυναμικές συμπεριφορές για

αλληλεπίδραση και κινούμενα σχέδια. Το λειτουργικό στυλ της D3 επιτρέπει την επαναχρησιμοποίηση του κώδικα μέσω μιας ποικίλης συλλογής επίσημων αλλά και ελεύθερα διακινούμενων τμημάτων κώδικα.

Το εργαλείο αυτό χρησιμοποιήθηκε για την οπτικοποίηση δεδομένων της βάσης δεδομένων **fungibase** η οποία περιγράφεται σε επόμενο κεφάλαιο.

Prefuse Flare

Το Flare [16] είναι μία βιβλιοθήκη ανοιχτού κώδικα γραμμένη σε ActionScript για τη δημιουργία οπτικοποιήσεων που εκτελούνται με τον Adobe Flash Player. Ήταν μία από τις πρώτες και πιο δημοφιλείς βιβλιοθήκες για δημιουργία οπτικοποιήσεων όσο ήταν δημοφιλές το Adobe Flash. Αυτό άλλαξε μετά την έλευση των τεχνολογιών που στηρίζονται σε html5.



Ο σχεδιασμός του Flare προέκυψε από το Prefuse, ένα εργαλείο οπτικοποίησης για Java.

ProcessingJS

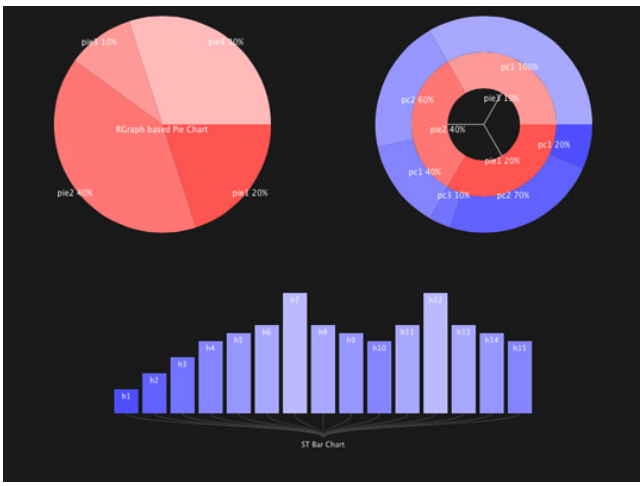
Το Processing.js είναι το αδελφικό πρόγραμμα της δημοφιλούς γλώσσας οπτικού προγραμματισμού Processing, η οποία έχει σχεδιαστεί ειδικά για το διαδίκτυο. Το Processing.js δημιουργεί απεικονίσεις δεδομένων, ψηφιακή τέχνη,

διαδραστικά κινούμενα σχέδια, εκπαιδευτικά γραφήματα, βιντεοπαιχνίδια κ.λπ. χρησιμοποιώντας πρότυπα ιστού και χωρίς plug-ins.

Αρχικά η Processing αναπτύχθηκε από τους Ben Fry και Casey Reas. Ξεκίνησε ως γλώσσα προγραμματισμού ανοιχτού κώδικα βασισμένη στη Java για να βοηθήσει τις κοινότητες ηλεκτρονικών τεχνών και οπτικής σχεδίασης να χειριστούν τα βασικά του προγραμματισμού των υπολογιστών σε ένα οπτικό πλαίσιο. Το Processing.js μεταφέρει αυτό στο επόμενο επίπεδο, επιτρέποντας την εκτέλεση του κώδικα επεξεργασίας από οποιοδήποτε συμβατό πρόγραμμα περιήγησης HTML5, συμπεριλαμβανομένων των πρόσφατων εκδόσεων του Firefox, του Safari, του Chrome, του Opera και του Internet Explorer. Το Processing.js φέρνει το καλύτερο οπτικό πρόγραμμα στον ιστό, τόσο για επεξεργαστές όσο και για web developers.

Javascript Infovis Toolkit

Το εργαλείο JavaScript InfoVis Toolkit είναι μια βιβλιοθήκη που έχει γραφτεί από τον Nicolas Belmonte. Περιλαμβάνει μια σπονδυλωτή (modular) δομή που επιτρέπει τους επισκέπτες να μεταφορτώσουν ό,τι είναι απολύτως απαραίτητο για την εμφάνιση των απεικονίσεων των δεδομένων. Αυτή η βιβλιοθήκη έχει μια σειρά από μοναδικά στυλ και εφέ κινουμένων σχεδίων και είναι ελεύθερης χρήσης.



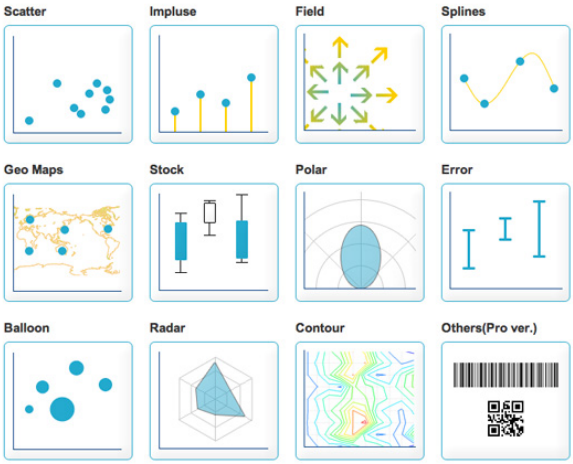
Το Javascript Infovis Toolkit είναι μία βιβλιοθήκη οπτικοποιήσεων που δίνει τη δυνατότητα κατασκευής πολύπλοκων δομών οπτικοποίησης όπως TreeMaps, Rgraph, Hypetree, TreeMap, SpaceTree, Icicle, Sunburst, ForceDirected.

PowerBi

Το PowerBI [19] είναι μια υπηρεσία επιχειρηματικών αναλύσεων (business analytics service) που παρέχεται από τη Microsoft. Παρέχει διαδραστικές οπτικοποιήσεις με δυνατότητες επιχειρηματικής ευφυΐας (business intelligence), όπου οι χρήστες μπορούν να δημιουργούν reports και dashboards από μόνοι τους (serf service BI) χωρίς να εξαρτώνται από κάποιο ειδικευμένο προσωπικό πληροφορικής ή κάποιον διαχειριστή βάσης δεδομένων. Επιπλέον δίνει την δυνατότητα δημιουργίας νέων οπτικοποιήσεων εκτός από αυτές που υποστηρίζει εγγενώς μέσω του custom visuals οι οποίες βασίζονται στη βιβλιοθήκη d3.js. Το power BI είναι μία σχετικά νέα υπηρεσία και προσφέρεται σαν SaaS (Software as a Service).

jpGraph

Η βιβλιοθήκη γραφικών jpGraph [20] προσφέρει μια λύση βασισμένη σε PHP με ένα ευρύ φάσμα τύπων γραφημάτων. Λειτουργεί από την πλευρά του διακομιστή, ενώ είναι δωρεάν για μη εμπορική χρήση διαθέτοντας εκτενή τεκμηρίωση.



Ανάλογα με την απόδοση του διακομιστή, το jpGraph παρέχει εγγυημένα ένα αισθητικά όμορφο οπτικό αποτέλεσμα σε βάρος όμως της διαδραστικότητας και της προσβασιμότητας.

Chart.js

Αν και υποστηρίζει μόνο έξι τύπους γραφημάτων, η βιβλιοθήκη ανοιχτού κώδικα Chart.js [21] είναι το τέλειο εργαλείο απεικόνισης δεδομένων για χόμπι και μικρά έργα. Χρησιμοποιώντας HTML5 για την απόδοση γραφημάτων το Chart.js δημιουργεί ανταποκρίσιμα (responsive) επίπεδα σχέδια, ενώ εξελίσσεται γρήγορα και αναμένεται να γίνει μια από τις πιο δημοφιλείς ανοικτού κώδικα βιβλιοθήκες χαρτογράφησης.

Αναφορές – Βιβλιογραφία κεφαλαίου

1. https://www.interaction-design.org
2. Principles of Data Visualization-What We See in a Visual, https://www.fusioncharts.com/whitepapers/downloads/Principles-of-Data-Visualization.pdf
3. Ware, C. (2012). Information Visualization: Perception for Design. 3rd Edition. Morgan Kaufmann, 1st Jun 2012. 536 pages. ISBN 0123814642
4. Dunham, M.H. (2002). Data Mining: Introductory and Advanced Topics, Prentice Hall, Upper Saddle River, New Jersey, USA.
5. Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Κόκκορας, Φ., και Σακελλαρίου, Η. (2006). Τεχνητή Νοημοσύνη (3η Έκδοση), Εκδόσεις Β. Γκιούρδας Εκδοτική, Θεσσαλονίκη.
6. Choosing the right charting component for your product, https://www.fusioncharts.com/whitepapers/downloads/Choosing-the-right-charting-component.pdf
7. http://extremepresentation.typepad.com/blog/2006/09/choosing_a_good.html
8. https://datavizcatalogue.com
9. https://medium.com/@Elijah_Meeks/2017-data-visualization-survey-results-40688830b9f2
10. https://www.microsoft.com/silverlight/pivotviewer
11. https://www.microsoft.com/silverlight
12. https://jquery.com
13. https://github.com/openlink/html5pivotviewer
14. https://docs.microsoft.com/en-us/previous-versions/windows/silverlight/dotnet-windows-silverlight/cc645050(v=vs.95)
15. http://d3js.org
16. http://flare.prefuse.org
17. http://processingjs.org
18. https://philogb.github.io/jit/index.html
19. https://powerbi.microsoft.com
20. https://jpgraph.net
21. http://www.chartjs.org

Η πρόσβαση στις ηλεκτρονικές πηγές - αναφορές επικαιροποιήθηκε τον Μάρτιο 2018.

Κεφάλαιο 3ο
Σκοπός της εργασίας

Σκοποί και Στόχοι

Όπως ήδη έχει καταγραφεί στα προηγούμενα κεφάλαια, τα τελευταία χρόνια είναι έντονο το φαινόμενο της αύξησης και συσσώρευσης βιολογικών δεδομένων σε διεθνείς βάσεις δεδομένων. Ο μεγάλος αριθμός των καταχωρημένων νουκλεοτιδικών αλληλουχιών (π.χ. ολόκληρα γονιδιώματα, ESTs DNA και mRNA) και πρωτεϊνικών αλληλουχιών, των τρισδιάστατων δομών των βιομακρομορίων, καθώς επίσης και η χαρτογράφηση και αλληλούχηση ολόκληρων γονιδιωμάτων, είχε ως αποτέλεσμα τη συγκέντρωση ενός τεράστιου όγκου ακατέργαστων δεδομένων εκτεταμένης πολυπλοκότητας.

Η πρόκληση για τους ερευνητές ώστε να ανταποκριθούν στη διαχείριση του τεράστιου όγκου δεδομένων, οδήγησε απαραίτητα στη χρήση ηλεκτρονικών υπολογιστών. Η συμβολή τους στην αποτελεσματική αποθήκευση, οργάνωση, ανάλυση και ερμηνεία αυτής της πληθώρας βιολογικών δεδομένων κρίθηκε εξαιρετικά αναγκαία. Με την ταυτόχρονη έντονη ανάπτυξη της Πληροφορικής, δημιουργήθηκαν οι κατάλληλες προϋποθέσεις για την ανάπτυξη ενός νέου και συνεχώς εξελισσόμενου επιστημονικού πεδίου, της Βιοπληροφορικής.

Επίσης, παρατηρείται ταυτόχρονη ανάπτυξη εργαλείων για την απεικόνιση των δεδομένων με σκοπό την ανάλυση και ερμηνεία τους και την εξαγωγή συμπερασμάτων.

Ο σκοπός της παρούσας εργασίας κινείται σ' αυτή την κατεύθυνση. Συγκεκριμένα, προτείνει υπολογιστικά εργαλεία τα οποία συμβάλουν στην προσπάθεια των ερευνητών για τη διαχείριση του όγκου των ερευνητικών δεδομένων τους, δίνοντάς τους ταυτόχρονα δυνατότητες ανάλυσης και ερμηνείας τους για την εξαγωγή συμπερασμάτων.

Αυτό γίνεται μέσω του σχεδιασμού και της δημιουργίας βάσεων δεδομένων, οι οποίες καθοδηγούνται αφενός από το είδος, το πλήθος και την πολυπλοκότητα των ερευνητικών δεδομένων και αφετέρου από τις ερευνητικές ανάγκες ανάλυσης και ερμηνείας των δεδομένων από τους ερευνητές. Η προσπάθεια αυτή περιλαμβάνει και την ταυτόχρονη ανάπτυξη, εφαρμογή και μελέτη εργαλείων, διαφορετικών τεχνολογιών για την απεικόνιση και εξόρυξη δεδομένων από τις βάσεις δεδομένων.

Συγκεκριμένα οι βασικοί στόχοι για την επίτευξη του σκοπού αυτής της μελέτης είναι:

(α) Η αποδοτική οργάνωση των υπαρχόντων βιολογικών δεδομένων και η πρόσβαση σε αυτά καθώς και οι δυνατότητες συσσώρευσης νέων δεδομένων.

(β) Η ανάπτυξη μεθόδων και υπολογιστικών εργαλείων οπτικοποίησης των δεδομένων με στόχο την εξαγωγή πληροφοριών από τα δεδομένα.

(γ) Η χρήση των εργαλείων αυτών για την ανάλυση και ερμηνεία των δεδομένων με ένα βιολογικά αποδεκτό τρόπο (Reichhardt, 1999).

Κεφάλαιο 4ο

Η βάση δεδομένων RGDtrip

Το τριπεπτίδιο RGD

Τα πεπτίδια διαδραματίζουν καίριο ρόλο στις θεμελιώδεις φυσιολογικές και βιοχημικές λειτουργίες της ζωής. Πεπτίδιο είναι ένα μόριο που δημιουργείται από την ένωση δύο ή περισσότερων αμινοξέων. Όταν ο αριθμός των αμινοξέων είναι μικρός (μέχρι 50 περίπου) αυτά τα μόρια ονομάζονται πεπτίδια, ενώ οι μεγαλύτερες ακολουθίες αμινοξέων αναφέρονται ως πρωτεΐνες.

Τα πεπτίδια (ή πρωτεΐνες) υπάρχουν σε κάθε ζωντανό κύτταρο και επιτελούν μια ποικιλία από βιοχημικές δραστηριότητες.

Το τριπεπτίδιο Αργινίνη-Γλυκίνη-Ασπαρτικό οξύ (RGD) είναι κάτι περισσότερο από μια τυχαία ακολουθία αμινοξέων. Περιέχει ένα μικρό και ουδέτερου φορτίου αμινοξύ (Glycine-G) μεταξύ δύο μεγαλύτερων αμινοξέων αντίθετων φορτίων. Το αρνητικά φορτισμένο (-) Ασπαρτικό οξύ (D) με πλευρική αλυσίδα καρβοξυλικού άλατος και την Αργινίνη (R), με θετικά (+) φορτισμένη ομάδα γουανιδίνης.

Η συνηθέστερη διαμόρφωση του τριπεπτιδίου RGD είναι ένας βρόχος και αυτός μπορεί να βρεθεί σε όλες τις μορφές ζωής, από τον ιό στον άνθρωπο (1), καθώς τα αμινοξέα που συνθέτουν κωδικοποιούνται από τις ίδιες τριπλέτες σε όλες τις κύριες ποικιλίες του γενετικού κώδικα (λέγοντας "κύριες" εννοούμε τις ποικιλίες των κεντρικών και όχι των περιφερειακών γονιδιωμάτων, όπου και αν υπάρχουν, όπως στα πλαστίδια ή τα μιτοχόνδρια).

Στις ομόλογες πρωτεΐνες, το τριπεπτίδιο RGD μπορεί να διατηρηθεί ή να μη διατηρηθεί, η δε τελευταία περίπτωση συνήθως παρουσιάζει μια αλλαγή ή απώλεια λειτουργικότητας του τροποποιημένου ομόλογου [1,2].

Η πρώτη εμφάνισή του ήταν μέσω πρωτεϊνών που εμπλέκονται στην κυτταρική προσκόλληση - μια εργασία που είναι πιο κοινή σε όλους τους ζώντες οργανισμούς και οντότητες [3, 4] αλλά αυτό το τριπεπτίδιο μπορεί να είναι κάτι περισσότερο από αυτό.

Ο σχηματισμός βρόχου, με τις πλευρικές αλυσίδες που προσδίδουν δραστικότητα D και R που κοιτάζουν προς τα έξω και μακριά ο ένας από τον άλλο σχηματίζοντας μία πολύ ενεργή και πολύ διακριτή ηλεκτροχημική οντότητα 3-D, είναι αναγνωρίσιμη και περιέχει θετικό και αρνητικό φορτίο, σε pH = 7 [4], παρόλο που ο βρόχος ως σύνολο είναι ουδέτερος [5, 6]. Αυτός ο συνδυασμός επιτρέπει ένα ευρύ φάσμα αλληλεπιδράσεων σύνδεσης, σηματοδότησης και αγκύρωσης, γεγονός που εξηγεί το ρόλο των βρόχων RGD στη μεταγωγή σήματος καθώς μπορεί να σταθεροποιήσει την τοπολογία διαφορετικών οντοτήτων συνδέτη. Βρίσκεται συνήθως στις διαμεμβρανικές πρωτεΐνες, αλλά και στις περιφερειακές μεμβρανοειδείς, συνήθως στραμμένες προς τα έξω (εξωκυτταρικά) για προφανείς λόγους πρόσφυσης.

Επίσης, τα κυκλικά πεπτίδια RGD που εγχύονται σε κύτταρα σε nM συγκεντρώσεις επηρεάζουν σημαντικά τον κυτταρικό μετασχηματισμό, υποδηλώνοντας ότι πρέπει επίσης να υπάρχουν και λειτουργούν ενδοκυτταρικοί RGD βρόχοι [6].

Σε εκκρινόμενη μορφή, μπορεί να υποβοηθήσει τη σύνδεση της μεταφερόμενης πρωτεΐνης με τον στόχο της όπως στην περίπτωση της διαλυτής πρωτεΐνης δέσμευσης IGF [1].

Όταν παρουσιάστηκε η πρώτη βιβλιογραφική ανασκόπηση και αξιολόγηση όλων των αλληλουχιών RGD και βρόχων σε υποδοχείς μεταξύ των ειδών το 1998 [1], γεννήθηκε ο προβληματισμός αν η εμφάνιση τέτοιων αλληλουχιών σε υποδοχείς θα μπορούσε να σημαίνει ότι αυτές οι αλληλουχίες θα ήταν σε δομές τύπου βρόχου / βρόχου και ότι αυτές οι δομές θηλειάς θα

μπορούσαν να συνεπάγονται επίσης λειτουργία κυτταρικής προσκόλλησης για υποδοχείς.

Στο επόμενο χρονικό διάστημα, αυτό έχει επαληθευτεί σε πολλές περιπτώσεις, προσθέτοντας περαιτέρω στην ενδιαφέρουσα υπόθεση ότι οι βρόχοι RGD μπορούν να προσθέσουν μια συνάρτηση κυτταρικής προσκόλλησης στους υποδοχείς. Για παράδειγμα, η αλληλουχία HLA-DQβ167-169RGD που απαντάται σε διάφορα αλληλόμορφα, προβλέπεται να είναι σε βρόχο [7, 8] και στη συνέχεια αποδείχθηκε ότι είναι σε ένα τέτοιο βρόχο με κρυσταλλογραφία τεσσάρων διαφορετικών αλληλόμορφων [9, 10, 11 , 12], με δύο δομές αλληλόμορφων που προσδιορίστηκαν δύο φορές από διαφορετικές ομάδες [10, 13] και άλλες δύο παρουσιάζουν την ίδια διαμόρφωση όταν συνδέονται με έναν συγγενή υποδοχέα Τ κυττάρων [11, 13, 14]. Η λειτουργία κυτταρικής προσκόλλησης των αλληλουχιών RGD σε υποδοχείς έχει δειχθεί σε μία αξιοσημείωτη περίπτωση: ο υποδοχέας νουκλεοτιδίων P2Y2 που είναι επαγώγιμος στον εγκέφαλο με φλεγμονή με τη μεσολάβηση IL-1b διαθέτει μια RGD αλληλουχία στον πρώτο εξωκυτταρικό βρόχο της και κατά την ενεργοποίηση με UTP ο υποδοχέας P2Y2 αυξάνει την έκφραση των αVβ3/5 integrins in astrocytes που με τη σειρά τους συνδέονται απευθείας στον υποδοχέα P2Y2 χρησιμοποιώντας αυτή την RGD αλληλουχία [15].

Η εκμετάλλευση της αλληλεπίδρασης RGD έχει οδηγήσει σε επιτυχείς κλινικές εφαρμογές στην καρδιαγγειακή ιατρική [16], την απεικόνιση όγκων στον άνθρωπο [17, 18], τη στόχευση κυτοκινών σε περιοχές ανθρώπινου όγκου [19] και την επιφανειακή τροποποίηση εμφυτευμάτων μακρόσωμων οστών σε μικρά και μεγάλα ζώα μοντέλα [20]. Υπάρχουν επίσης αρκετές κλινικές δοκιμές στη Φάση ΙΙ και τη Φάση ΙΙΙ για φαρμακευτικά αντικαρκινικά φάρμακα με βάση RGD που στοχεύουν σε συγκεκριμένους όγκους του ανθρώπου, αν και αυτό το πεδίο αντιμετωπίζεται με δυσκολίες, όπως έδειξε μια αποτυχημένη μελέτη Στάδιο ΙΙΙ [21].

Αυτή η προβολή μπορεί να είναι κοντόφθαλμη. Το δυναμικό στερεοδιαμόρφωσης του τριπεπτιδίου θα μπορούσε να επιτρέψει την καμπυλότητα ενός κλώνου πρωτεΐνης ή έλικας, ως εύκαμπτη κεφαλή

παραμόρφωσης στην μηχανική [1]. Η δυνατότητα εφαρμογής αυτού του είδους δομής μπορεί να είναι μια θέση αναγνώρισης / δέσμευσης [22] (όπως στην περίπτωση πολλών υποδοχέων, για να αναφέρουμε μόνο τον υποδοχέα επιδερμικού αυξητικού παράγοντα [3] και την -διαλυτή-πρωτεΐνη-1 δέσμευσης IGF σε ανθρώπους [1]), ως εύκαμπτη άρθρωση για την αλλαγή της κατεύθυνσης ενός κλώνου ή έλικας - ίσως στην περίπτωση της πυροσταφυλικής κινάσης [1] - ή ως μια συσκευή σύλληψης για ασφαλή σύνδεση, όπως στη Fibronectin [4] HIV ιός [1]. Όλες αυτές οι προτεινόμενες ή βεβαιωμένες λειτουργίες ρυθμίζονται από γειτονικά αμινοξέα, τα οποία μπορεί να παρέχουν είτε ευκαμψία είτε ακαμψία. Ο πρώτος μεγεθύνει τον φάκελο αλληλεπίδρασης / αναγνώρισης ενώ ο τελευταίος περιορίζει και το περιέχει [1].

Η εξαιρετική διασπορά αυτού του πεπτιδικού σχεδίου σε ολόκληρη τη βιόσφαιρα [1, 22] και το πλήθος μεμονωμένων εργασιών και εφαρμογών που είναι εγγενείς σε μια τέτοια προσαρμόσιμη και ευέλικτη δομή [6, 9] δημιουργούν την ανάγκη για μαζική σύγκριση δεδομένων και προσπάθεια εξόρυξης γνώσης, αν είναι να διασαφηνιστεί ο ρόλος της στα βιολογικά συστήματα και τη βιομηχανία.

Φυλογενετική και συγκριτική λειτουργική έρευνα, η οποία τεκμηριώνει τις εξελικτικές τάσεις (διατήρηση, σύγκλιση και απόκλιση) λειτουργικών προτύπων και ανιχνεύει εναλλακτικές λύσεις, απαιτεί ισχυρές αλλά διαισθητικές και φιλικές προς τον χρήστη μαζικές δοκιμές σύγκρισης για την επίτευξη αρχικών αποτελεσμάτων συσχετισμού.

Προκειμένου να επιτευχθεί η διερεύνηση του προαναφερθέντος πεδίου, οι ερευνητές συλλέγουν τις σχετικές πληροφορίες σε μεγάλα υπολογιστικά φύλλα, τα οποία δεν παρέχουν καμία δυνατότητα διερεύνησης, ανάλυσης και κατανόησης των δεδομένων κατά τρόπο φιλικό προς τον χρήστη. Ως αποτέλεσμα, υπάρχει μια επιτακτική ανάγκη να αντιπροσωπεύονται οι πληροφορίες σε οπτική μορφή, επιτρέποντας στους ερευνητές να αναλύουν τα δεδομένα και να αποκτούν ουσιαστικές γνώσεις, αποκαλύπτοντας τα υποκείμενα μοντέλα και ενδεχομένως, προηγούμενες αόρατες συσχετίσεις μεταξύ μεγάλων συνόλων δεδομένων. Έτσι, εντοπίζεται μια ανάγκη για την

απεικόνιση των δεδομένων, προσφέροντας στους πιθανούς χρήστες τη δυνατότητα να συλλέγουν δεδομένα από διάφορες πηγές και έτσι να ανιχνεύουν πρότυπα και συσχετισμούς [23].

Αυτό άλλωστε, αποτέλεσε το κίνητρο για τη δημιουργία και την κατάρτιση της βάσης δεδομένων RGDtrip (διαθέσιμη στη διεύθυνση http://www.biodata.gr/rgdtrip). Το περιβάλλον απεικόνισης δεδομένων από τη συλλογή πρωτεϊνών στην RGDtrip, προσφέρει στο χρήστη διαισθητική αλληλεπίδραση με τα δεδομένα έτσι ώστε να χειρίζεται υψηλό όγκο δεδομένων διατηρώντας παράλληλα μια προοπτική.

Η αρχική έκδοση βάσης δεδομένων τριπεπτιδίων RGD (RGDtrip) περιελάμβανε κάποια βασική λειτουργικότητα απεικόνισης δεδομένων, η οποία δεν επέτρεπε τη συσχέτιση και την απεικόνιση δεδομένων.

Στην εξέλιξη και ανάπτυξη της εφαρμογής, δημιουργήθηκε ένα ώριμο και εξελιγμένο διαδραστικό εργαλείο δημιουργίας ερωτημάτων και οπτικοποίησης των αποτελεσμάτων, που βασίζεται στον παγκόσμιο ιστό, το οποίο επιτρέπει στους χρήστες να συνδυάζουν μεγάλες ομάδες παρόμοιων στοιχείων και να εντοπίζουν κρυφές σχέσεις μεταξύ μεμονωμένων πληροφοριών. Αυτό το εργαλείο ανταποκρίνεται σε μία από τις σημαντικές προκλήσεις μεγάλων και σύνθετων συνόλων δεδομένων, δηλαδή την αποτελεσματική παρουσίαση και αλληλεπίδραση με τα δεδομένα. Συγκεκριμένα, δημιουργήσαμε ένα κομψό, web-based multimedia front-end, βασισμένο σε ένα λογισμικό που ξεκίνησε από τη Microsoft, το PivotViewer (Microsoft), προκειμένου να υποστηριχθεί η υψηλού επιπέδου οπτικοποίηση της συλλογής δεδομένων και της διαδικασίας εξόρυξης. Υποστηρίζει δε, οπτική απεικόνιση δεδομένων, διαλογή, οργάνωση και κατηγοριοποίηση.

Συλλογή δεδομένων

Όλα τα δεδομένα που καταχωρήθηκαν στην RGDtrip, αρχικά συλλέχθηκαν από άλλες βάσεις δεδομένων, όπως UNIProt, PDBdb, κτλ.

Η κατασκευή της RGDtrip ακολούθησε μια απλή προσέγγιση, ξεκινώντας από τη συλλογή δεδομένων και τον προσδιορισμό των απαιτήσεων με αποτέλεσμα το πρωτότυπο της εφαρμογής. Τα δεδομένα που συλλέχθηκαν από τις διαθέσιμες στο κοινό βάσεις δεδομένων Uniprot και PDB υποβλήθηκαν αρχικά σε μια διαδικασία ομογενοποίησης για να συμμορφωθούν με το μοντέλο οντοτήτων της βάσης δεδομένων RGDtrip.

Εργαλεία ανάπτυξης λογισμικού

Η διαδικασία ανάπτυξης λογισμικού είναι μια ιδιαίτερα ακριβή και χρονοβόρα διαδικασία. Γίνεται άμεσα αντιληπτό ότι η απόδοση παραγωγής κώδικα είναι ένας ιδιαίτερα σημαντικός παράγοντας και ενώ υπάρχουν διάφορα εργαλεία που μπορούν να την αυξήσουν σημαντικά, το κόστος τους είναι ιδιαίτερα υψηλό (συνήθως, η απόφαση αγοράς τους έρχεται ως αποτέλεσμα ανάλυσης κόστους/ωφέλειας). Είναι ιδιαίτερα σημαντικό, και πρέπει να σημειωθεί ότι ένα από τα πλεονεκτήματα του ακαδημαϊκού χώρου, είναι η δωρεάν πρόσβαση σε ένα μεγάλο αριθμό από τέτοια εργαλεία. Χάρη στη δυνατότητα αυτή, η επιλογή αυτών που χρησιμοποιήθηκαν έγινε καθαρό, με γνώμονα την μεγιστοποίηση της απόδοσης αγνοώντας το κόστος, το οποίο κάτω από κανονικές συνθήκες θα ήταν απαγορευτικό.

Microsoft SQL Server 2008

Το Microsoft SQL Server είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων το οποίο κατασκευάστηκε από την Microsoft. Για την ανάπτυξη της βάσης μας χρησιμοποιήθηκε η έκδοση SQL Server 2008 Enterprise R2 η οποία παρέχεται δωρεάν από το πανεπιστήμιο Πατρών. Χρησιμοποιείται σε περιβάλλον Windows και παρέχει μια πληθώρα από εργαλεία που επιτρέπουν την εύκολη ανάπτυξη και διαχείριση βάσεων δεδομένων. Υπάρχουν και άλλες, υλοποιήσεις ελεύθερης πρόσβασης σχεσιακών βάσεων όπως η MySQL αλλά δεδομένου ότι δεν είχαμε να επωμιστούμε το κόστος αγοράς του SQL Server, δεν υπήρχε λόγος να τις χρησιμοποιήσουμε.

Microsoft Visual Studio 2008-2012

Για την ανάπτυξη του προγράμματος ενοποίησης των δεδομένων καθώς και του γραφικού περιβάλλοντος της βάσης, χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE. Integrated Development Environment) Microsoft Visual Studio 2010 Ultimate και στην συνέχεια η αντίστοιχη 2012 έκδοση, τα οποία όπως και ο SQL Server παρέχονται δωρεάν στα μέλη του πανεπιστημίου Πατρών. Σε ένα ολοκληρωμένο περιβάλλον ανάπτυξης υπάρχουν συγκεντρωμένα απαραίτητα αλλά και βοηθητικά εργαλεία ανάπτυξης ενός λογισμικού. Η χρησιμοποίηση ενός επαγγελματικού εργαλείου ανάπτυξης λογισμικού παρέχει ένα επιπρόσθετο σύνολο εργαλείων οργάνωσης, διαχείρισης και βελτιστοποίησης πηγαίου κώδικα, η χρήση των οποίων ήταν απαραίτητη δεδομένου ότι για την εργασία αυτή απαιτήθηκε η παραγωγή ενός σημαντικού μεγέθους λογισμικού. Στο περιβάλλον αυτό, μπορούν να χρησιμοποιηθούν διάφορες προγραμματιστικές γλώσσες, όπως οι C#, Visual C++, Visual Basic, F# και Python.

C#

Η γλώσσα που χρησιμοποιήθηκε ήταν η C#, η οποία δημιουργήθηκε από τις εταιρίες Hewlett-Packard, Intel, και Microsoft το 2001. Αν και κατασκευάστηκε με σκοπό να λειτουργεί σε περιβάλλον Windows, σήμερα, υπάρχουν υλοποιήσεις της και για Linux και Macintosh. Είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού. σε σύγκριση με άλλες όπως η C ή η Assembly, το οποίο σημαίνει ότι μπορεί να περιγράψει τις ίδιες διαδικασίες με πολύ μικρότερο αριθμό εντολών. Συνήθως, για τον ίδιο προγραμματιστή, η ταχύτητα συγγραφής κώδικα και η πιθανότητα εισαγωγής σφαλμάτων είναι ίδιες ανεξαρτήτως γλώσσας. Αυτό σημαίνει ότι με τις γλώσσες υψηλού επιπέδου ένας προγραμματιστής μπορεί να περιγράψει περισσότερες λειτουργίες στον ίδιο χρόνο και περιγράφοντας την ίδια ποσότητα λειτουργιών να εισάγει μικρότερο αριθμό σφαλμάτων. Η γλώσσα αυτή είναι εφάμιλλη των γλωσσών

Java ή Python και επιλέχθηκε για λόγους προτίμησης και εμπειρίας του προγραμματιστή.

Αρχιτεκτονική συστήματος και δομή βάσης δεδομένων

Η λειτουργία της RGDtrip, στηρίζεται σε μια σχεσιακή βάση δεδομένων που αναπτύχθηκε με τον Microsoft SQL Server, ένα ευέλικτο προϊόν λογισμικού που προσφέρει προηγμένες δυνατότητες για ανάπτυξη βάσεων δεδομένων, διαχειρισιμότητα και αποθήκευση δεδομένων.

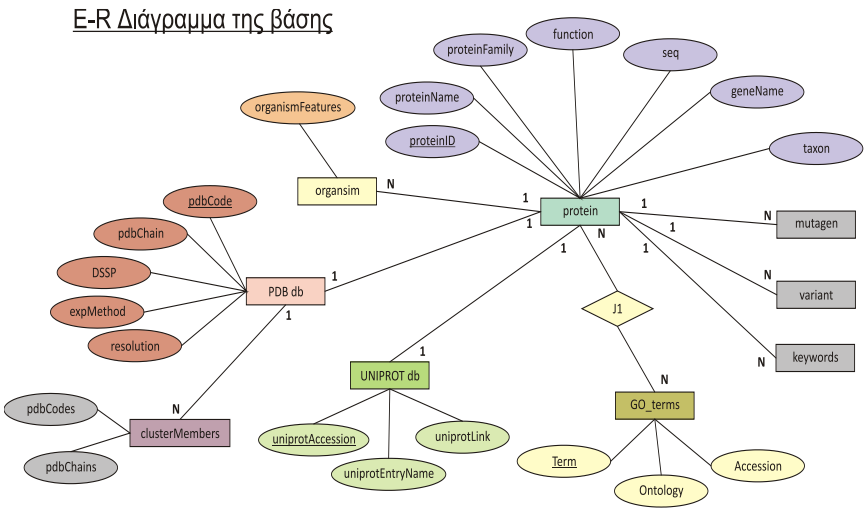
Μοντέλο Οντοτήτων-Συσχετίσεων

Το μοντέλο οντοτήτων-συσχετίσεων (μοντέλο Ο/Σ ή E-R model) είναι ένα αφαιρετικό μοντέλο δεδομένων το οποίο έχει καθορισμένη δομή. Χρησιμοποιείται για να παρέχει ένα εννοιολογικό σχήμα κατά τη σχεδίαση βάσεων δεδομένων, ως μοντέλο δεδομένων ενός συστήματος και των απαιτήσεών του με top-down προσέγγιση. Ένα διάγραμμα που δημιουργείται με αυτή τη διαδικασία σχεδίασης λέγεται διάγραμμα οντοτήτων-συσχετίσεων (διάγραμμα Ο/Σ ή ΟΣΔ). Προτάθηκε αρχικά το 1976 από τον Peter Chen, ωστόσο στη συνέχεια επινοήθηκαν πολλές παραλλαγές της διαδικασίας. Σκοπός του είναι να περιγράφει τις αναγκαίες πληροφορίες οι οποίες πρόκειται να αποθηκευτούν στη βάση δεδομένων ή τον τύπο τους και χρησιμοποιείται στο πρώτο στάδιο σχεδίασης ενός συστήματος πληροφοριών, κατά την ανάλυση των απαιτήσεων του. Βάση για τα μοντέλα Ο/Σ είναι η κατηγοριοποίηση αντικειμένων και των σχέσεων τους μεταξύ τους.

• Οντότητα (entity) είναι ένα αντικείμενο ενδιαφέροντος στον πραγματικό κόσμο το οποίο ξεχωρίζει από τα υπόλοιπα. Μια οντότητα λειτουργεί αφαιρετικά σε έναν πολύπλοκο τομέα. Οντότητες μπορεί να είναι άνθρωποι, μέρη, αντικείμενα, γεγονότα, έννοιες κλπ.

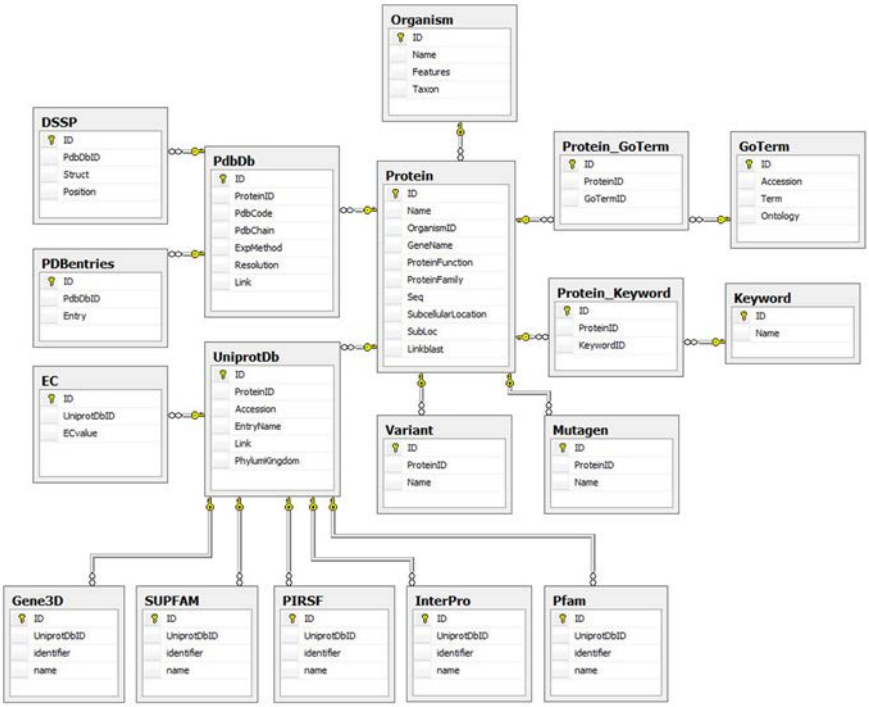
• Συσχέτιση (relationship) είναι η σύνδεση δύο ή περισσότερων τύπων οντοτήτων που παρουσιάζει ενδιαφέρον για σχεδιασμό. Με συσχετίσεις μπορούν να συνδέονται και χαρακτηριστικά οντοτήτων.

Τα δεδομένα που έχουν συλλεχθεί από άλλες βάσεις βιολογικών δεδομένων, όπως UNIprot, PDBdb, Gene3D, SUPfam, Pfam, PIRSF, InterPro περιέχουν στοιχεία που αναφέρονται σε πρωτεΐνες, όπως, το όνομα πρωτεΐνης, ο αντίστοιχος οργανισμός, κωδικοί που ταυτοποιούν την πρωτεΐνη, λειτουργίες κά. Μετά τη μελέτη των δεδομένων έγινε σχεδίαση του Ο-Σ διαγράμματος, που εμφαίζεται παρακάτω:



Εικόνα 29 Διάγραμμα οντοτήτων-συσχετίσεων (E-R) της βάσης

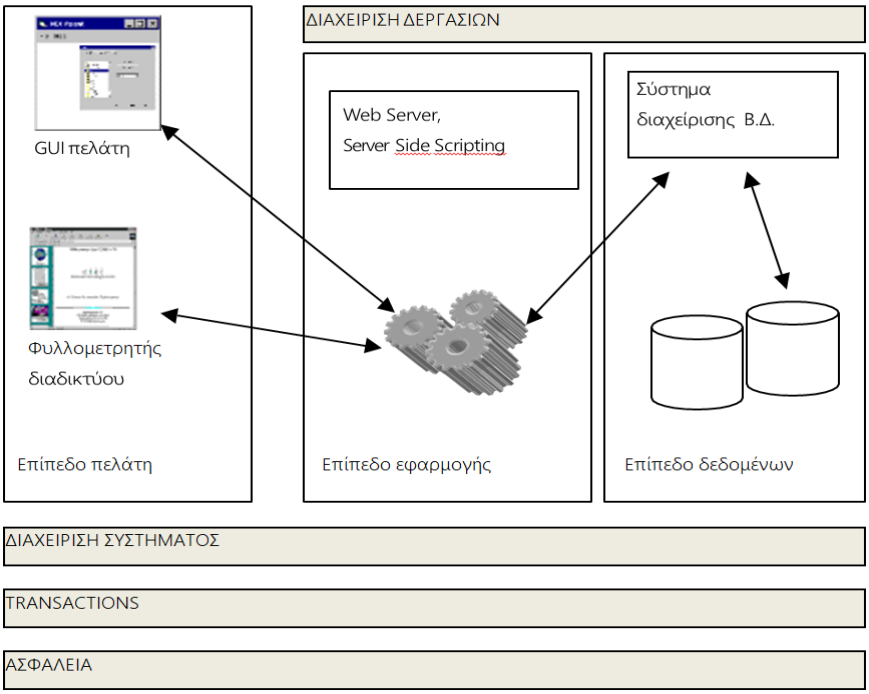
Από το παραπάνω Ο-Σ διάγραμμα, προέκυψε το τελικό σχήμα της βάσης δεδομένων, όπως απεικονίζεται στο σχήμα 1.



Εικόνα 30 Το σχήμα (scheme) της βάσης

Η συνολική αρχιτεκτονική του συστήματος βασίζεται σε ένα μοντέλο πελάτη-διακομιστή τριών επιπέδων [33], που περιλαμβάνει τρία βασικά στοιχεία: την εφαρμογή πελάτη, τον διακομιστή εφαρμογών και τον διακομιστή βάσης δεδομένων.

Η αρχιτεκτονική τριών επιπέδων προορίζεται να επιτρέψει την αναβάθμιση ή την αντικατάσταση οποιασδήποτε από τις βαθμίδες ανεξάρτητα από τις αλλαγές στις απαιτήσεις ή την τεχνολογία. Η εφαρμογή πελάτη περιέχει μόνο τη λογική παρουσίασης. Ως αποτέλεσμα, απαιτούνται λιγότεροι πόροι από το τμήμα του σταθμού εργασίας του χρήστη (πελάτη) και δεν απαιτείται τροποποίηση του χρήστη σε περίπτωση αλλαγής της θέσης της βάσης δεδομένων. Οι αλλαγές στην επιχειρησιακή λογική επιβάλλονται αυτόματα από το διακομιστή και οι πιθανές μελλοντικές αλλαγές περιορίζονται στο λογισμικό του διακομιστή εφαρμογών που θα πρέπει να εγκατασταθεί.



Εικόνα 31 Αρχιτεκτονική τριών επιπέδων

Η αρχιτεκτονική τριών επιπέδων είναι ένα ισχυρό μοντέλο, αρκετά ευέλικτο ώστε να συγκεντρώνει πολλαπλές πηγές πληροφοριών και να ενσωματώνει την αρθρωτή ανάπτυξη [25]. Η διεπαφή χρήστη της βαθμίδας των εφαρμογών-πελάτη, η λογική της λειτουργικής διαδικασίας ("επιχειρησιακή λογική") του επιπέδου εξυπηρετητή εφαρμογών και η αποθήκευση δεδομένων υπολογιστή και η πρόσβαση στα δεδομένα (και τα δύο επίπεδα διακομιστή βάσης δεδομένων) αναπτύσσονται και διατηρούνται ως ανεξάρτητες μονάδες.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν εφαρμογές της Microsoft. Συγκεκριμένα, για την αποθήκευση δεδομένων στο «Επίπεδο Δεδομένων» χρησιμοποιείται το σχεσιακό μοντέλο δεδομένων και συγκεκριμένα το σύστημα διαχείρισης βάσεων δεδομένων Microsoft SQL Server της Microsoft. Το «Επίπεδο εφαρμογής» υλοποιήθηκε από τον εξυπηρετητή διαδικτύου Internet Information Services (IIS) για Windows® Server. Είναι ένας ευέλικτος, ασφαλής και διαχειρίσιμος εξυπηρετητής για δημοσίευση περιεχομένου στο web. Τέλος, το PivotViewer χρησιμοποιήθηκε για την

υλοποίηση της κύριας διεπαφής της εφαρμογής της RGDtrip, στο «Επίπεδο Πελάτη».

Η πρωταρχική συλλογή της βάσης δεδομένων κατηγοριοποιήθηκε με βάση την (υπο)κυτταρική θέση (subcellular Location) εμφάνισης της πρωτεΐνης, αφού η θέση του υποκυτταρικού επιπέδου είναι θεμελιώδης για τη λειτουργία της πρωτεΐνης. Μια εξεζητημένη σειρά κριτηρίων φιλτραρίσματος επιτρέπει περαιτέρω χειρισμούς των πρωτεϊνών.

Σχεδιασμός διεπαφής εφαρμογής

Οι αναζητήσεις με τα ερωτήματα στο σύνολο των πρωτεϊνών της RGDtrip, μπορούν να εκτελεστούν χρησιμοποιώντας τον εργαλείο PivotViewer [26, 27]. Πρόκειται για plug-in προγράμματος περιήγησης που στηρίζεται στην εφαρμογή Silverlight. Το Microsoft Silverlight είναι ένα πλαίσιο εφαρμογής για τη σύνταξη και εκτέλεση εφαρμογών Rich Internet, με χαρακτηριστικά και σκοπούς παρόμοια με εκείνα του Adobe Flash.

Το PivotViewer χρησιμοποιήθηκε για την υλοποίηση της κύριας διεπαφής της εφαρμογής της RGDtrip, καθώς χρησιμοποιεί το Deep Zoom, το οποίο είναι η ταχύτερη, ομαλότερη τεχνολογία ζουμ στο διαδίκτυο.

Ως αποτέλεσμα, εμφανίζει περιεχόμενο πλήρους και υψηλής ανάλυσης χωρίς μεγάλους χρόνους φόρτωσης, ενώ οι κινούμενες εικόνες και οι φυσικές μεταβάσεις παρέχουν το περιβάλλον και εμποδίζουν τους χρήστες να αισθάνονται υπερφορτωμένοι από μεγάλες ποσότητες πληροφοριών. Το PivotViewer επιτρέπει στους χρήστες να αλληλεπιδρούν με χιλιάδες αντικείμενα ταυτόχρονα και να ταξινομούν τα δεδομένα με τρόπο που τους βοηθάει να βλέπουν τις τάσεις και να βρίσκουν γρήγορα αυτό που ψάχνουν.

Η απεικόνιση στην εξόρυξη δεδομένων είναι μια καινοφανής και πολλά υποσχόμενη προσέγγιση για την επεξήγηση των δεδομένων, γνωστή ως Οπτική Εξόρυξη Δεδομένων. Προέκυψε από την τεχνολογική σύζευξη αυτοματοποιημένων αλγορίθμων εξόρυξης δεδομένων και τεχνικών απεικόνισης.

Εισαγωγή συλλεγμένων δεδομένων στη βάση

Τα δεδομένα των πρωτεϊνών έχουν εισαχθεί στη βάση δεδομένων RGDtrip μέσω αρχείων ASCII (απλού κειμένου). Ένα δείγμα των στοιχείων μιας πρωτεΐνης που είναι αποθηκευμένα στο αρχείο κειμένου φαίνεται παρακάτω.

```
>uniprotAccession
Q16647
>uniprotLink
http://www.uniprot.org/uniprot/Q16647
>uniprotEntryName
PTGIS_HUMAN
>proteinName
Prostacyclin synthase
>geneName
PTGIS
>organism
Homo sapiens (Human)
>taxon
Eukaryota
>function
Catalyzes the isomerization of prostaglandin H2 to prostacyclin (= prostaglandin
I2).
>proteinFamily
cytochrome P450 family
>GO_terms
GO:0001516      prostaglandin biosynthetic process       biological process
GO:0004497      monooxygenase activity   molecular function
GO:0005788      endoplasmic reticulum lumen      cellular component
GO:0005789      endoplasmic reticulum membrane   cellular component
GO:0008116      prostaglandin-I synthase activity        molecular function
GO:0009055      electron carrier activity        molecular function
GO:0016021      integral to membrane     cellular component
GO:0020037      heme binding     molecular function
>keywords
3D-structure
Complete proteome
Endoplasmic reticulum
Fatty acid biosynthesis
Heme
Iron
Isomerase
Lipid synthesis
Membrane
Metal-binding
Polymorphism
Prostaglandin biosynthesis
Transmembrane
```

```
Transmembrane helix
>variant
variant1: 379 R-S (in allele CYP8A1*4; dbSNP:rs56195291). /FTId=VAR_010917.
>mutagen
NA
>seq
MAWAALLGLLAALLLLLLLSRRRTRRPGEPPLDLGSIPWLGYALDFGKDAASFLTRMKEKHGDIFTILVGGRYVTVLLDPH
SYDAVVWEPRTRLDFHAYAIFLMERIFDVQLPHYSPSDEKARMKLTLLHRELQALTEAMYTNLHAVLLGDATEAGSGWHEM
GLLDFSYSFLLRAGYLTLYGIEALPRTHESQAQDRVHSADVFHTFRQLDRLLPKLARGSLSVGDKDHMCSVKSRLWKLLSP
ARLARRAHRSKWLESYLLHLEEMGVSEEMQARALVLQLWATQGNMGPAAFWLLLFLLKNPEALAAVRGELESILWQAEQPV
SQTTTLPQKVLDSTPVLDSVLSESLRLTAAPFITREVVVDLAMPMADGREFNLRRGDRLLLFPFLSPQRDPEIYTDPEVFK
YNRFLNPDGSEKKDFYKDGKRLKNYNMPWGAGHNHCLGRSYAVNSIKQFVFLVLVHLDLELINADVEIPEFDLSRYGFGLM
QPEHDVPVRYRIRP
>pdbCode
3B6H
>pdbChain
B
>DSSP
RGD     TT_     379 380 381
>title
CRYSTAL STRUCTURE OF HUMAN PROSTACYCLIN SYNTHASE IN COMPLEX WITH INHIBITOR
MINOXIDIL
>fragment
UNP RESIDUES 18-500
>expMethod
X-RAY DIFFRACTION
>resolution
1.62
>clusterMembers
3B6H    A
3B6H    B
2IAG    A
2IAG    B
$$$$
```

Αυτά τα δεδομένα συλλέχθηκαν από διαφορετικές βάσεις δεδομένων, όπως την Uniprot και την PDB και μετατράπηκαν σε αυτή τη μορφή απλού κειμένου. Η μορφή που έχει χρησιμοποιηθεί στο παραπάνω αρχείο είναι η εξής.

Κάθε εγγραφή τελειώνει με τα σύμβολα $$$$

Κάθε γραμμή που αρχίζει με το σύμβολο > περιγράφει την πληροφορία που υπάρχει στην επόμενη γραμμή ή στις επόμενες γραμμές. Όταν δεν υπάρχει κάποια πληροφορία, είτε απουσιάζει το αντίστοιχο πεδίο είτε γράφεται NA (not available). Στη συνέχεια περιγράφονται τα πεδία που τελικά εισήχθησαν στη βάση δεδομένων.

```
>uniprotAccession
```

Ο χαρακτηριστικός κωδικός της βάσης δεδομένων UNIPROT.

```
π.χ. Q16647
>uniprotLink
```

Το link στη βάση δεδομένων UNIPROT.

```
π.χ. http://www.uniprot.org/uniprot/Q16647
>proteinName
```

Το όνομα της πρωτεΐνης.

```
π.χ. Prostacyclin synthase
>geneName
```

Το όνομα του γονιδίου.

```
π.χ. PTGIS
>organism
```

Ο οργανισμός από τον οποίο προέρχεται η συγκεκριμένη πρωτεΐνη. Σε ορισμένες περιπτώσεις, μπορεί να εμφανίζεται μόνο το η επιστημονική ονομασία (κατά Λινναίο) ενώ σε άλλες μπορεί να εμφανίζεται επιπλέον σε παρένθεση η κοινή ονομασία (common name).

```
π.χ. Homo sapiens ή Homo sapiens (Human)
>taxon
```

Το αντίστοιχο Superkingdom (επικράτεια).

```
π.χ. Eukaryota
>function
```

Η λειτουργία της πρωτεΐνης.

```
π.χ. Catalyzes the isomerization of prostaglandin H2 to prostacyclin (=
prostaglandin I2).
>proteinFamily
```

Η οικογένεια στην οποία ανήκει η πρωτεΐνη.

```
π.χ. cytochrome P450 family
>GO_terms
```

Οι όροι Οντολογίας Γονιδίων (Gene Ontology Terms). Σε κάθε γραμμή, καταχωρούνται τα Accession, Term, Ontology διαχωρισμένα με tabs.

π.χ.

```
GO:0001516      prostaglandin biosynthetic process       biological process
GO:0004497      monooxygenase activity   molecular function
GO:0005788      endoplasmic reticulum lumen      cellular component
>keywords
```

Λέξεις κλειδιά.

π.χ.

```
Isomerase
Lipid synthesis
Transmembrane helix
>variant
```

Αναφέρεται σε πολυμορφισμούς, μεταλλάξεις που σχετίζονται με ασθένειες κά. Σημειώνονται ο αριθμός του καταλοίπου στην αλληλουχία, το είδος της μετάλλαξης καθώς και τυχόν υπάρχουσες πληροφορίες.

π.χ.

```
variant1: 386 R-C (in MPS4A; severe form). /FTId=VAR_007228. R-H (in MPS4A).
/FTId=VAR_024913.
variant2: 388 D-N (in MPS4A). /FTId=VAR_024914.
>mutagen
```

Αναφέρεται σε πειραματικές μεταλλάξεις και σημειώνονται ο αριθμός του καταλοίπου στην ακολουθία, το είδος της μετάλλαξης καθώς και τα αποτελέσματα της μετάλλαξης.

π.χ.

```
mutagen1: 646 R-A,H: No binding to transferrin. R-K: 5% binding to transferrin.
mutagen2: 647 G-A: Large effect on affinity for transferrin. 4-fold reduced
affinity for HFE.
mutagen3: 648 D-A: 16% binding to transferrin. D-E: 57% binding to transferrin.
>seq
```

Η αλληλουχία της πρωτεΐνης.

```
π.χ.
MAWAALLGLLAALLLLLLLSRRRTRRPGEPPLDLGSIPWLGYALDFGKDAASFLTRMKEKHGDIFTILVGGRYVTVLLDPH
SYDAVVWEPRTRLDFHAYAIFLMERIFDVQLPHYSP
>pdbCode
```

Ο χαρακτηριστικός κωδικός της βάσης δεδομένων PDB. Το αντίστοιχο link σχηματίζεται προσθέτοντας τον κωδικό στο τέλος του string http://www.pdb.org/pdb/explore/explore.do?structureId=

π.χ. 3B6H

```
http://www.pdb.org/pdb/explore/explore.do?structureId=3B6H
>pdbChain
```

Η αλυσίδα της συγκεκριμένης εγγραφής της PDB.

π.χ. B

```
>DSSP
```

Η δευτεροταγής δομή και η αρίθμηση του RGD κατά PDB.

π.χ.

```
RGD     _SS     74C 74D 74E
RGD     TT_     21 22 23
RGD     TT_     179 180 181
>expMethod
```

Η πειραματική μέθοδος για τον προσδιορισμό της δομής.

```
π.χ. X-RAY DIFFRACTION
>resolution
```

Η διακριτική ικανότητα στην οποία έχει λυθεί η δομή.

π.χ. 1.62

```
>clusterMembers
```

Τα μέλη της PDB που έχουν υψηλή ομοιότητα σε επίπεδο ακολουθίας (90%) και ανήκουν στο ίδιο cluster. Σημειώνονται τα pdbCodes και pdbChains.

π.χ.

```
3B6H    A
3B6H    B
2IAG    A
2IAG    B
```

Σε μία εγγραφή ενδέχεται να υπάρχουν δεδομένα μόνο από την PDB, μόνο από την UNIPROT (η πλειονότητα των εγγραφών), ή και από τις δύο βάσεις δεδομένων. Επίσης σε κάποιες περιπτώσεις, ένας κωδικός UNIPROT αντιστοιχεί σε δύο αλυσίδες της PDB (όταν έχουν διευκρινιστεί διαφορετικά τμήματα της πρωτεΐνης που περιέχουν και τα δύο το τριπεπτίδιο RGD).

Στη συνέχεια δημιουργήθηκε απαραίτητο λογισμικό σε γλώσσα προγραμματισμού C#, το οποίο δέχεται ως αρχείο εισόδου το παραπάνω αρχείο κειμένου με όλα τα δεδομένα των πρωτεϊνών και τα εισάγει στους κατάλληλους πίνακες της βάσης δεδομένων.

Οπτικοποίηση δεδομένων με το PivotViewer

Αρχικά το σύνολο των πρωτεϊνών ομαδοποιήθηκαν σε ένα σημαντικό αριθμό κατηγοριών με βάση την κυτταρική τοποθεσία (74) που εντοπίζεται η πρωτεΐνη. Ωστόσο, ένας σημαντικός αριθμός εγγραφών (12.271) παραμένουν σε μια «απροσδιόριστη» κατηγορία και παρουσιάζονται συλλογικά ως ενιαία κάρτα, μιας και δεν προσδιορίζεται η θέση εμφάνισής τους.

Ο μεγάλος όγκος και η πολυπλοκότητα της συλλογής δεδομένων των πρωτεϊνών RGDtrip, καθώς και οι ποικίλες και πολυάριθμες σχέσεις μεταξύ των δεδομένων, καθιστούν δύσκολο για τους ερευνητές να διατηρούν μια συνολική εικόνα ολόκληρου του συνόλου δεδομένων.

Με γνώμονα την ανάγκη να βοηθηθούν οι επιστήμονες, ώστε να αποκτήσουν μια ευρύτερη εικόνα των υποκειμένων συνόλων δεδομένων, ώστε να εντοπίζονται και να εξάγονται κρυφές συσχετίσεις, δόθηκε ιδιαίτερη προσοχή στην ανάπτυξη της απεικόνισης τόσο των δεδομένων όσο και της διαδικασίας αναζήτησης. Έτσι οι ερευνητές μπορούν να αλληλεπιδρούν άμεσα με το τεράστιο ποσό των διαθέσιμων δεδομένων με διαισθητικό και ουσιαστικό τρόπο.

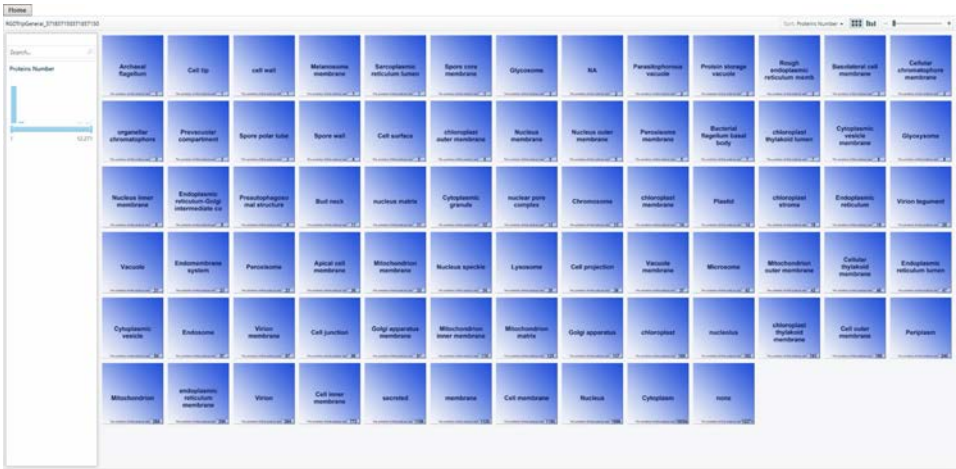
Οι βασικές πληροφορίες μιας εγγραφής (πρωτεΐνης) στην RGDtrip περιλαμβάνουν σε μορφή κάρτας όλα τα διαθέσιμα στοιχεία, οντολογία, παραπομπές βάσης δεδομένων, δεδομένα αλληλουχίας και δομής. Αυτά τα στοιχεία προέρχονται από αλληλουχίες UniProtKB [28] που περιέχουν το τριπεπτίδιο RGD. Τα γενικά χαρακτηριστικά περιέχουν τα ονόματα των γονιδίων και των πρωτεϊνών, το όνομα και το πεδίο superkingdom / domain του οργανισμού προέλευσης, τον αριθμό Enzyme Commission, την περιγραφή της πρωτεϊνικής λειτουργίας και της οικογένειας και την πρωτεϊνική υποκυτταρική θέση. Τα δεδομένα οντολογίας αποτελούνται από έναν κατάλογο λέξεων-κλειδιών και την επιλογή των όρων γονιδιακής οντολογίας [29] που ανακτώνται από το UniProtKB, ενώ επιπλέον στοιχεία αντλούνται από τις βάσεις δεδομένων Gene3D [26], PIRSF [30], Pfam [31] InterPro [32] και SUPFAM [33].

Επίσης, αποθηκεύονται πληροφορίες για την ακριβή θέση του τριπεπτίδιου RGD. Επίσης, σχετικές εγγραφές περιλαμβάνουν τις θέσεις που έχουν πειραματικά μεταβληθεί με μεταλλαξογένεση και φυσικές παραλλαγές της πρωτεϊνικής αλληλουχίας. Εάν υπάρχουν εγγραφές PDB μιας εγγραφής UniProtKB που περιέχουν το τριπεπτίδιο RGD, αποθηκεύονται και οι πληροφορίες της δομής με τον αντίστοιχο κωδικό της PDB ανάλυσης. Τα δεδομένα συλλέγονται από την τράπεζα δεδομένων πρωτεϊνών (PDB) [25, 34]

και επίσης από τη βάση δεδομένων DSSP [35, 36] σε περιπτώσεις δευτερεύουσας ταξινόμησης δομών.

Εισαγωγή και εξερεύνηση της RGDtrip

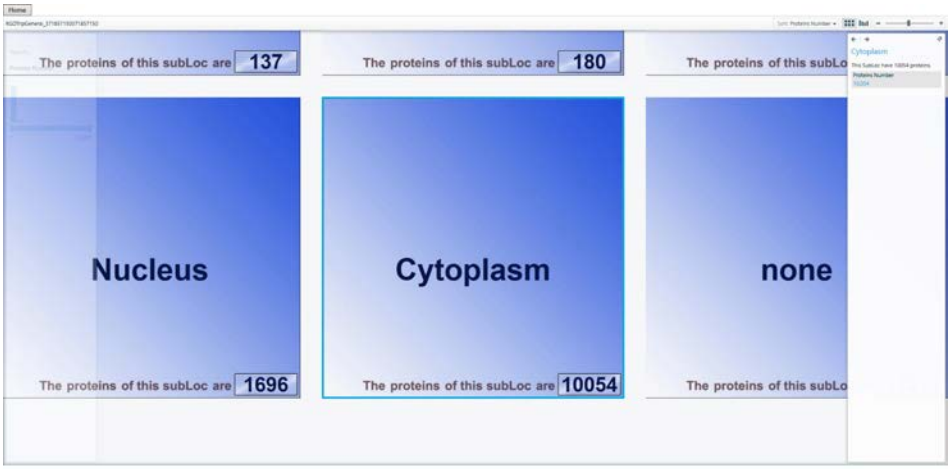
Το πρώτο επίπεδο ολόκληρης της συλλογής δεδομένων RGDtrip, όπως παράγεται από το εργαλείο απεικόνισης.



Εικόνα 32 Το πρώτο επίπεδο της συνολικής συλλογής δεδομένων RGDtrip, όπως παράγεται από το εργαλείο απεικόνισης, με 74 διαθέσιμες κάρτες/sublocs

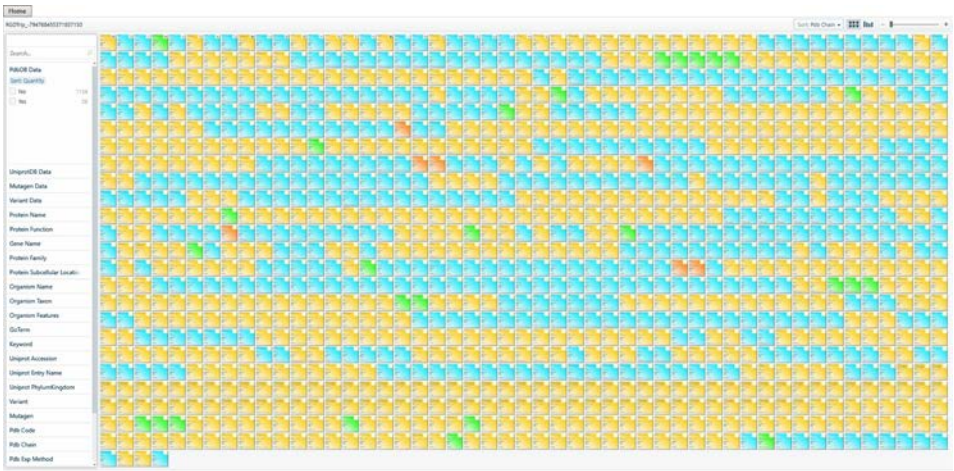
Κάθε μία από τις 74 διαθέσιμες κάρτες περιέχει πρωτεΐνες ομαδοποιημένες από την υποκυτταρική θέση τους (subLoc), προκειμένου να παρέχει μια πιο ανθρώπινη οπτική προσέγγιση και εμφανίζει το όνομα του subLoc.

Μετά τη μεγέθυνση για μια πιο προσεκτική εμφάνιση σε κάθε κάρτα, ο αριθμός των πρωτεϊνών που περιέχονται στην ομάδα subLoc και ένας πίνακας με τις ιδιότητες της κάρτας μπορεί να φαίνεται εμφανής στη δεξιά πλευρά.



Εικόνα 33 Διερεύνηση του subloc "Cytoplasm"

Με την είσοδο σε μια συλλογή δεδομένων (το διπλό κλικ στην κάρτα αρκεί) το εργαλείο απεικόνισης εμφανίζει όλες τις πρωτεΐνες της ομάδας subLoc. Κάθε κάρτα στη διασύνδεση αντιπροσωπεύει μια πρωτεΐνη και το χρώμα της κάρτας εξαρτάται από το "Organism Taxon". Οι μπλε κάρτες είναι για την Eukaryota, οι κίτρινες είναι για Βακτήρια, πράσινες για την Archea και οι πορτοκαλί κάρτες για τους ιούς.



Εικόνα 34 Κάθε κάρτα στη διασύνδεση αντιπροσωπεύει μια πρωτεΐνη και το χρώμα της κάρτας εξαρτάται από τον "Organism Taxon"

Ένας πίνακας φιλτραρίσματος δεδομένων είναι διαθέσιμος, προσφέροντας μια σειρά από 24 κριτήρια φιλτραρίσματος που μπορεί να εφαρμόζονται στην υποκείμενη συλλογή δεδομένων.

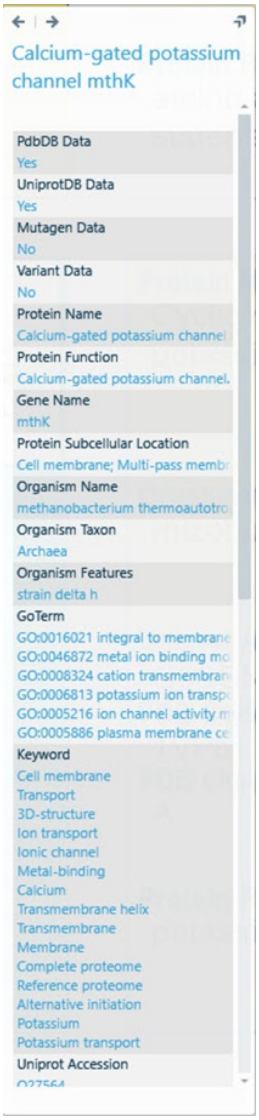
Εικόνα 35 Πίνακας φιλτραρίσματος δεδομένων με 24 κριτήρια

Η εφαρμογή PivotViewer επιτρέπει στους χρήστες να αναζητούν ομαλά και γρήγορα τα υποκείμενα σύνολα δεδομένων και να συμπεριλαμβάνουν ή να αποκλείουν συγκεκριμένα στοιχεία εφαρμόζοντας φίλτρα ενώ οι χρήστες μπορούν να αλλάξουν ταυτόχρονα τον τρόπο εμφάνισης του προκύπτοντος συνόλου καρτών επιλέγοντας μεταξύ του πλέγματος και της προβολής γραφημάτων κάνοντας κλικ στο αντίστοιχο κουμπί στην επάνω δεξιά γωνία της σελίδας.

Εικόνα 36 Αλλαγή του τρόπου εμφάνισης του συνόλου καρτών επιλέγοντας μεταξύ του πλέγματος και της προβολής γραφημάτων

Με αυτόν τον τρόπο, οι χρήστες μπορούν να ταξινομούν, να οργανώνουν και να κατηγοριοποιούν τα δεδομένα δυναμικά σύμφωνα με τα χαρακτηριστικά από το μενού ερωτημάτων δεδομένων και στη συνέχεια να μεγεθύνουν για πιο προσεκτική εμφάνιση, είτε φιλτράροντας περαιτέρω τη συλλογή για να φτάσουν σε ένα υποσύνολο ενδιαφέροντος είτε κάνοντας κλικ σε μια συγκεκριμένη κάρτα. Επιτρέποντας στους χρήστες να επικεντρωθούν σε μια συγκεκριμένη περιοχή ή να σμικρύνουν ώστε να έχουν συνολική εικόνα των δεδομένων, όπου μπορεί να αποκαλυφθούν διάφορες σχέσεις.

Η κάρτα που χρησιμοποιείται για την περιγραφή κάθε πρωτεΐνης αποτελείται από το όνομα και την οικογένεια των πρωτεϊνών, τον οργανισμό και τα δεδομένα UNIProt Accession. Το χρώμα της κάρτας εξαρτάται από τον τομέα / superkingdom του οργανισμού που παράγει την πρωτεΐνη. Όλα τα δεδομένα που είναι διαθέσιμα από τη βάση δεδομένων PDB, τον κώδικα PDB και την αλυσίδα PDB εμφανίζονται επίσης, καθώς και η μικρογραφία πρωτεΐνης στην επάνω δεξιά γωνία, αν υπάρχει.



Εικόνα 37 Όταν οι χρήστες μεγεθύνουν την κάρτα στη δεξιά πλευρά, παρέχονται πληροφορίες για την πρωτεΐνη

Ένας πίνακας πληροφοριών πλευρικής γραμμής εμφανίζεται στη δεξιά πλευρά, όταν οι χρήστες μεγεθύνουν και κάνουν κλικ στην κάρτα. Ο πίνακας παρέχει λεπτομερείς πληροφορίες σχετικά με την πρωτεΐνη.

Εικόνα 38 Εμφανίζεται η μικρογραφία (PDB) πρωτεΐνης στην επάνω δεξιά γωνία, εάν υπάρχει

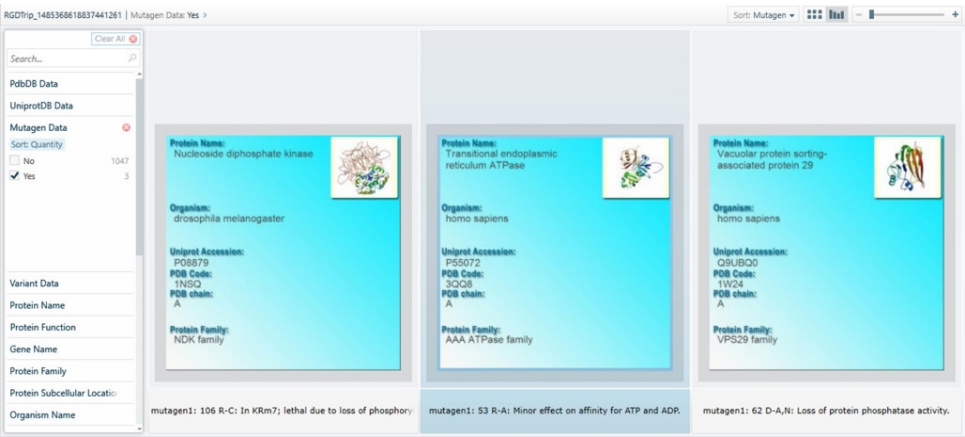
Επιπλέον, για κάθε πρωτεΐνη που συνοδεύεται από δομικά δεδομένα από τη βάση δεδομένων PDB, είναι διαθέσιμη μια δομική άποψη αυτού: μπορεί κανείς να χρησιμοποιήσει (με διπλό κλικ στη μικρογραφία) το λογισμικό Jmol, δείχνοντας χημικές δομές σε 3D με τη θέση του τριπεπτιδίου RGD που επισημαίνονται με κίτρινο χρώμα.

Εικόνα 39 Η 3D δομή με τη θέση του RGD σε κίτρινο χρώμα

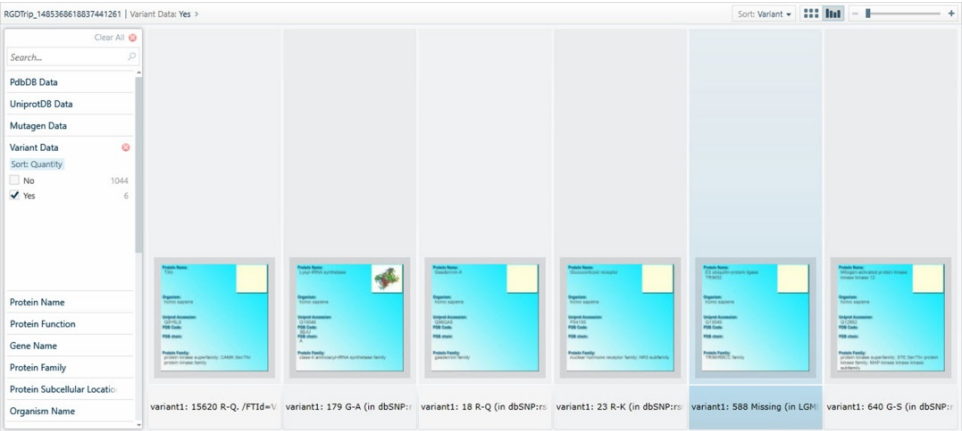
Πιο περίπλοκες αναζητήσεις θα μπορούσαν να γίνουν στο RGDtrip ενεργώντας απευθείας στους πίνακες της βάσης δεδομένων.

Επίδειξη ερωτήματος

Με την αξιοποίηση των παραπάνω λειτουργιών, οι χρήστες έχουν την ευκαιρία να πειραματιστούν με συγκεκριμένα σενάρια, τα οποία μπορούν να τα καθοδηγήσουν για να ανακαλύψουν νέες σχέσεις που δεν είχαν παρατηρηθεί προηγουμένως ή για να εκτελέσουν σύνθετα ερωτήματα. Για παράδειγμα, η έρευνα για το υποκείμενο "Κυτταρόπλασμα" περιέχει 10054 πρωτεΐνες, οι οποίες ομαδοποιούνται περαιτέρω βάσει του χαρακτηριστικού "phylumKingdom". Όταν εφαρμόζονται τα αντίστοιχα κριτήρια φιλτραρίσματος, μόνο 3 από τις 1050 πρωτεΐνες "Metazoa" συνδέονται με πειραματικές μεταλλάξεις και 6 με φυσικές παραλλαγές.



Εικόνα 40 Όταν εφαρμόζονται τα αντίστοιχα κριτήρια φιλτραρίσματος, μόνο 3 πρωτεΐνες συνδέονται με πειραματικές μεταλλάξεις



Εικόνα 41 Όταν εφαρμόζονται τα αντίστοιχα κριτήρια φιλτραρίσματος, μόνο 6 πρωτεΐνες συνδέονται με φυσικές παραλλαγές

Έτσι, οι μεταλλάξεις του τριπεπτιδίου RGD σε συγκεκριμένους οργανισμούς θα μπορούσαν εύκολα να εντοπιστούν και οι επιδράσεις τους στην πρωτεϊνική λειτουργία θα μπορούσαν να μελετηθούν.

Αναφορές – Βιβλιογραφία κεφαλαίου

1. Papadopoulos GK, Ouzounis C, Eliopoulos E. RGD sequences in several receptor proteins: novel cell adhesion function of receptors? International Journal of Biological Macromolecules. 1998;22:51–7.
2. Xiong, J-P, Stehle T, Zhang R., Joachimiak A, Frech M,Goodman S-L, Arnaout MA. Crystal structure of the extracellular segment of integrin aVb3 in complex with an Arg-Gly-Asp ligand. Science. 2002: 296:151–5.
3. Takagi J, Springer TA. Integrin activation and structural rearrangement. Immunol. 2002; 186:141–63.
4. D'Souza SE, Ginsberg MH, Plow EF. Arginyl-glycyl-aspartic acid (RGD): a cell adhesion motif. Trends in Biochemical Sciences. 1991; 16:246-50.
5. Fujii Y, et al. Crystal Structure of Trimestatin, a Disintegrin Containing a Cell Adhesion Recognition Motif RGD Original Research Article. Journal of Molecular Biology. 2003; 332:1115-22.
6. Buckley CD, et al. RGD peptides induce apoptosis by direct caspase-3 activation. Nature. 1999;397:534-9. See also commentary article in same issue by Ruoslahti E, Reed J. :479-80.
7. Routsias J, Papadopoulos GK. Polymorphic structural features of modelled HLA-DQ molecules segregate according to susceptibility or resistance to IDDM. Diabetologia. 1995;38:1251-61.
8. Paliakasis K, Routsias J, Petratos K, Ouzounis C, Kokkinidis M, Papadopoulos GK. Novel structural features of the human histocompatibility molecules HLA-DQ as revealed by modelling based on the published structure of the related molecule HLA-DR1. J Struct Biol. 1996;117:145-63.
9. Lee KH, Wucherpfennig KW, Wiley DC. Structure of a human insulin peptide - HLA-DQ8 complex and susceptibility to type 1 diabetes. Nat Immunol. 2001;2:501-7.
10. 10. Kim C-Y, Quarsten H, Bergseng E, Khosla C, Sollid LM. Structural basis for HLA-DQ2-mediated presentation of gluten epitopes in celiac disease. Proc Nat Acad Sci USA. 2004;101:4175-9.
11. Sethi DK, Schubert DA, Anders AK, Heroux A, Bonsor DA, Thomas CP, Sundberg EJ, et al. A highly tilted binding mode by a self-reactive T cell receptor results in altered engagement of peptide and MHC. J Exp Med. 2011;208:91-102.
12. Siebold C, Hansen B, Wyer JR, Harlos K, Esnouf RE, Svejgaard A, Bell JI, et al. Crystal structure of HLA-DQ0602 that protects against type 1 diabetes and confers strong susceptibility to narcolepsy. 2004;101:1999-2004.

13. Henderson KN, Tye-Din JA, Reid HH, et al. A Structural and Immunological Basis for the Role of Human Leukocyte Antigen DQ8 in Celiac Disease. Immunity. 2007;27:23-34.
14. Broughton SE, Petersen J, Theodossis A, et al. Biased T Cell Receptor Usage Directed against Human Leukocyte Antigen DQ8-Restricted Gliadin Peptides Is Associated with Celiac Disease. Immunity. 2012;37:1-11.
15. Peterson TS, Camden JM, Wang Y. P2Y2 Nucleotide Receptor-Mediated Responses in Brain Cells. Mol Neurobiol. 2010;41:356–66.
16. Humphries MJ. Insights into integrin-ligand binding and activation from the first crystal structure. Arthritis Res. 2002;4(3):S69-78.
17. Xiong J-P, Goodman SL, Arnaout MA. Purification, analysis, and crystal structure of integrins. Methods Enzymol. 2007;426:307-36.
18. Luo BH1, Carman CV, Springer TA. Structural basis of integrin regulation and signaling. Annu Rev Immunol. 2007;25:619-47.
19. Dong X, Zhao B, Iacob RE, Zhu J, Koksal AC, Lu C, Engen JR, Springer TA.Force interacts with macromolecular structure in activation of TGF-β. Nature. 2017;542:55-9. doi: 10.1038/nature21035.
20. Ohkura N, Hamaguchi M, Sakaguchi S. FOXP3+ regulatory T cells control of FOXP3 expression by pharmacological agents. Trends Pharmacol Sci. 2011;32:158-66. doi: 10.1016/j
21. Geeganage C, Wilcox R, Bath PM. Triple antiplatelet therapy for preventing vascular events: a systematic review and meta-analysis. BMC Med. 2010;8:36.
22. Meyer A, Auernheimer J, Modlinger A, Kessler H. Targeting RGD-recognizing integrins: drug development, biomaterial research, tumor imaging and targeting. Curr Pharm Des 2006, 2006;12:2723-47.
23. Li Z, Huang P, Zhang X, Lin J, Yang S, Liu B, Gao F, et al. RGD-conjugated dendrimer-modified gold nanorods for in vivo tumor targeting and photothermal therapy. Mol Pharm. 2010;7:94-104.
24. Corti A, Curnis F, Rossoni G, Marcucci F, Gregorc V. Peptide-Mediated Targeting of Cytokines to Tumor Vasculature: The NGR-hTNF Example. BioDrugs. 2013;27:591-603.
25. Förster Y, Rentsch C, Schneiders W, et al. Surface modification of implants in long bone. Biomatter. 2012;2:149-57.
26. Sun CC1, Qu XJ, Gao ZH. Arginine-Glycine-Aspartate-Binding Integrins as Therapeutic and Diagnostic Targets. Am J Ther. 2016 Jan-Feb;23(1):e198-207. doi: 10.1097/MJT
27. Marelli UK, Rechenmacher F, Sobahi TRA, Mas-Moruno C, Kessler H. Tumor targeting via integrin ligands. Front Oncol. 2013;3:1-12.

28. Kwan BH, Zhu EF, Tzeng A, Sugito HR, Eltahir AA, Ma B, Delaney MK, Murphy PA, Kauke MJ, Angelini A, Momin N, Mehta NK, Maragh AM, Hynes RO, Dranoff G, Cochran JR, Wittrup KD. Integrin-targeted cancer immunotherapy elicits protective adaptive immune responses. J Exp Med. 2017;214:1679-90.
29. Finlay BB. Cell adhesion and invasion mechanisms in microbial pathogenesis. Current Opinion in Cell Biology. 1990;2:815-20.
30. Harkiolaki M, Tsirka T, Lewitzky M, Simister PC, Joshi D, Bird LE, Jones EY, O'Reilly N, Feller SM. Distinct binding modes of two epitopes in Gab2 that interact with the SH3C domain of Grb2. Structure. 2009;17:809-22.
31. PubMed accessed on 19th May, 2017, 20.18 Eastern European Time, using the terms "RGD" and "review".
32. Viennas E, Gkantouna V, Ioannou M, Georgitsi M, Rigou M, Poulas K, Patrinos GP, Tzimas G. Population-ethnic group specific genome variation allele frequency data: A querying and visualization journey. Genomics. August 2012;100(2);93-101.
33. Eckerson WW. Three tier client/server architecture: achieving scalability, performance, and efficiency in client server applications. Open Information Systems. 1995;3:20.
34. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, et al. The Protein Data Bank. Nucleic Acids Res. 2000;28:235-42.
35. Yeats C, Lees J, Reid A, Kellam P, Martin N, Liu X, Orengo C. Gene3D: comprehensive structural and functional annotation of genomes. Nucleic Acids Res. 2008;36(Database issue):D414-8.
36. Papadopoulos P, Viennas E, Gkantouna V, Pavlidis C, Bartsakoulia M, Ioannou ZM, Ratbi I, Sefiani A, Tsaknakis J, Poulas K, Tzimas G, Patrinos GP. Developments in FINDbase worldwide database for clinically relevant genomic variation allele frequencies. Nucleic Acids Res. 2014 Jan;42(Database issue):D1020-6.
37. Consortium, The UniProt. Reorganizing the protein space at the Universal Protein Resource (UniProt). Nucleic Acids Res. 2012;40:D71-5.
38. Consortium, The Gene Ontology. Gene ontology: tool for the unification of biology. Nat. Genet. 2000;25(1):25-9.
39. Wu CH, Nikolskaya A, Huang H, Yeh L-SL, Natale DA, Vinayaka CR, Hu Z-Z, et al. PIRSF: family classification system at the Protein Information Resource. 2004;32:D112-4.
40. Punta M, Coggill PC, Eberhardt RY, Mistry J, Tate J, et al. The Pfam protein families database. Nucleic Acids Res. 2014;42(Database Issue):D222-30.
41. Hunter S, Jones P, Mitchell A, Apweiler R, Attwood TK, Bateman A, Bernard T, et al. InterPro in 2011: new developments in the family and domain prediction database. Nucl Acids Res. 2011;D1:D306-12.

42. Wilson D, Pethica R., Zhou Y, Talbot C, Vogel C, et al. SUPERFAMILY - Comparative Genomics, Datamining and Sophisticated Visualisation. Nucleic Acids Res. 2009;37(Database issue):D380-6.
43. Rose PW, Beran B, Bi C, Bluhm WF, Dimitropoulos D, et al. The RCSB Protein Data Bank: redesigned web site and web services. Nucleic Acids Res. 2011;39(Suppl 1):D392-401.
44. Joosten RP, Te Beek, TAH, Krieger E, Hekkelman ML, Hooft RWW, et al. A series of PDB related databases for everyday needs. Nucleic Acids Res. 2011;39(Suppl 1):D411-9.
45. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 1983;22:2577-637.
46. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215:403-10.

Η πρόσβαση στις ηλεκτρονικές πηγές - αναφορές επικαιροποιήθηκε τον Μάρτιο 2018.

Κεφάλαιο 5ο

Η βάση δεδομένων fungibase

Φύση των Μυκήτων

Μύκητας είναι ένας ανώτερος ευκαρυωτικός οργανισμός που πολλαπλασιάζεται με σεξουαλική (αγενή) ή ασεξουαλική (εγγενή) αναπαραγωγή. Η ταξινόμηση των μυκήτων γίνεται με βάση την θαλλική τους μορφή (δηλαδή τη σωματική τους μορφή) και τη μορφολογία των αναπαραγωγικών τους οργάνων (βλαστική μορφή) στην οποία προτεραιότητα έχει η μορφολογία οργάνων εγγενούς (σεξουαλικής, φυλετικής) αναπαραγωγής έναντι αυτών αγενούς (ασεξουαλικής, αφυλετικής) αναπαραγωγής. Ο θαλλός μυκήτων εμφανίζεται στις κάτωθι μορφές:

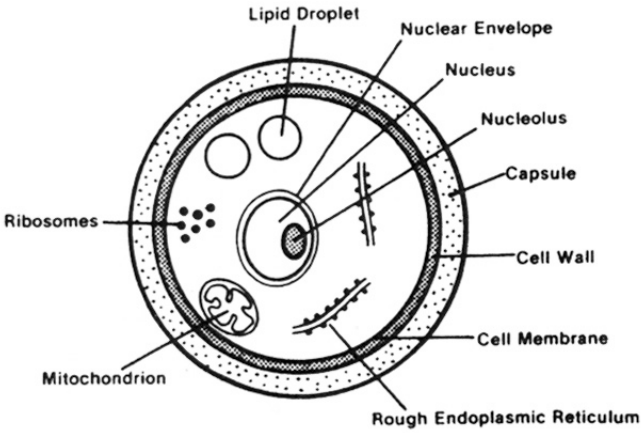
1) μονοκύτταρη (πχ Ζυμοκύτταρο ζυμοειδών μυκήτων)

2) πολυκύτταρη α) κοινοκυτταρική, (πχ Πλασμώδιο μυξομυκήτων, χωρίς κυτταρικό τοίχωμα, κοινοκυτταρικό Μυκήλιο Ζυγομυκήτων) β) γνήσια πολυκύτταρη (διαφραγματοφόρο Μυκήλιο μυκηλιακών μυκήτων, Αμανίτης στα μανιτάρια)

3) ψευδοπολυκύτταρη (πχ Ψευδομυκήλιο, αποικιακή μορφή ζυμοειδών μυκήτων).

Οι μύκητες είναι εξελικτικά πλησιέστερα στα ζώα παρά στα φυτά. Σε αυτό συνηγορεί η ύπαρξη χιτίνης στο κυτταρικό τοίχωμα των περισσοτέρων (η οποία ανευρίσκεται και στον εξωσκελετό των εντόμων και των αρθροπόδων) αντί της κυτταρίνης, η χρήση του γλυκογόνου ως αποθησαυριστικού

πολυσακχαρίτη αντί του αμύλου και η απουσία χλωροπλαστών. Από την άλλη, δεν διαθέτουν νευρομυικό σύστημα και αυτοτελή κίνηση του θαλλού, σε αντίθεση με τα ζώα, διαθέτουν κυτταρικό τοίχωμα και γενικώς αναπτύσσονται με διχοτόμηση όπως τα φυτικά κύτταρα και όχι με σύσφιξη των κυττάρων τους όπως συμβαίνει στα ζώα και η βασική στερόλη στην κυτταρική τους μεμβράνη είναι η εργοστερόλη κι όχι η χοληστερόλη, που χρησιμοποιούν τα ζωικά κύτταρα. Τέλος, οι μύκητες έχουν μόνο δύο επίπεδα κυτταρικής οργάνωσης (κύτταρο-όργανο), σε αντίθεση με τα ζώα και τα φυτά που διαθέτουν και το ενδιάμεσο επίπεδο των ιστών, και ενίοτε και το ανώτερο των οργανικών συστημάτων.



Εικόνα 42 Το μυκητιακό κύτταρο είναι ευκαρυωτικό

Οι οργανισμοί που ανήκουν στους αληθείς μύκητες είναι γνωστοί και ως ευμύκητες. Δεν φωτοσυνθέτουν και είναι οσμότροφοι, σαπροβιοτικοί (σαπροφυτικοί), συμβιωτικοί (ενίοτε δε παρασιτικοί) οργανισμοί. Υπάρχουν μύκητες που έχουν φωτοσυνθετικές ικανότητες για παραγωγή, υπό συγκεκριμένες συνθήκες, μέρους των απαιτούμενων θρεπτικών συστατικών, αλλά και αυτοί πλέον κατατάσσονται σε άλλο βασίλειο. Η ποικιλία βιοτόπων και ξενιστών που μπορούν να εκμεταλλευτούν τους καθιστά επικίνδυνα - ευκαιριακά συνήθως- παθογόνα ανθρώπων, φυτών και ζώων. Έχουν δε αναπτύξει εξαιρετικούς μηχανισμούς διάδοσης σε μεγάλες αποστάσεις και

επιβίωσης σε δυσμενή περιβάλλοντα με αποτέλεσμα να βρίσκονται σχεδόν παντού. Εκτός από βλαστική και θαλλική μορφή, πολλοί μύκητες έχουν και ανθεκτική, μεταβολικώς αδρανή μορφή που λέγεται εφησυχάζουσα, αντίστοιχη με τα σπόρια κάποιων βακτηρίων και τις κύστες κάποιων πρωτοζώων.

Το μυκητιακό κύτταρο είναι ευκαρυωτικό με υψηλή διαμερισματοποίηση, τα τυπικά ευκαρυωτικά οργανίδια και πληθώρα μεμβρανών. Έχει μεγάλη ωσμορρυθμιστική ικανότητα χάρη στο κυτταρικό του τοίχωμα και μπορεί να διαμορφώνει ευνοϊκά το μικροπεριβάλλον του. Το κυτταρικό του τοίχωμα αποτελείται κυρίως από χιτίνη (αντί της κυτταρίνης των φυτικών κυττάρων), β-1,6 γλυκάνη και σε χρώση Gram εμφανίζεται Gram (+). Η κυτταρική του μεμβράνη φέρει ως βασική στερόλη την εργοστερόλη (αντί της χοληστερόλης των ζωικών κυττάρων). Ελάχιστοι αληθείς μύκητες είναι μονοκυττάριοι, ενώ οι περισσότεροι διαθέτουν πολυκυττάριες ή κοινοκυττάριες υφές με απλοειδείς πυρήνες, οι οποίες αθροιζόμενες σχηματίζουν το μυκήλιο. Οι μύκητες γενικά είναι χημειοετερότροφοι, εδραίοι οργανισμοί, αλλά με μικρού βαθμού κυτταρική διαφοροποίηση στην περίπτωση πολυκύτταρων θαλλών. Τόσο στον αγενή όσο και στον εγγενή πολλαπλασιασμό, οι μύκητες χρησιμοποιούν την σπορογονία για ταχεία εξάπλωση και αύξηση του πληθυσμού. Τα σπόρια έχουν ανάλογα με το μύκητα διάφορες προβλέψεις. Η αδιαβροχότητα (υδατοστεγανότητα), το ανθεκτικό τοίχωμα, η εν γένει ανθεκτικότητα και οι προβλέψεις πτήσης ανεμοπορίας ή διασποράς είναι απαραίτητα χαρακτηριστικά για πλάνητες βλαστικές μορφές. Δεν αποτελούν τις ανθεκτικές μορφές του μύκητα, αν και είναι σαφώς ανθεκτικότερα από τα θαλλοκύτταρα. Οι ανθεκτικές (εφησυχάζουσες) μορφές του μύκητα, αντίστοιχες με τις κύστεις των πρωτοζώων και τα σπόρια των βακτηρίων, είναι τα σκληρώτια και τα χλαμυδοσπόρια.

Η αγενής αναπαραγωγή, με τη μαζική και ταχύτατη παραγωγή σπορίων γενετικά ταυτόσημων με τον γονικό μύκητα, επιτρέπει την ταχεία εξάπλωση του οργανισμού, ιδίως σε ευνοϊκές διατροφικές συνθήκες (μαζική μόλυνση φυτειών ή σαπροφυτία επί συγκεντρωμένων νεκρών οργανισμών λόγω φυσικού συμβάντος). Για την επίτευξη εξέλιξης όμως χρησιμοποιείται η αμφιγονική αναπαραγωγή, προκειμένου να υπάρχει δυνατότητα γενετικού

ανασυνδυασμού. Καθώς οι μύκητες στερούνται δυνατότητας αυτόνομης μετακίνησης η αμφιγονική αναπαραγωγή είναι δυσχερής, επαφίεται εν πολλοίς στην τύχη και είναι εξαιρετικά γόνιμη: γίνεται δε είτε με επαφή και συγχώνευση υφών (αναστόμωση), με αποτέλεσμα νέο γενετικό περιεχόμενο σε ενυπάρχον κύτταρο που ακολούθως αναπαράγεται, είτε με εξειδικευμένα όργανα εγγενούς αναπαραγωγής που θα παράγουν μαζικά σπόρια τα οποία θα φέρουν το ανασυνδυασμένο γενετικό περιεχόμενο [1].

Σημασία των Μυκήτων στις φαρμακευτικές επιστήμες

Το βασίλειο των μυκήτων περιλαμβάνει μικροσκοπικούς και μακροσκοπικούς ευκαρυωτικούς οργανισμούς με φυσιολογία ανάμεσα σε αυτές φυτών και ζώων. Κάποια από τα μέλη του μπορούν να αποβούν εξαιρετικά παθογόνα με πολύ μεγάλα ποσοστά κατάληξης αν δεν υποβληθούν οι ασθενείς σε θεραπεία. Οι δυσκολίες που προκύπτουν στην ιατρική αντιμετώπιση των αντίστοιχων νοσημάτων («μυκητιάσεων») οφείλονται

(α) στα μη ειδικά συμπτώματα, που παραπέμπουν σε πληθώρα μολυσματικών νοσημάτων βακτηριακής, μυκητιακής και πολλές φορές και ιογενούς παθολογίας, δυσχεραίνοντας τη διάγνωση

(β) στην ευκαρυωτική φύση των μυκήτων, που σημαίνει ότι διαθέτουν εξελιγμένη βιοχημική μηχανή η οποία (ι) αντιδρά γρήγορα σε πολλά φάρμακα και καθίσταται επίσης γρήγορα ανθεκτική σε όσα στην αρχή δεν αντιδρούσε ενώ (ιι) τα τοξικά για αυτούς σκευάσματα συνήθως είναι τοξικά και για τον άνθρωπο-ξενιστή με αποτέλεσμα μικρό σχετικά αριθμό αντιμυκητιακών φαρμάκων

(γ) στο εξαιρετικά πολύπλοκο σύστημα ταξινόμησης των μυκήτων, που είναι διττό λόγω της ύπαρξης τόσο μονογονικής όσο και αμφιγονικής αναπαραγωγής, που όμως δεν εκτείνεται σε όλα τα στελέχη με αποτέλεσμα απρόβλεπτης ταξινομικής κλίμακας συγκλίσεις και αποκλίσεις, ιδίως ως προς τα μορφολογικά και βιοχημικά χαρακτηριστικά.

Για το λόγο αυτό η έλευση της μοριακής βιολογίας στην κλινική μυκητολογία, στη δεκαετία του 1990, υπήρξε καταλυτική: αφενός οι μοριακές τεχνικές επέτρεπαν σχετικά γρήγορη και σαφή διάκριση των μυκητιάσεων από άλλης αιτιολογίας νοσήματα, ενώ ταυτόχρονα παρείχαν και ταυτοποίηση του ενεχόμενου μύκητα κατά αντικειμενικότερο τρόπο σε σχέση με τις συμβατικές μεθόδους, καθώς ο έλεγχος του γενετικού υλικού δεν επηρεαζόταν από την πιθανή δυσγονική φύση και την αντίστοιχη μορφολογία, όπως η συμβατική ταυτοποίηση. Παράλληλα, γινόταν ευχερής η λεπτομερέστερη και ακριβής τυποποίηση, για τεκμηρίωση επιδημιών και περιστατικών υποτροπής.

Εκτός όμως από τη διαγνωστική μυκητολογία, οι μοριακές τεχνικές χρησιμοποιήθηκαν για την βιοτεχνολογική ταυτοποίηση και τυποποίηση μυκήτων ώστε να ανακαλυφθούν μεταβολίτες ενδιαφέροντος, με κορυφαίους μεταξύ αυτών δυνητικούς αντιμικροβιακούς παράγοντες (αντιβιοτικά).

Τέλος, σε μια σύζευξη των δύο προηγουμένων, έγινε εφικτή η μοριακή διερεύνηση κλινικών στελεχών για ύπαρξη ή ανυπαρξία συγκεκριμένων γονιδίων αντοχής στα αντιμυκητιακά σκευάσματα, ώστε να εξοικονομείται χρόνος με καλύτερα στοχευμένες αντιμυκητιασικές θεραπείες αντί προσεγγίσεων βρόγχου ανατροφοδότησης (trial and error).

Τα ανωτέρω τεκμηριώνουν τη σημασία της μοριακής (αλλά και της συμβατικής) μυκητολογίας στη Φαρμακευτική έρευνα. Η έκρηξη αυτής, σε ιατρικό, φαρμακευτικό και βιοτεχνολογικό επίπεδο δημιούργησε ένα χάος ως προς την πρόσβαση στα αποτελέσματα της αντίστοιχης έρευνας που υπήρξε ογκώδης, άναρχη, πολυδιασπασμένη και ευκαιριακή [1].

Μελέτη αντιμυκητιακών φαρμάκων

Η πλειονότητα των αντιμυκητιακών φαρμάκων δρουν, με άμεσο ή έμμεσο τρόπο, επί του κυτταρικού τοιχώματος ή επί της κυτταρικής μεμβράνης του μύκητα. Το αποτέλεσμα είναι η απώλεια ομοιοστασίας του μύκητα που οδηγεί στην αναστολή ανάπτυξης αυτού ή, αν είναι σοβαρότερη, στον νεκρωσικό κυτταρικό θάνατο αυτού. Η αναστολή ανάπτυξης είναι επαρκής

ιατρικός στόχος. Τα μικρόβια όταν παύσουν να πολλαπλασιάζονται («αναστολή ανάπτυξης») γενικώς παύουν να είναι επικίνδυνα για τον οργανισμό –στόχο («ξενιστή») αν και ενίοτε παραμένουν επικίνδυνα λόγω παραγωγής τοξινών (όπου αυτή υπάρχει και υλοποιείται).

Τα διάφορα γένη, είδη αλλά και στελέχη μυκήτων έχουν διαφορετική ευαισθησία στην ποικιλία αντιμυκητιακών σκευασμάτων που υπάρχουν. Η επιλογή του κατάλληλου, ιδίως στις περιπτώσεις θεραπευτικής ή προφυλακτικής χρήσης σε ανθρώπους και λοιπούς ζωντανούς οργανισμούς, είναι πολύ σημαντική για την επιτυχή έκβαση της θεραπείας, με τη μικρότερη δυνατή επιβάρυνση του λήπτη του φαρμάκου, του περιβάλλοντος και με τη μικρότερη χρονική διάρκεια και κόστος της θεραπείας. Εκτός από τις πολλές και ποικίλες παραμέτρους ενός φαρμακευτικού σκευάσματος, όπως τοξικότητα σε συγκεκριμένους κυτταρικούς τύπους/ ιστούς/ όργανα του λήπτη («κυτταροτοξικότητα»), ταχύτητα διάδοσης του σκευάσματος στα διάφορα όργανα και ιστούς του λήπτη και από εκεί στους κυτταρικούς σχηματισμούς και οργανίδια του μικροοργανισμού-στόχου, παραμονή του στο βιοχημικό σύστημα του λήπτη και του στόχου, μεταβολισμός-εξουδετέρωση κλπ («φαρμακοκινητική») η πλέον βασική παράμετρος ενός φαρμάκου είναι η τρωτότητα του μικροοργανισμού σε αυτό. Αυτή ορίζεται ως «Ευαισθησία» και προσδιορίζεται εργαστηριακά είτε επί πειραματοζώων (“in vivo”) είτε επί καλλιεργειών των μικροοργανισμών (“in vitro”). Σε μερικές περιπτώσεις, η in vitro δοκιμασία γίνεται σε κυτταροκαλλιέργειες που μολύνονται με το μικροοργανισμό, είτε επειδή αυτός δεν μπορεί να καλλιεργηθεί σε τεχνητά υποστρώματα (πχ Ιοί), είτε για αυξημένη πιστότητα των συνθηκών δράσης του φαρμάκου. Πρόκειται για την πλέον πολύπλοκη μέθοδο, που εδώ δεν θα εξεταστεί περαιτέρω.

Ο «εν δοκιμίω» (in vitro) έλεγχος φαρμάκων είναι μια αποκλειστική δοκιμασία που επιτρέπει τον αποκλεισμό ακατάλληλων σκευασμάτων ως θεραπευτικές επιλογές. Η καταλληλότητα ή ακαταλληλότητα ελέγχονται ανά απομονωθέν στέλεχος μικροοργανισμού. Γενικά συμπεράσματα για την ευαισθησία μεγαλύτερων ταξινομικών βαθμίδων (είδους, γένους, οικογένειας κλπ) σε κάποιο σκεύασμα μπορούν να προκύψουν (και χρησιμοποιούνται σε

αρκετά πεδία γνώσης), αλλά δεν χορηγείται θεραπεία με βάση αυτά. Τα αντιμυκητογράμματα είναι ανά στέλεχος (πρότυπο ή κλινικής απομόνωσης).

Καθώς είναι αδύνατο –και πιθανώς άχρηστο- να ελεγχθεί ένα σκεύασμα στις δυσμενέστερες δυνατές συνθήκες, ώστε να επιλεγεί το σίγουρα αποτελεσματικό, προτιμάται η αντίθετη μέθοδος: τα σκευάσματα ελέγχονται σε σχεδόν ιδανικές συνθήκες δράσης, και αν αποδειχτεί ότι υπό αυτές τις συνθήκες ο μικροοργανισμός έχει αντοχή (δηλαδή η αποτελεσματικότητα του σκευάσματος είναι μικρή ως μηδενική) συμπεραίνεται ότι αποκλείεται στις δυσμενείς συνθήκες που ενέχει η θεραπεία ασθενούς να προβεί αποτελεσματικό. Τονίζεται ότι μια ποικιλία παραγόντων καθιστούν το σώμα του ασθενούς πολύ δυσχερέστερο πεδίο δράσης για τα φαρμακευτικά σκευάσματα, αφού αλληλεπιδρούν πολλοί παράγοντες, ιδιαίτερα δε φαινόμενα κυτταροτοξικότητας και φαρμακοκινητικής που περιπλέκουν τη δοσολογία. Επίσης τονίζεται ότι είναι σχεδόν αδύνατον να εξομοιωθεί το περιβάλλον ενός ασθενούς για ρεαλιστικό έλεγχο του φαρμάκου. Αυτό όχι μόνο είναι βιοχημικώς αφάνταστα πολύπλοκο, αλλά επιπλέον διαφέρει ανά ασθενή. Το τελευταίο πρόβλημα ελπίζεται ότι στο εγγύς μέλλον θα αντιμετωπιστεί με προσεγγίσεις του γνωστικού κλάδου της «φαρμακογονιδιωματικής»[1].

Μυκητολογική βάση fungibase

Η βάση δεδομένων FUNGIBASE υπήρξε απότοκος αυτής της πραγματικότητας, όπως και της φύσης των διαφόρων ψηφιακών καταθετηρίων, που επικεντρώνονταν σε χαρακτηριστικά του μικροοργανισμού ενδιαφέροντος (αλληλουχίες, μεταβολικές οδούς, φαινοτύπος) και όχι στον υποκείμενο μεθοδολογικό ιστό.

Αντικείμενό της είναι η καταγραφή των μεθοδολογικών παραμέτρων στη μυκητολογική έρευνα, τόσο των επιτυχημένων (με τη μορφή δημοσιεύσεων) όσο και των αποτυχημένων προσπαθειών. Η καταγραφή επιτρέπει την ανέξοδη και άμεση πρόσβαση κάθε ενδιαφερόμενου ερευνητή στο μεθοδολογικό κομμάτι πληθώρας δημοσιεύσεων χωρίς την ανάγκη

σταχυολόγησης αυτών, εξοικονομώντας πιθανότατα πόρους και σίγουρα χρόνο. Γνωρίζοντας τα μοριακά εργαλεία και το αποτέλεσμά τους θα είναι εφικτή η χρήση (α) μοριακών πρωτοκόλλων που έχουν ήδη χρησιμοποιηθεί επιτυχώς για άλλη εφαρμογή ώστε να υποστηριχθεί με ελαχιστοποίηση του χρόνου και του κόστους διαφορική χρήση τους ή επέκταση της χρήσης τους σε άλλες εφαρμογές προκειμένου να επιταχύνονται έργα καινοτομίας (β) νέων πρωτότυπων μοριακών πρωτοκόλλων με εστιασμένο τρόπο χωρίς πιθανότητα ακούσιας επανάληψης για εφαρμογές έρευνας και καινοτομίας (γ) προηγούμενων μεθοδολογικών δεδομένων για μεταναλύσεις. Προκειμένου να αποφεύγονται αδιέξοδες προσεγγίσεις του παρελθόντος ή να είναι εφικτή η εξαρχής γνώση των παραμέτρων τους ώστε να επανελεγχθούν ή τροποποιηθούν, η βάση επιτρέπει την καταγραφή μη δημοσιευμένων-αποτυχημένων ή αδιέξοδων αποτελεσμάτων σε ξεχωριστή οντότητα, με βάση το ερευνητικό ίδρυμα. Η δομή αυτής της οντότητας είναι παρόμοια με τη δομή καταγραφών των επιτυχημένων μεθόδων και πρωτοκόλλων, που γίνεται με βάση την δημοσίευση.

Η δομή της βάσης στηρίζεται σε είδη/ταξινομικές βαθμίδες μυκήτων, με τον χρήστη να μπορεί να επιλέγει υπάρχουσες καταγραφές ειδών για να μελετήσει ή για να προσθέσει τη δική του. Αν ο μύκητας ενδιαφέροντος δεν υπάρχει, ο χρήστης μπορεί να τον προσθέσει. Κατόπιν εισέρχεται στα πεδία καταγραφής και επιλέγει την τεχνική και παραθέτει βιβλιογραφικά ή ιδρυματικά δεδομένα για επιτυχείς και αποτυχημένες ή απλά αδημοσίευτες προσπάθειες αντίστοιχα. Η βάση αναπτύχθηκε για μοριακές τεχνικές, αλλά με την πρόβλεψη και για επέκταση σε φαινοτυπικές, από μικροσκοπία και καλλιεργιωματική μέχρι φαρμακοδυναμικούς ελέγχους/αντιμυκητογράμματα. Στην επιλεγείσα τεχνική υπάρχουν βασικά πεδία προς συμπλήρωση, τα οποία αποτελούν και όρους έρευνας, ενώ είναι εφικτή η εισδοχή αρχείων αποτελεσμάτων, όπως χρωμογράμματα, αλληλουχίες, φωτογραφίες πηκτωμάτων, τρυβλίων ή μικροσκοπίας.

Σε ότι αφορά εφαρμογές έρευνας η fungibase δεν έχει σχεδιαστεί για εντυπωσιακές λειτουργίες και επεξεργασία δεδομένων ή εξόρυξη δεδομένων. Μπορεί να ερευνηθεί με βάση τη μεθοδολογία, τις μεθόδους και τα επιμέρους

πεδία αυτών ή/και με βάση τα είδη μυκήτων, προκειμένου να ελεγχθεί αν μια τεχνική έχει εφαρμοστεί σε συγκεκριμένο μύκητα και τι αποτελέσματα είχε, ή το σε ποια είδη έχει εφαρμοστεί μια μέθοδος ή παράμετρος μεθόδου και τα αποτελέσματα, μαζί με τις μεθοδολογικές αποκλίσεις (πχ εκκινητές PCR σε διαφορετικά είδη μυκήτων με διαφορετικά προγράμματα θερμικών κυκλοποιητών).

Ωστόσο, έχει δοθεί ιδιαίτερη προσοχή στο σχεδιασμό της διεπαφής, ώστε να είναι εύχρηστη και να δίνει αρκετές δυνατότητες αναζήτησης στον ερευνητή.

Ο σχεδιασμός της εφαρμογής

Η συνολική αρχιτεκτονική του συστήματος βασίζεται σε ένα μοντέλο πελάτη-διακομιστή τριών επιπέδων [2], που περιλαμβάνει τρία βασικά στοιχεία: την εφαρμογή πελάτη, τον διακομιστή εφαρμογών και τον διακομιστή βάσης δεδομένων.

Στην fungibase, για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε «Ελεύθερο Λογισμικό / Λογισμικό Ανοικτού Κώδικα» (ΕΛ/ΛΑΚ). Ο όρος ΕΛ/ΛΑΚ ομαδοποιεί το Ελεύθερο Λογισμικό (ΕΛ) και το Λογισμικό Ανοικτού Κώδικα (ΛΑΚ). Ως σύνολο περιγράφει λογισμικό το οποίο διατίθεται με ειδικές άδειες, οι οποίες επιτρέπουν στους χρήστες να μελετήσουν, να τροποποιήσουν και να βελτιώσουν το λογισμικό. Η κύρια διαφορά μεταξύ τους είναι ότι ο όρος Ελεύθερο Λογισμικό εστιάζει στις ελευθερίες που παρέχονται στο χρήστη, μέσω της αδειοδότησης, ενώ το Λογισμικό Ανοικτού Κώδικα δίνει έμφαση στο τεχνικό σημείο της διαθεσιμότητας του πηγαίου κώδικα και της δυνατότητας τροποποίησης και συνεργατικής ανάπτυξης [3].

Συγκεκριμένα, για την αποθήκευση δεδομένων στο «Επίπεδο Δεδομένων» χρησιμοποιείται το σχεσιακό μοντέλο δεδομένων και συγκεκριμένα το σύστημα διαχείρισης βάσεων δεδομένων MySQL Server [4].

Το «Επίπεδο εφαρμογής» υποστηρίζεται από τον Apache Web Server. Είναι ο δημοφιλέστερος διακομιστής HTTP ανοιχτού κώδικα για σύγχρονα λειτουργικά συστήματα, συμπεριλαμβανομένων των UNIX και των Windows. Παρέχει έναν ασφαλή, αποδοτικό και επεκτάσιμο διακομιστή που παρέχει υπηρεσίες HTTP σε συγχρονισμό με τα τρέχοντα πρότυπα HTTP [5].

Τέλος, η υλοποίηση της κύριας διεπαφής της εφαρμογής, στο «Επίπεδο Πελάτη», στηρίχθηκε στη γλώσσα προγραμματισμού PHP. Η PHP (Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script. Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστότοπων. Η ευρύτητα στη χρήση της είναι απόρροια της ευκολίας που παρουσιάζει ο προγραμματισμός με αυτή αλλά και στο γεγονός πως είναι μια γλώσσα η οποία βρίσκεται σχεδόν σε κάθε διακομιστή [6, 7].

Σχεδίαση άμεσης ανταπόκρισης (responsive design)

Η τεχνολογία με το πέρασμα των χρόνων αναπτύσσεται όλο και περισσότερο, καθώς χρόνο με τον χρόνο, δημιουργούνται καινούργιες συσκευές τόσο σε επίπεδο επιτραπέζιων υπολογιστών όσο και σε επίπεδο κινητών συσκευών. Οι υπολογιστικές δυνατότητες που κατέχουν σήμερα τα κινητά τηλέφωνα αλλά και άλλες μοντέρνες συσκευές είναι τεράστιες σε σχέση με τα παλιά τα χρόνια και σε συνδυασμό με την φορητότητα και τις πολλές

χρήσιμες εφαρμογές που προσφέρουν, έχουν οδηγήσει στην αύξηση της χρήσης τους δραστικά.

Ανάγκες για responsive design

Λόγω των πιο πάνω συνθηκών, έχει παρατηρηθεί τα τελευταία χρόνια μια μεγάλη αύξηση στην χρήση του διαδικτύου από κινητές συσκευές. Έτσι μέσα από αυτό, προέρχεται και η ανάγκη για εφαρμογή της τεχνικής που ονομάζεται responsive design για τον σχεδιασμό των ιστοσελίδων. Σύμφωνα με αυτή την τακτική μια ιστοσελίδα σχεδιάζεται με τέτοιο τρόπο έτσι ώστε το περιεχόμενο της, να διαμορφώνεται και να προσαρμόζεται ανάλογα με την συσκευή και το μέγεθος της οθόνης που καλείται να εμφανιστεί η σελίδα.

Πλεονεκτήματα responsive design

Τα πλεονεκτήματα αυτής της τεχνικής σχεδιασμού είναι πολλά. Ο κύριος στόχος της είναι μια ιστοσελίδα να είναι ευανάγνωστη και ευπαρουσίαστη προς τον χρήστη από όποια συσκευή και να επιλέξει να την επισκεφθεί, χωρίς να χρειάζεται να κάνει πλάγια μετακίνηση η κάποια μεγέθυνση σε αυτή.

Σημαντικό ρόλο παίζει και στον τομέα των επιχειρήσεων, καθώς επέρχεται περισσότερο κέρδος στις επιχειρήσεις όταν ο χρήστης μπορεί με ευκολία να έχει πρόσβαση στην ιστοσελίδα της επιχείρησης από οποιαδήποτε συσκευή. Και αυτό γιατί σε αντίθετη περίπτωση οι χρήστες θα ήταν δυσάρεστη μένοι με το περιεχόμενο, κάτι που θα πολύ πιθανό να σήμαινε απώλεια επισκεπτών για την σελίδα.

Μια άλλη πιθανή λύση στο πρόβλημα που δημιουργείται λόγω των σημερινών αναγκών θα ήταν να δημιουργηθεί μια ξεχωριστή εφαρμογή για κάθε διαθέσιμη συσκευή και πλατφόρμα που υπάρχει. Αυτή η λύση όμως θα ήταν αρκετά χρονοβόρα, με περισσότερο κόστος και καθόλου αποτελεσματική ενώ επίσης είναι αδύνατο να προβλεφθεί η κυκλοφορία καινούργιων συσκευών και οι καινούργιες ανάγκες που πιθανόν να προκύψουν.

Έτσι, η δημιουργία μιας ανάλογα προσαρμόσιμης εφαρμογής, αποτελεί μια πιο ικανοποιητική λύση, κάτι που θα διευκολύνει επίσης οποιαδήποτε μελλοντική συντήρηση χρειαστεί να γίνει από τον δημιουργό.

Περιγραφή της εφαρμογής

Η πλατφόρμα στην οποία βασίζεται η εφαρμογή δεν αποτελεί ένα ολοκληρωμένο σύστημα σχεδιασμού, κατασκευής και διαχείρισης ιστοσελίδων, αλλά μία δομημένη, αρθρωτή συλλογή από εργαλεία γλώσσας σήμανσης, προγραμματισμού διακομιστή και βάσης δεδομένων, τα οποία είναι επιλεγμένα για την αλληλεπίδραση και συνεργασία τους στην ανάπτυξη εφαρμογών ιστού και ιστοσελίδων.

Το περιβάλλον εργασίας που δημιουργεί η πλατφόρμα βρίσκεται σε μια φάση λειτουργικής και σχεδιαστικής ωριμότητας. Βασικός προσανατολισμός παραμένει η ενίσχυση και η υποστήριξη της δημιουργικής δραστηριότητας μέσα από ένα εύχρηστο περιβάλλον τεχνολογίας αιχμής. Ο στόχος της, ο σχεδιασμός, η δημιουργία και η υποστήριξη ολοκληρωμένων εφαρμογών και ιστοσελίδων προσφέροντας στον σχεδιαστή ένα δυναμικό περιβάλλον οργάνωσης και ένα ανοικτό, ασφαλές και αξιόπιστο σύστημα, επιτυγχάνεται με αποτελεσματικότητα λόγω της αξιοποίησης της συσσωρευμένης εμπειρίας, οικονομίας κλίμακας και εποικοδομητικής χρήσης της υπάρχουσας δικτυακής υποδομής.

Παράλληλα, οι σχεδιαστικοί άξονες που επιτυγχάνονται από την αρθρωτή δομή της συλλογής αποδίδουν προσαρμοστικότητα σε λειτουργικές απαιτήσεις, ευελιξία, ευκολία στη χρήση και δυνατότητες αναβάθμισης και επέκτασης. Η χρήση ανοικτών προτύπων που επιτρέπει την ελεύθερη διάθεση χωρίς την απαίτηση αδειών χρήσης είναι απλά ακόμη ένα πλεονέκτημα.

Τεχνολογίες που χρησιμοποιήθηκαν

Η συλλογή που χρησιμοποιείται για την ανάπτυξη της εφαρμογής περιλαμβάνει HTML 5, σύμφωνα με τις συστάσεις του World Wide Web Consortium (W3C), της κύριας οργάνωσης διεθνών προτύπων για τον παγκόσμιο ιστό, με την ενίσχυση οπτικής παρουσίασης που προσφέρει η χρήση Cascading Style Sheets (CSS) και την ευχρηστία που προσφέρει η χρήση Javascript. Για τις λειτουργίες που απαιτούν βάση δεδομένων γίνεται χρήση της γλώσσας προγραμματισμού PHP και διαχείριση βάσης δεδομένων MySQL [7, 8, 9].

Τα εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής αναλύονται στη συνέχεια.

Εικόνα 43 Τεχνολογίες ανάπτυξης της fungibase

HTML5

Χρησιμοποιήθηκε για την ανάπτυξη των βασικών ιστοδελίδων της εφαρμογής [8].

Η HTML είναι η βασική γλώσσα σήμανσης που χρησιμοποιείται για τη δημιουργία και την οπτική αναπαράσταση μιας ιστοσελίδας, ενώ τα αρχικά της σημαίνουν "HyperText Markup Language".

Η ιστορία της HTML ξεκινάει το 1980, ενώ το 1993 εκδόθηκε το πρώτο πρόχειρο κείμενο προδιαγραφών της HTML. Το 1995 παρουσιάστηκε η πρώτη

επίσημη έκδοση της HTML με την ονομασία "HTML 2.0". Από το 1996 και μετά, οι προδιαγραφές της HTML ορίζονται από το World Wide Web Consortium (W3C, http://www.w3.org. Το 1997 εκδόθηκε η "HTML 4.0", η οποία εισήγαγε πολλά νέα χαρακτηριστικά, όπως περισσότερες επιλογές πολυμέσων, φύλλα στυλ, καλύτερες δυνατότητες εκτύπωσης, καθώς και προσιτότητα για τους χρήστες με αναπηρίες. Το 1999 υπήρξε μία αναθεώρηση των προδιαγραφών με την "HTML 4.01", ενώ σε αυτήν δηλώθηκαν τρεις ορισμοί προτύπου με μικρές διαφοροποιήσεις, τα: HTML 4.01 Strict. HTML 4.01 Transitional και HTML 4.01 Frameset.

Το 2008 εκδόθηκαν οι πρώτες (πρόχειρες) προδιαγραφές της "HTML5", ενώ στα τέλη του 2014 οριστικοποιήθηκε, ορίζοντας πλέον το πρότυπο ως αυτόνομο και όχι σαν περιγραφή βασισμένη στην SGML. Η HTML5, προσθέτει πολλά νέα χαρακτηριστικά, ειδικότερα στον τομέα των πολυμέσων με τις ετικέτες όπως <video>, <audio> και <canvas>, καθώς και την ενσωμάτωση περιεχομένου scalable vector graphics (SVG) και μαθηματικές φόρμουλες MathML. Τον Ιούλιο 2015, δημοσιεύθηκε το νεότερο πρόχειρο του προτύπου "HTML5.1".

Για τη δημιουργία ενός αρχείου HTML, απαιτείται μόνο ένας απλός επεξεργαστής κειμένου, ενώ επίσης δεν απαιτείται μεταγλώττιση του κώδικα ή η εγκατάσταση επιπλέον λογισμικού web. Τα αρχεία html έχουν κατάληξη .html ή .htm.

Ως πρότυπο υλοποίησης της διεπαφής της εφαρμογής, χρησιμοποιήθηκε το HTML5 λόγω των νεότερων χαρακτηριστικών και δυνατοτήτων που προσφέρει.

CSS3

Χρησιμοποιήθηκε για την ανάπτυξη των παρουσιαστικών στοιχείων του ιστοχώρου [9].

Το CSS είναι μια γλώσσα φύλλων στυλ, η οποία χρησιμοποιείται ως μέρος των ιστοσελίδων, για να ορίζει πώς θα εμφανίζονται τα στοιχεία HTML.

Προστέθηκε στην HTML για να λύσει το πρόβλημα του διαχωρισμού της μορφοποίησης και του περιεχομένου, ενώ τα αρχικά του σημαίνουν "Cascading Style Sheets".

Ο διαχωρισμός αυτός πέραν του βασικού προβλήματος που επιλύει, καθιστά δυνατή την παρουσίαση της ίδιας ιστοσελίδας ή αντικειμένου σε διαφορετικά στυλ, για διαφορετικές μεθόδους απόδοσης, όπως για παράδειγμα ανάλογα με το μέγεθος της οθόνης ή της συσκευής στην οποία γίνεται η προβολή.

Η πρώτη έκδοσή του "CSS1" δημοσιεύθηκε από το W3C το 1996, ενώ το 2002 οριστικοποιήθηκε το "CSS2". Αν και επίσημα δεν έχει υπάρξει δημοσίευση ενός προτύπου με το όνομα "CSS3", σήμερα το W3C εργάζεται στην ανάπτυξη του προτύπου "CSS4" καθώς εντοπίζονται νέες ανάγκες.

Σήμερα η χρήση του CSS αποτελεί αναπόσπαστο σημείο μίας ιστοσελίδας, ενώ πάνω σε αυτό έχουν στηριχθεί νεότερες υλοποιήσεις όπως τα CSS Frameworks και το "Responsive web design" (RWD), τα οποία γνωρίζουν σημαντική επιτυχία τα τελευταία χρόνια.

PHP5

Χρησιμοποιήθηκε για την ανάπτυξη των ιστοσελίδων της εφαρμογής με περιεχόμενο από τη βάση δεδομένων [10].

Η PHP είναι μία γενικής χρήσης γλώσσα προγραμματισμού για δημιουργία δυναμικού περιεχομένου, ενώ τα αρχικά της σημαίνουν "PHP Hypertext Preprocessor". Λειτουργεί σε συνδυασμό με κάποιον διακομιστή web, όπως για παράδειγμα ο Apache HTTP Server, ο Internet Information Services (IIS) κ.ά., ενώ μπορεί να εκτελεστεί σε διάφορα λειτουργικά συστήματα ή πακέτα web υπηρεσιών, όπως το GNU/Linux και τα Microsoft Windows, κάτι το οποίο της προσφέρει το χαρακτηριστικό διαπλατφορμικής υποστήριξης.

Βασικό, επίσης, χαρακτηριστικό της είναι ότι λειτουργεί στον διακομιστή (server-side), το οποίο σημαίνει ότι τα PHP scripts βρίσκονται και εκτελούνται

στον διακομιστή (server) και όχι στον πελάτη (client), ο οποίος δεν έχει άμεση πρόσβαση σε αυτά αλλά μόνο στο παραγόμενο αποτέλεσμα.

Τα PHP scripts περιέχουν κατά βάση κώδικα HTML, CSS και Javascript, όπου παρεμβάλλεται κώδικας php από τον οποίο παράγεται ως αποτέλεσμα συνήθως μια δυναμική ιστοσελίδα, η οποία με την σειρά της προσφέρεται στον πελάτη (client). Υπάρχει η δυνατότητα ακόμα, ο κώδικας της PHP να χρησιμοποιηθεί για εξαγωγή εικόνων, αρχείων PDF, ακόμη και αρχείων Flash. Η PHP μπορεί επίσης να μετατρέψει οποιοδήποτε κείμενο ή δεδομένα σε XHTML και XML μορφή. Επίσης αρκετά συχνά, ειδικότερα για την παραγωγή δυναμικού περιεχομένου, συνδυάζεται με διάφορα συστήματα βάσεων δεδομένων, όπως MySQL. MariaDB, Oracle, ODBC, κά.

Ένα από τα πλεονεκτήματα της PHP είναι η ευκολία συγγραφής κώδικα σε αυτή. Για την δημιουργία ενός PHP script, δεν απαιτείται κάποιο εξειδικευμένο λογισμικό ή εφαρμογή, παρά μόνο ένας απλός επεξεργαστής κειμένου. Επίσης δεν απαιτείται η μεταγλώττιση (compile) των PHP script ή η εγκατάσταση επιπλέον λογισμικού web.

Η ανάπτυξη της PHP ξεκίνησε το 1994. Τον Ιούλιο 2004, κυκλοφόρησε η PHP5, που περνλαμβάνει νέα χαρακτηριστικά, όπως βελτιωμένη υποστήριξη για τον αντικειμενοστραφή προγραμματισμό, τις επεκτάσεις αντικειμένων PHP Data (POP) και πολλές βελτιώσεις επιδόσεων. Ωστόσο, το 2008 η PHP5 έγινε η μόνη σταθερή έκδοση υπό ανάπτυξη, με αρκετές σειρές υποεκδόσεων που συνεχίζουν ως σήμερα (πχ 5.4.x, 5.5.x και 5.6.x.).

Το Δεκέμβριο 2015 παρουσιάστηκε επίσημα η PHP 7.x, η οποία φέρει νέα χαρακτηριστικά και σημαντικές βελτιώσεις ταχύτητας. (Δεν υπήρξε έκδοση με ονομασία PHP 6.x). Στους περισσότερους web διακομιστές, για λόγους συμβατότητας, η βασική εγκατεστημένη έκδοση της PHP είναι η 5.5.x. και η 5.6.x ενώ με ταχύ ρυθμό προχωρά η αντικατάστασή της με τη νεότερη 7.

Τα PHP script έχουν κατάληξη .php, ενώ λόγω των παραπάνω διαφοροποιήσεων χρησιμοποιούνται συχνά και οι καταλήξεις .php5 ή .php7, με τις οποίες δηλώνεται εμμέσως η έκδοση της PHP με την οποία είναι συμβατός ο κώδικας που περιέχει το εκάστοτε αρχείο.

Κατά την υλοποίηση της διεπαφής της εφαρμογής, χρησιμοποιήθηκε PHP5. Ωστόσο, η εφαρμογή έχει δοκιμαστεί και λειτουργεί απροβλημάτιστα και σε διακομιστή (server) με εγκατεστημένη την έκδοση 7.

jQuery σε βάση Javascript

Χρησιμοποιήθηκε για την ανάπτυξη των διαδραστικών χαρακτηριστικών του ιστοχώρου και τη συμβατότητα αυτού με διαφορετικά προγράμματα περιήγησης [11].

Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού, η οποία χρησιμοποιείται ως μέρος των ιστοσελίδων, για να προγραμματίσει τη συμπεριφορά τους. Βασικό, χαρακτηριστικό της είναι ότι εκτελείται από την πλευρά του πελάτη (client-side). Τα αρχεία JavaScript βρίσκονται στον διακομιστή (server) αλλά κατεβαίνουν και εκτελούνται στον περιηγητή του πελάτη (client) κατά τη φόρτωση της ιστοσελίδας.

Κάποιες από τις χρήσεις της JavaScript είναι:

- έλεγχος των πεδίων μίας φόρμας,
- αυτόματη συμπλήρωση σε πλαίσια αναζήτησης,
- φόρτωση πληροφοριών χωρίς την ανάγκη επαναφόρτωσης της ιστοσελίδας,
- χρήση στη γραφική διεπαφή μίας ιστοσελίδας.

Επίσης, σήμερα γνωρίζουν μεγάλη διάδοση υλοποιήσεις της, όπως το Ajax, το οποίο επιτρέπει την ενημέρωση των πληροφοριών των ιστοσελίδων ασύγχρονα, αλλά και η βιβλιοθήκη jQuery.

Η JQuery είναι μια βιβλιοθήκη της JavaScript, που περιέχει συναρτήσεις και χρήση διαφόρων μεταβλητών που απλοποιούν σε μεγάλο βαθμό τον προγραμματισμό σε JavaScript. Αυτό σημαίνει ότι με την χρήση της JQuery μειώνεται η ποσότητα του κώδικα που πρέπει να γραφτεί, κάτι που επιτυγχάνεται με τις διάφορες συναρτήσεις που έχει και τον τρόπο που χειρίζεται τα διάφορα στοιχεία του HMTL εγγράφου για την αλληλεπίδραση του με την ιστοσελίδα. Επίσης τα ονόματα των συναρτήσεων και των

μεταβλητών που χρησιμοποιούνται διακρίνονται για την σαφήνεια τους και για την ευκολία χρήσης τους κάτι που συμβάλλει στο διευκόλυνση της συγγραφής κώδικα από τον προγραμματιστή.

Για την δημιουργία ενός αρχείου JavaScript, απαιτείται μόνο ένας απλός επεξεργαστής κειμένου, κατά αντιστοιχία με την δημιουργία PHP script ή HTML αρχείων. Τα αρχεία JavaScript έχουν κατάληξη .js.

Στα πλαίσια υλοποίησης της εφαρμογής, χρησιμοποιήθηκε για τη βελτίωση των διεπαφών και την εκτέλεση ελέγχων στις φόρμες δεδομένων.

Υλοποίηση της βάσης δεδομένων σε MySQL

Η MySQL είναι ένα από τα δημοφιλέστερα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων. Υποστηρίζει διάφορα λειτουργικά συστήματα και πακέτα web υπηρεσιών, όπως το GNU/Linux και τα Microsoft Windows.

Η βάση δεδομένων δημιουργήθηκε από μια σουηδική εταιρεία, την MySQL AB, με την πρώτη έκδοσή της να εκδίδεται το 1995. Το 2008 η MySQL AB εξαγοράστηκε από την Sun Microsystems, ενώ λίγο αργότερα εκόόθηκε η έκδοση 5.1. Στις αρχές του 2010 η Sun Microsystems εξαγοράστηκε από την Oracle και η MySQL πέρασε κάτω από την διαχείρισή της, κάτι το οποίο συνεχίζει ως σήμερα. Η τελευταία σειρά σταθερών εκδόσεων της MySQL, η 5.6.x εκδόθηκε το 2013, ενώ σήμερα προσφέρεται και η σειρά 5.7.x.

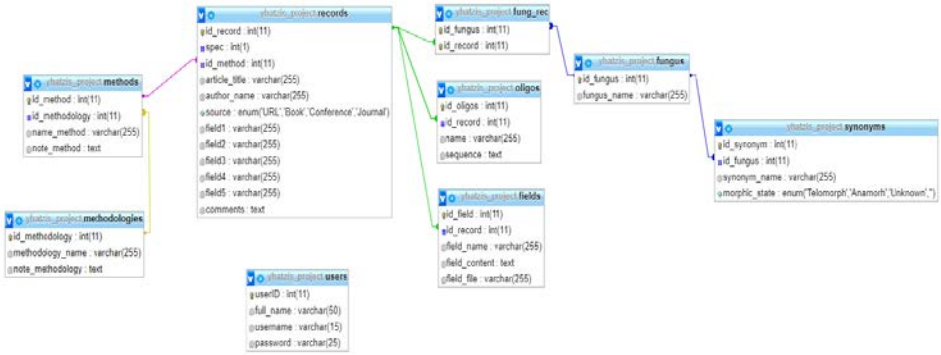
Στα πλαίσια υλοποίησης της εφαρμογής, επιλέχθηκε η χρήση της MySQL 5.6.x και της μηχανής αποθήκευσης InnoDB, ενώ η εφαρμογή δοκιμάστηκε με επιτυχία και στη MariaDB 10.0.x.

Σχεδιασμός της βάσης δεδομένων

Μια από τις περισσότερο σημαντικές διαδικασίες στην ανάπτυξη ενός συστήματος διαχείρισης και επεξεργασίας δεδομένων είναι ο σωστός σχεδιασμός του σχήματος της βάσης δεδομένων, μία διαδικασία που

προκύπτει όταν γίνουν πλήρως αντιληπτά το είδος και ο ρόλος των δεδομένων που θα αποθηκευθούν σ' αυτή καθώς και οι ανάγκες της χρήσης της. Η σωστή σχεδίαση μίας βάσης δεδομένων αποσκοπεί στο διαχωρισμό δεδομένων σε πίνακες με ευκρινή σκοπό. Συγκεκριμένα σε μία εφαρμογή διαδικτύου, είναι σημαντικό να σχεδιάσουμε την βάση με τέτοιο τρόπο ώστε να βελτιστοποιήσουμε την δυνατότητα λήψης και τροποποίησης δεδομένων με τις ελάχιστες δυνατές ερωτήσεις.

Το αποτέλεσμα που προέκυψε, προκειμένου να κατασκευαστεί μια λειτουργική και ευέλικτη βάση δεδομένων, που θα αποτελέσει ένα χρήσιμο εργαλείο στη μελέτη των αποθηκευμένων δεδομένων της αλλά και θα έχει προοπτική ανάπτυξης και επέκτασης, φαίνεται στο παρακάτω σχήμα.



Εικόνα 44 Σχήμα (schema) της βάσης

Ο σχεδιασμός στηρίζεται στο βασικό πίνακα records, ο οποίος περιέχει τα στοιχεία από την καταγραφή των μεθοδολογικών παραμέτρων στη μυκητολογική έρευνα, τόσο των επιτυχημένων (με τη μορφή δημοσιεύσεων) όσο και των αποτυχημένων προσπαθειών που έχουν προκύψει πειραματικά.

Επίσης, ο πίνακας users χρησιμοποιείται για την αποθήκευση των δεδομένων κάθε λογαριασμού χρήστη. Η κάθε εγγραφή στον πίνακα αυτό αντιπροσωπεύει έναν ξεχωριστό λογαριασμό. Ο πίνακας αυτός είναι απαραίτητος για την ύπαρξη ενός ολοκληρωμένου συστήματος χρηστών και αποτελείται από ένα σύνολο γνωρισμάτων τόσο για την επιτέλεση αναγκαίων λειτουργιών, όσο για την ταυτοποίηση αλλά και επικοινωνία με τον χρήστη.

Σχεδιασμός διεπαφής εφαρμογής

Για το σχεδιασμό της εφαρμογής χρησιμοποιήθηκε το Bootstrap framework [12]. Το Bootstrap είναι ένα HTML, CSS και JavaScript πλαίσιο. Παρέχει έτοιμες CSS κλάσεις γενικής χρήσης και JavaScript γεγονότα πάνω σε κάποιες από αυτές τις κλάσεις για συνήθεις λειτουργίες. Δίνει τη δυνατότητα να βασιστεί ο βασικός σκελετός του θέματος εμφάνισης της εφαρμογής πάνω σε ένα δοκιμασμένο και έμπειρο σύστημα. Το πλαίσιο Bootstrap που χρησιμοποιήθηκε είναι η 3η έκδοση και συγκεκριμένα η έκδοση Bootstrap 3.3.7.

Οι δυνατότητες που προσφέρει το Bootstrap είναι πολλές, αλλά υπάρχουν κάποιες από αυτές που οδήγησαν στην επιλογή αυτού του εργαλείου και αξίζει να αναφερθούν ξεχωριστά.

Η πρώτη δυνατότητά του, είναι ότι ανταποκρίνεται σωστά στις διαστάσεις της οθόνης του χρήστη. Καθώς θεωρείται ότι είναι σημαντικό για την εφαρμογή να προσφέρει σε κάθε χρήστη την δυνατότητα προσπέλασης της από ποικιλία συσκευών, είναι φανερό γιατί ενδιαφέρει σε μεγάλο βαθμό μια τέτοια δυνατότητα.

Μία ακόμη δυνατότητα του Bootstrap είναι η χρήση πλέγματος. Οι βασικές κλάσεις που χρησιμοποιεί το Bootstrap για την δημιουργία μίας ιστοσελίδας προτείνουν τον διαχωρισμό των στοιχείων σε σειρές και στήλες δημιουργώντας ένα πλέγμα. Εκτός από την πολύ καλή συνεργία αυτής της τεχνικής με την προηγούμενη δυνατότητα, η χρήση πλέγματος οδηγεί ταυτόχρονα σε μια οργανωμένη εμφανισιακά ιστοσελίδα που προσφέρει στο χρήστη τόσο την καλαισθησία που απαιτείται όσο και την ευκολία κατανόησης και εύρεσης των επιθυμητών λειτουργιών.

Τέλος, το Bootstrap προσφέρει ένα σύνολο κλάσεων και JavaScript λειτουργιών που χρησιμοποιούνται ευρέως και απαλλάσσουν από το χρόνο επαναδημιουργίας τους αλλά ταυτόχρονα προσφέρονται με οργανωμένο τρόπο ώστε να επιτρέπουν την περαιτέρω επέκτασή και εξειδίκευσή τους στις ανάγκες της εφαρμογής. Τέτοιες λειτουργίες ποικίλουν, όπως η δημιουργία

μπάρας πλοήγησης με υπομενού, η σωστή στοίχιση στοιχείων και η επαλήθευση του τύπου δεδομένων σε μία φόρμα αποστολής δεδομένων.

Η χρήση του πλαισίου Bootstrap για το σχεδισμό της διεπαφής της fungibase, έδωσε τις βασικές λειτουργίες που χρειάζονται για τη δημιουργία ενός θέματος εμφάνισης, ενώ παράλληλα επέτρεψε την εύκολη και άμεση τροποποίηση αλλά και εφαρμογή στις απαιτήσεις.

Σύντομη παρουσίαση του πλαισίου Bootstrap

Το Bootstrap δημιουργήθηκε από προγραμματιστές του Twitter, τους Mark Otto και Jacob Thornton, και πλέον συντηρείται από μια μεγάλη κοινότητα προγραμματιστών. Η αναγνώριση του Bootstrap είναι τόσο μεγάλη που πλέον θεωρείται ένα από τα καλύτερα stylesheet framework και υποστηρίζεται από κατασκευαστές.

Περιέχει οδηγίες για το πώς θα μορφοποιηθούν τα στοιχεία της σελίδας. Αρχικά, το αρχείο των προδιαγραφών του bootstrap.css αποθηκεύεται στο σύστημα. Κάθε stylesheet αρχείο, όπως και το bootstrap, υπάρχει σε δύο εκδόσεις. Η πρώτη bootstrap.css χρησιμοποιείται κατά την ανάπτυξη των εφαρμογών και είναι κώδικας που μπορεί να διαβαστεί και να αποσφαλματωθεί, ενώ η δεύτερη bootstrap.min.css είναι μια συμπιεσμένη έκδοση του κώδικα, το minified αρχείο όπως αποκαλείται, δηλαδή δεν είναι σε μορφή που μπορεί να διαβαστεί αλλά προσφέρει μεγαλύτερη ταχύτητα κατά την φόρτωση του. Το bootstrap framework περιέχει και τα αντίστοιχα .js αρχεία, που είναι κώδικας javascript, και πάλι με τις ίδιες εκδόσεις.

Το Bootstrap είναι σπονδυλωτό και αποτελείται ουσιαστικά από μια σειρά στυλ (stylsheets) που εφαρμόζουν τα διάφορα συστατικά του πακέτου εργαλείων. Ένα στυλ που ονομάζεται bootstrap.less περιλαμβάνει τα συστατικά stylesheets. Οι προγραμματιστές μπορούν να προσαρμόσουν το αρχείο Bootstrap, επιλέγοντας τα στοιχεία που θέλουν να χρησιμοποιήσουν στο έργο τους.

Προσαρμογές είναι δυνατές σε περιορισμένη έκταση μέσω ενός κεντρικού στυλ διαμόρφωσης. Η χρήση γλώσσας στυλ επιτρέπει τη χρήση για μεταβλητές, λειτουργίες και φορείς (operators), ένθετους επιλογείς, γνωστά και ως μείγματα mixin.

Από την έκδοση 2.0, η διαμόρφωση του Bootstrap έχει επίσης μία ειδική επιλογή "Προσαρμογή" στην τεκμηρίωση (documentation). Επιπλέον, ο σχεδιαστής του έργου επιλέγει σε μια φόρμα τα επιθυμητά συστατικά και τα προσαρμόζει, εάν είναι αναγκαίο, σε τιμές διαφόρων εναλλακτικών λύσεων για τις ανάγκες του. Στη συνέχεια δημιουργείται ένα πακέτο που περιλαμβάνει ήδη το προ-χτισμένο CSS στυλ.

Τα βασικά πλεονεκτήματα του Bootstrap framework είναι:

- Εύκολο στη χρήση
- Ανταποκρίσιμος σχεδιασμός (responsive design)
- Ταχύτητα ανάπτυξης
- Εύκολα προσαρμοζόμενο ανάλογα με τις ανάγκες
- Συνεκτικότητα
- Υποστήριξη
- Ενσωμάτωση JavaScript
- Εύκολη ενσωμάτωση σε άλλες πλατφόρμες ανάπτυξης
- Ευέλικτο πλέγμα τοποθέτησης στοιχείων σε μια σελίδα
- Προσχεδιασμένα στοιχεία

Σύστημα πλέγματος (Grid System) και ανταποκρίσιμος σχεδιασμός (responsive design)

Το Bootstrap έρχεται με ένα σύστημα πλέγματος που επιτρέπει τη χρήση έως και 12 στηλών σε ολόκληρη τη σελίδα. Όμως, το σύστημα δίνει τη δυνατότητα ομαδοποίησης στηλών ώστε να δημιουργηθούν στήλες μεγαλύτερου πλάτους για τη διάταξη και τη στοίχιση του περιεχομένου. Αυτό δίνει τη δυνατότητα στον προγραμματιστή να δημιουργεί παραλλαγές για χρήση σε διάφορες αναλύσεις οθόνης και τύπους συσκευών, όπως κινητά τηλέφωνα, ταμπλέτες και υπολογιστές με χαμηλή και υψηλή ανάλυση.

Πλήρες σύνολο μορφοποιήσεων CSS

Το Bootstrap παρέχει ένα σύνολο στυλ που παρέχουν βασικούς ορισμούς στυλ για όλα τα βασικά στοιχεία HTML. Αυτά παρέχουν ενιαία, σύγχρονη εμφάνιση για πίνακες, μορφοποίηση κειμένου, καθώς και στοιχεία μιας φόρμας.

Επαναχρησιμοποιήσιμα συστατικά

Εκτός από τα βασικά HTML στοιχεία, το Bootstrap περιέχει και άλλα στοιχεία περιβάλλοντος που χρησιμοποιούνται συχνά. Αυτά περιλαμβάνουν κουμπιά με προηγμένα χαρακτηριστικά (π.χ. ομαδοποίηση κουμπιών ή επιλογή drop-down, οριζόντιες και κάθετες καρτέλες, πλοήγηση, σελιδοποίηση, κλπ.), ετικέτες, προηγμένες τυπογραφικές δυνατότητες, εικονίδια, προειδοποιητικά μηνύματα και μια γραμμή προόδου.

JavaScript στοιχεία

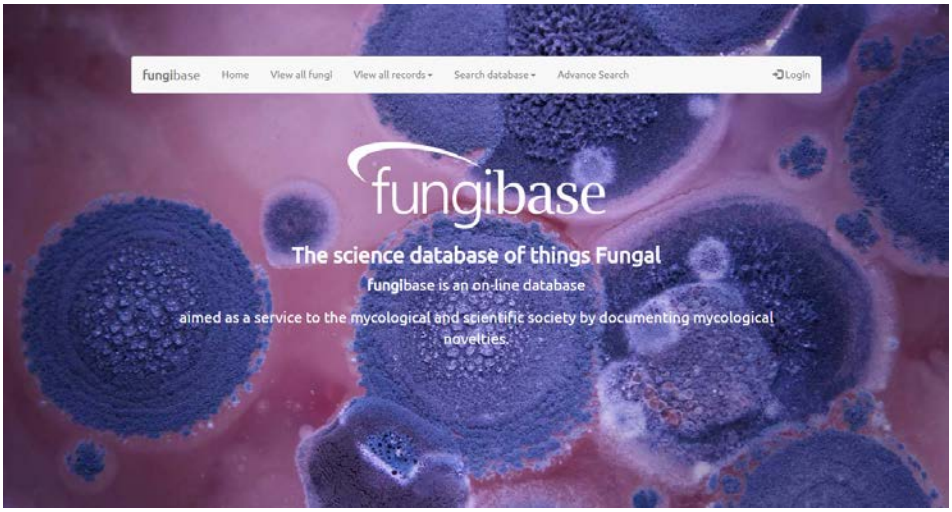
Το Bootstrap έρχεται με πολλά συστατικά JavaScript σε μια μορφή jQuery plugin. Παρέχουν πρόσθετη διεπαφή χρήστη με στοιχεία όπως παράθυρα διαλόγου, επεξηγήσεις, και καρουσέλ. Μπορούν επίσης να επεκτείνουν τη λειτουργικότητα ορισμένων υφιστάμενων στοιχείων της διασύνδεσης, όπως

για παράδειγμα μια αυτόματη πλήρη λειτουργία για πεδία εισαγωγής. Στην έκδοση 2.0, υποστηρίζονται τα ακόλουθα JavaScript plugins: Modal, Αναπτυσσόμενο, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel και Typeahead.

Περιβάλλον διεπαφής εφαρμογής

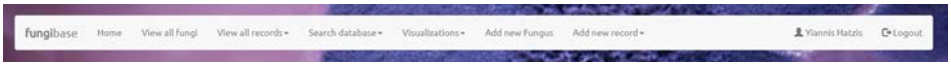
Η σχεδίαση της οθόνης της εφαρμογής είναι απλή και ακολουθεί την ίδια σε δομή σε όλες τις επιμέρους σελίδες ώστε να μην προβληματίζει το χρήστη κατά την πλοήγηση.

Διακρίνεται κεντρικά, το βασικό μενού πλοήγησης της εφαρμογής.



Εικόνα 45 Αρχική οθόνη της fungibase

Μετά την επιτυχή είσοδο στην εφαρμογή, ανάλογα με το ρόλο του χρήστη, ενεργοποιούνται οι επιλογές για τη δυνατότητα προσθήκης νέων εγγραφών στην εφαρμογή.



Εικόνα 46 μενού επιλογών μετά την επιτυχή είσοδο στην εφαρμογή

Διαθέσιμοι ρόλοι χρηστών

Η πιστοποίηση και η εξουσιοδότηση είναι στοιχεία απαραίτητα για μία ιστοσελίδα, η οποία χρειάζεται να περιορίζει την πρόσβαση σε ορισμένους χρήστες. Η πιστοποίηση περιγράφει την πράξη κατά την οποία επαληθεύεται εάν ένα άτομο παρέχει την πραγματική του ταυτότητα. Η διαδικασία αυτή συνήθως απαιτεί ένα όνομα χρήστη και έναν προσωπικό κωδικό, μπορεί όμως να περιλαμβάνει επίσης και άλλες μεθόδους απόδειξης της ταυτότητας, όπως μία «έξυπνη κάρτα», αποτυπώματα, κλπ. Η εξουσιοδότηση αφορά τη διαδικασία κατά την οποία διαπιστώνεται εάν ένα άτομο, αφού πιστοποιηθεί, επιτρέπεται να χρησιμοποιήσει συγκεκριμένους πόρους. Το γεγονός αυτό συνεπάγεται την εξακρίβωση του ρόλου που διαθέτει το άτομο αυτό στην ιστοσελίδα.

Οι ρόλοι ανατίθενται σε ομάδες χρηστών, αλλά μπορεί να ανατίθενται και σε μεμονωμένους χρήστες.

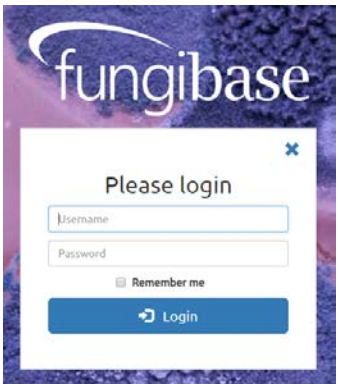
Σε έναν ιστότοπο, εξ ορισμού, υπάρχουν δύο ρόλοι, αυτοί του ανώνυμου χρήστη (ενός χρήστη, ο οποίος δεν έχει συνδεθεί στην ιστοσελίδα) και του πιστοποιημένου χρήστη (ενός χρήστη, ο οποίος έχει συνδεθεί και έχει εξουσιοδοτηθεί). Η χρήση ρόλων μπορεί να επιφέρει πλεονεκτήματα, ακόμη και εάν μία ιστοσελίδα διαθέτει μόνο λίγους χρήστες.

Οι χρήστες της fungibase, διαφοροποιούνται ως προς τους ρόλους τους ανάλογα με τα δικαιώματά τους για εισαγωγή, ανανέωση και διαγραφή δεδομένων. Οι ρόλοι που έχουν μέχρι στιγμής υλοποιηθεί είναι οι εξής:

Ανώνυμος χρήστης (guest). Σ' αυτή την κατηγορία ανήκει ένας χρήστης, ο οποίος δεν έχει συνδεθεί στο σύστημα. Ο χρήστης αυτός δε διαθέτει κανένα δικαίωμα για εισαγωγή, ανανέωση ή διαγραφή δεδομένων. Έχει τη δυνατότητα να ενεργεί αναζητήσεις και να παρακολουθεί αποτελέσματα από αυτές.

Διαχειριστής (administrator). Αποτελεί την ανώτατη βαθμίδα στο σύστημα ρόλων και έχει πλήρη δικαιώματα για εισαγωγή, ανανέωση και διαγραφή δεδομένων.

Ωστόσο, έχει προβλεφθεί (δεν έχει ακόμη υλοποιηθεί) και ένας ενδιάμεσος ρόλος. Ο ρόλος του **Ενδιάμεσου (moderator)** αφορά το χρήστη που διαθέτει δικαιώματα εισαγωγής, ανανέωσης και διαγραφής δεδομένων, οποιαδήποτε όμως ενέργειά του θα πρέπει να εγκριθεί από το διαχειριστή της εφαρμογής, πριν οριστικοποιηθεί και δημοσιευθεί.



Εικόνα 47 Φόρμα εισαγωγής στοιχείων εισόδου χρήστη

Για την είσοδό του ο χρήστης χρησιμοποιεί τη διαδικασία σύνδεσης (Login), όπου για την πιστοποίησή του απαιτείται ένα όνομα χρήστη (username) και ένας προσωπικός κωδικός (password).

Μετά την επιτυχή εισαγωγή του, ο χρήστης έχει τη δυνατότητα τροποποίησης του προσωπικού του κωδικού εισόδου (password).

Εικόνα 48 Φόρμα αλλαγής προσωπικού κωδικού χρήστη (password)

Εισαγωγή δεδομένων

Η fungibase παρέχει δυνατότητα εισαγωγής νέων δεδομένων από πιστοποιημένους χρήστες, ένα ακόμη χαρακτηριστικό, το οποίο συμβάλλει στην αποτελεσματικότητά της.

Βασική δυνατότητα προσθήκης δεδομένων στη βάση δεδομένων, είναι η εισαγωγή μήκυτα, ο οποίος δεν υπάρχει ήδη καταχωρημένος στη βάση δεδομένων. Υπάρχει πρόβλεψη για την αποφυγή επανακαταχώρησης μύκητα με την ίδια ονομασία. Η διαδικασία εισαγωγής πραγματοποιείται από την παραπάνω φόρμα.

Εικόνα 49 Φόρμα εισαγωγής νέου μύκητα

Ώστόσο, η περισσότερο σημαντική λειτουργία είναι η δυνατότητα προσθήκης των δεδομένων για τα οποία έχει σχεδιαστεί η εφαρμογή. Συγκεκριμένα, τα στοιχεία που αφορούν την καταγραφή των μεθοδολογικών παραμέτρων από τη μυκητολογική έρευνα, τόσο των επιτυχημένων (με τη μορφή δημοσιεύσεων) όσο και των προσπαθειών που έχουν πραγματοποιηθεί σε ερευνητικά εργαστήρια. Ο διαχωρισμός αυτών των καταγραφών στην εφαρμογή αναφέρεται, για τις μεν πρώτες ως "Records with reference" και για δε τις δεύτερες ως "Unreferenced".

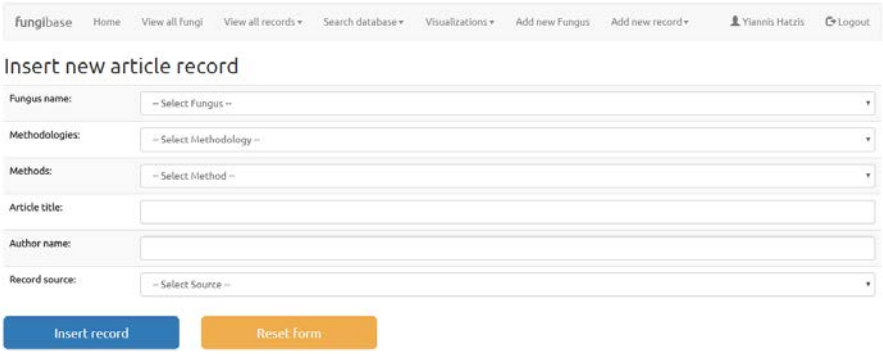
Κατά τη διαδικασία προσθήκης νέας καταχώρησης, επιλέγεται η αντίστοιχη επιλογή από το μενού και εμφανίζεται η φόρμα με τα απαραίτητα πληροφοριακά πεδία εισαγωγής για κάθε περίπτωση.

Οι πληροφορίες που απαιτούνται για την πλήρη εισαγωγή μιας νέας καταχώρησης είναι:

Για τις καταγραφές με δημοσίευση, είναι απαραίτητα:

Fungus name, Methodologies, Methods, Article title, Author name και Record source. Το πεδίο Record source επιτρέπει αποκλειστικά 4 επιλογές (URL, Book, Conference, Journal) και ανάλογα με την επιλογή ζητούνται επιπλέον οι αντίστοιχες απαραίτητες πληροφορίες.

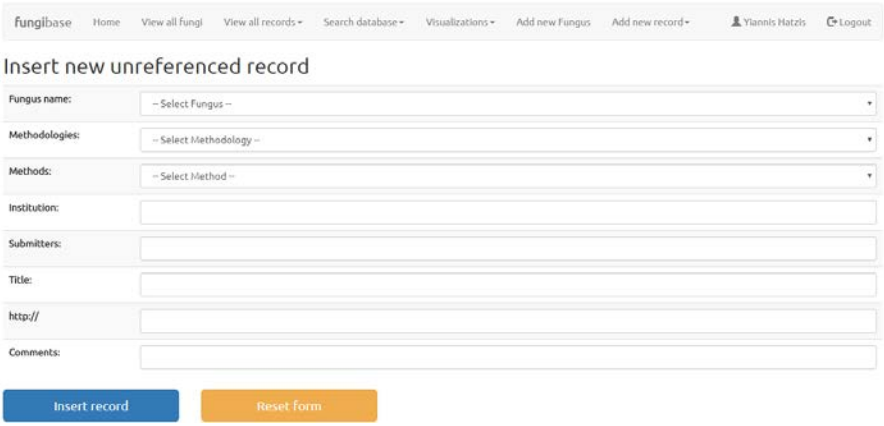
- URL > Σύνδεσμος (http://)
- Book > Book title, Year of publish, Pages
- Conference > Main title, Year, City, More info
- Journal > Journal title, Year, Volume, Pages, DOI ή σύνδεσμος (http://)



Εικόνα 50 Φόρμα εισαγωγής καταχώρησης με δημοσίευση (with reference)

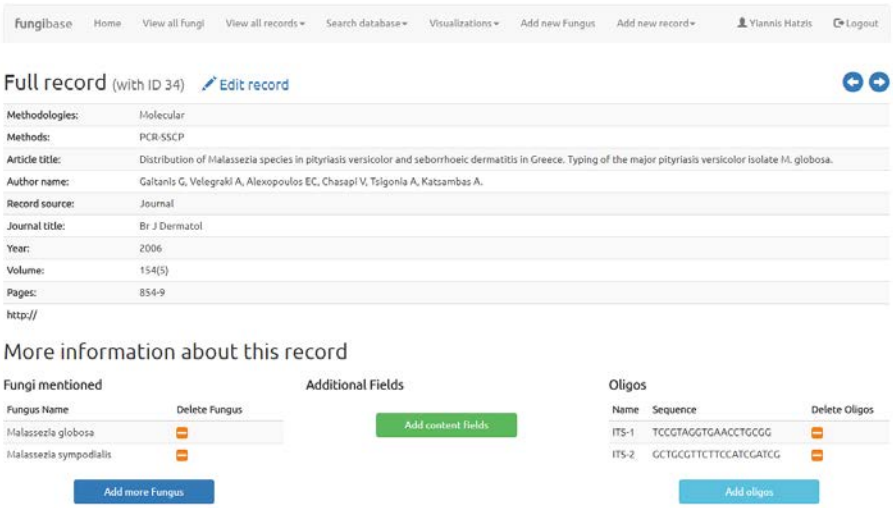
Για τις καταγραφές χωρίς δημοσίευση, είναι απαραίτητα:

Fungus name, Methodologies, Methods, Institution, Submitters, Title, Σύνδεσμος (http://), Comments.



Εικόνα 51 Φόρμα εισαγωγής καταχώρησης χωρίς δημοσίευση (unreferenced)

Μετά την επιτυχημένη εισαγωγή της καταχώρησης δίνεται η δυνατότητα στο χρήστη να επεξεργαστεί την εγγραφή αλλά και προσθέσει περισσότερα δεδομένα ή να αφαιρέσει, για τη συγκεκριμένη μυκητολογική έρευνα και συγκεκριμένα:



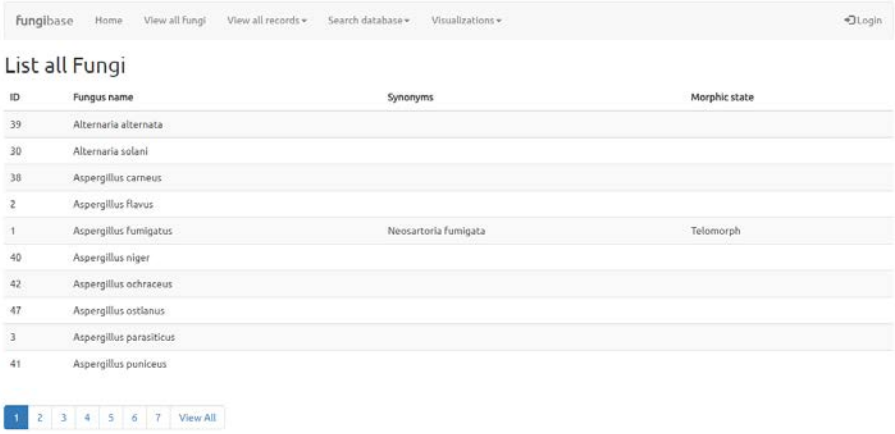
Εικόνα 52 Φόρμα επεξεργασίας της καταχώρησης αλλά και προσθήκης ή διαγραφής επιπλέον δεδομένων

Προβολή και Αναζήτηση δεδομένων

Στη fungibase ο χρήστης έχει τη δυνατότητα να περιηγηθεί και να εμφανίσει όλες τις καταχωρημένες εγγραφές. Φυσικά όμως παρέχεται και η δυνατότητα για περιορισμό των εγγραφών, μέσα από μια διαδικασία αναζήτησης με συγκεκριμένα κριτήρια.

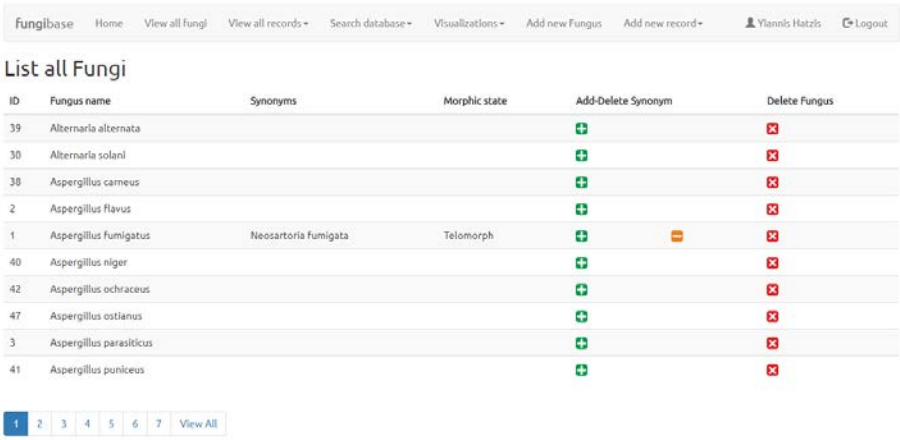
Προβολή όλων των μυκήτων της βάσης

Δίνεται η δυνατότητα σε όλους τους επισκέπτες της fungibase, από το μενού "View all fungi" να πλοηγηθούν σε όλους τους καταχωρημένους μύκητες της βάσης δεδομένων.



Εικόνα 53 Προβολή λίστας με όλους τους καχωρημένους μύκητες

Σε περίπτωση που έχει ήδη γίνει επιτυχής είσοδος με ρόλο Διαχειριστή, δίνεται στο χρήστη επιπλέον η δυνατότητα διαγραφής μύκητα ή η προσθήκη ή αφαίρεση συνώνυμης ονομασίας του.



Εικόνα 54 Μετά την είσοδο με ρόλο Διαχειριστή δίνονται δυνατότητες προσθήκης ή διαγραφής

Προβολή όλων των καταχωρήσεων

Ο επισκέπτης της fungibase έχει τη δυνατότητα, από το μενού "View all records" να επιλέξει τις καταχωρήσεις "With reference" ή τις "Unreferenced" και να πλοηγηθεί στην εμφανιζόμενη λίστα.

fungibase Home View all fungi View all records Search database Visualizations Login

List all Records with reference

ID	Fungus name	Methodologies	Methods	Article title	Author name	Record source	View Record
9	Candida glabrata	Molecular	Simplex PCR	Treatment of invasive candidiasis in the elderly: a review.	Flevari A, Theodorakopoulou M, Velegraki A, Armaganidis A, Dimopoulos G	Journal	
8	Malassezia	Molecular	Simplex PCR	Efficient identification of Malassezia yeasts by matrix-assisted laser desorption ionization-time of flight mass spectrometry (MALDI-TOF MS)	Kolecka A, Khayhan K, Arabatzis M, Velegraki A, Kostrzewa M, Andersson A, Scheynius A, Cafarchia C, Iatta R, Montagna MT, Youngchim S, Cabañes FJ, Hoopman P, Kraak B, Groenewald M, Boekhout T.	Journal	
7	Fusarium verticillioides	Molecular	Sequencing	Georgiadou SP, Velegraki A, Arabatzis M, Neonakis I, Chatzipanagiotou S, Dalekos GN, Petinaki E.	Cluster of Fusarium verticillioides bloodstream infections among immunocompetent patients in an internal medicine department after reconstruction works in Larissa, Central Greece	Journal	
6	Candida guilliermondii	Molecular	Simplex PCR	Caspofungin at catheter lock concentrations eradicates mature biofilms of Candida lusitaniae and Candida guilliermondii.	Simitsopoulou M, Kyrpitzi D, Velegraki A, Walsh TJ, Roilides E	Journal	
	Candida lusitaniae	//	//	//	//	//	
5	Pseudallescheria	Molecular	Sequencing	Proposed nomenclature for Pseudallescheria, Scedosporium and related genera	Michaela Lackner, G. Sybren de Hoog, Liyue Yang, Leandro Ferreira Moreno, Sarah A. Ahmed, Fritz Andreas, Josef Kaltseis, Markus Nagl, Cornelia Lass-Flörl, Brigitte Risslegger, Günter Rambach, Cornelia Speth, Vincent Robert, Walter B	Journal	
	Scedosporium	//	//	//	//	//	
4	Malassezia	Molecular	Simplex	Malassezia infections in humans and animals:	Velegraki A, Cafarchia C, Gaitanis G, Iatta R, Boekhout T.	Journal	

Εικόνα 55 Πλήρης λίστα καταχωρήσεων στη βάση δεδομένων

Έχει τη δυνατότητα να επιλέξει, από το σχετικό εικονίδιο στα δεξιά της, την προβολή της καταχώρησης με όλα τα λεπτομερή στοιχεία της.

fungibase Home View all fungi View all records Search database Visualizations Login

Full record (with ID 24)

Methodologies:	Molecular
Methods:	PCR-REA
Article title:	Identification of medically significant fungal genera by polymerase chain reaction followed by restriction enzyme analysis
Author name:	Aristea Velegraki , Manousos E. Kambouris , George Skiniotis , Marianna Savala Angeliki Mitroussia-Ziouva , Nicholas J. Legakis
Record source:	Journal
Journal title:	FEMS Immunology and Medical Microbiology
Year:	1999
Volume:	23
Pages:	303-312
http://	

More information about this record

Fungi mentioned

Fungus Name
Aspergillus fumigatus
Aspergillus flavus
Aspergillus parasiticus

Additional Fields

Field Name	Field Content	File
Restriction enzymes	MspI, HinfI, EcoRI, HaeIII	

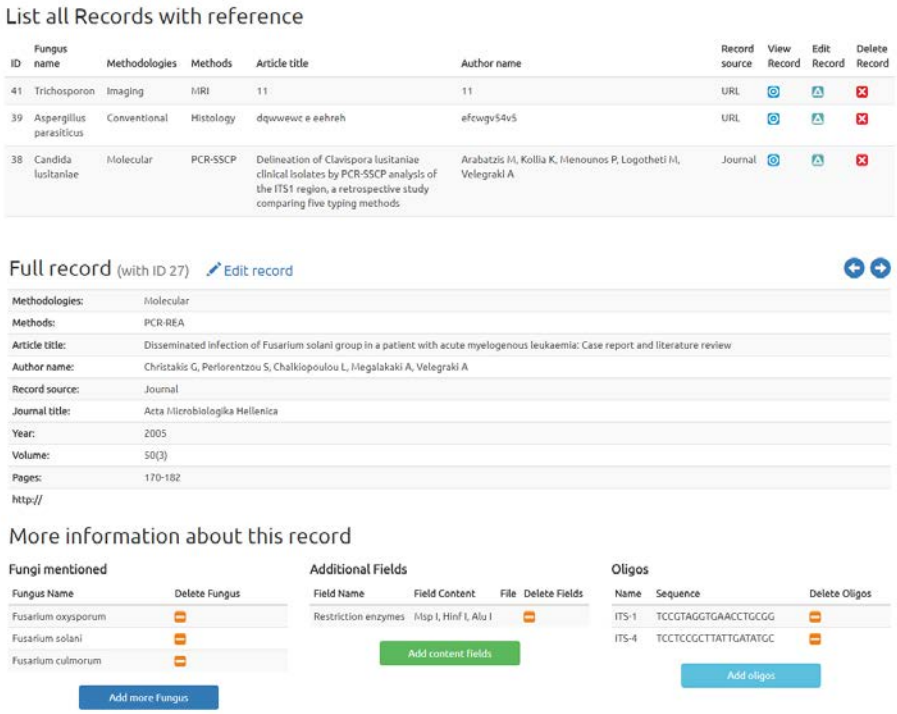
Oligos

Name	Sequence
ITS-4	TCCTCCGCTTATTGATATGC
ITS-1	TTCGTAGGTGAACCTGCGG

Εικόνα 56 Προβολή καταχώρησης με τα πλήρη στοιχεία της

Επίσης μέσω της παραπάνω φόρμας μπορεί να πλοηγηθεί στην επόμενη ή στην προηγούμενη καταχώρηση.

Σε όλες τις παραπάνω φόρμες ενεργοποιούνται δυνατότητες επεξεργασίας, προσθήκης ή διαγραφής στην προβαλόμενη καταχώρηση, μετά την επιτυχή είσοδο χρήστη με ρόλο Διαχειριστή.



Εικόνα 57 Δυνατότητες επεξεργασίας καταχώρησης μετά την είσοδο Διαχειριστή

Διαθέσιμες φόρμες αναζήτησης

Οι διαθέσιμες αναζητήσεις που μπορεί να κάνει ο χρήστης για τον εντοπισμό και την εμφάνιση των δεδομένων της ΒΔ είναι, θέτοντας κριτήρια για:

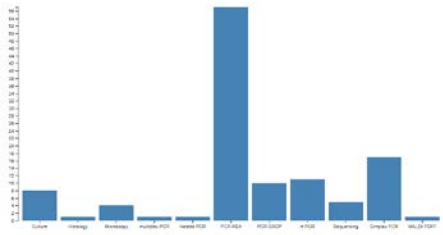
- το όνομα του μύκητα
- τη μέθοδο ανάλυσης
- συγκεκριμένα πρόσθετα πεδία (fields)

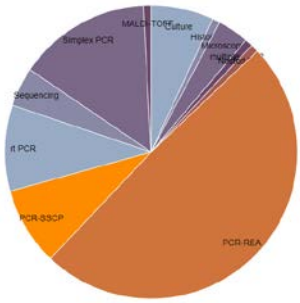
Εργαλεία οπτικοποίησης δεδομένων

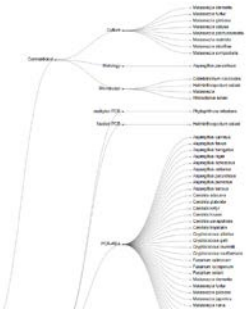
Για την οπτικοποίηση των δεδομένων που αποθηκεύονται στη fungibase χρησιμοποιήθηκαν εργαλεία που προέρχονται από τη βιβλιοθήκη **D3.js** [13]. Ο κύριος λόγος που επιλέχθηκε αυτή τη βιβλιοθήκη ήταν γιατί χρησιμοποιεί τεχνολογίες HTML, SVG και CSS και είναι συμβατή με όλους τους περιηγητές. Συνεργάζεται δε, άψογα με τις τεχνολογίες που χρησιμοποιήθηκαν κατά την ανάπτυξη της fungibase.

Τα εργαλεία οπτικοποίησης δεδομένων που εφαρμόστηκαν στη βάση δεδομένων είναι:

- Bar
- Pie
- Tidy Tree
- Radial Tidy Tree
- Bubble Chart
- Treemap







Εικόνα 58 Δείγματα από διαθέσιμες οπτικοποιήσεις δεδομένων

Η προβολή των οπτικοποιήσεων είναι διαθέσιμη για όλους τους επισκέπτες της εφαρμογής χωρίς να είναι απαραίτητη η είσοδος του χρήστη.

Αναφορές – Βιβλιογραφία κεφαλαίου

1. Καμπούρης, Ε.Μ., Βελεγράκη, Α. (2016). Εισαγωγή στη Μυκητολογία, Εκδ Παρισιάνου, Αθήνα
2. Eckerson WW. Three tier client/server architecture: achieving scalability, performance, and efficiency in client server applications. Open Information Systems. 1995;3:20.
3. https://mathe.ellak.gr/?page_id=132
4. https://dev.mysql.com
5. https://httpd.apache.org
6. https://el.wikipedia.org/wiki/PHP
7. https://www.w3schools.com/php/default.asp
8. https://www.w3schools.com/html/html5_intro.asp
9. https://www.w3schools.com/css
10. https://www.ntchosting.com/encyclopedia/scripting-and-programming/php/php5/
11. https://jquery.com
12. https://getbootstrap.com
13. http://synthesis.sbecker.net/articles/2012/07/08/learning-d3-part-1

Η πρόσβαση στις ηλεκτρονικές πηγές - αναφορές επικαιροποιήθηκε τον Μάρτιο 2018.

Κεφάλαιο 6ο
Συμπεράσματα

Αποτίμηση της χρήσης της RGDtrip

Με τη συνεχή αύξηση του όγκου δεδομένων, η ανάλυση γίνεται πιο δύσκολη. Ένας τρόπος αντιμετώπισης του προβλήματος φαίνεται να είναι ο συνδυασμός εφαρμογών εξόρυξης δεδομένων με προηγμένες έννοιες απεικόνισης. Αυτή η προσέγγιση συνδυάζει τις αναλυτικές δυνατότητες του τομέα πληροφορικής με τον ανθρώπινο παράγοντα, εμπλέκοντας τον τελευταίο στην εξερεύνηση δεδομένων σε πραγματικό χρόνο. Η απεικόνιση των αποθηκευμένων δεδομένων και η διαδικασία της εξόρυξης δεδομένων επιτρέπουν στο χρήστη να συμμετέχει ενεργά στη διαδικασία, ενώ διατηρεί μια οπτική γωνία και καλύτερη αντίληψη του συνόλου δεδομένων. Αυτός ο συνδυασμός ταχύτητας και ευελιξίας επιτρέπει την αφομοίωση και τη διερεύνηση μεγάλων όγκων δεδομένων.

Το κύριο χαρακτηριστικό των πρωτεϊνών που περιέχονται στη βάση δεδομένων RGDtrip είναι η ύπαρξη του τριπεπτιδίου RGD. Η βάση δεδομένων διευκολύνει τη συσσώρευση νέων βιολογικών εισόδων, η οποία αυξάνεται συνεχώς. Αν και η εξελικτική σύγκλιση είναι πιο δύσκολη, εν τούτοις, μπορούν να ανιχνευθούν διαφορετικές αλλά ανάλογες πρωτεΐνες. Μεγάλη πρόκληση δημιουργείται με την εξελικτική απόκλιση. Συγκεκριμένα, αν το RGD τροποποιηθεί ή αντικατασταθεί από άλλο λειτουργικό πρότυπο, η εναλλακτική δομή δεν θα βρίσκεται στη βάση δεδομένων, ανεξάρτητα από την τοποθεσία και την επικράτηση στη βιόσφαιρα. Σε τέτοιες περιπτώσεις, η απόκλιση πρέπει

να συνάγεται ex silencio και να επαληθεύεται με άλλα μέσα για γνωστές πρωτεΐνες. Η RGDtrip θα μπορούσε τουλάχιστον να δώσει μια πρώτη ιδέα ή μια υποψία για κάτι τέτοιο.

Οι χημικά δραστικές και δομικά πολυσθενείς αλληλουχίες αμινοξέων είναι σημαντικές για την κατανόηση πολλών βιολογικών αλληλεπιδράσεων αλλά και για την πρόβλεψη κοινών αποτελεσμάτων σε συγκεκριμένες προκλήσεις. Έτσι η RGDtrip, η οποία είναι αφιερωμένη σε ένα τέτοιο πρότυπο, μπορεί να χρησιμοποιηθεί με πολλούς τρόπους και στρατηγικές. Στρατηγικές αποκλεισμού, για τελεστές σχεδιασμένους να επηρεάζουν μία πρωτεΐνη ή ομάδα υπολειμμάτων και όχι άλλη από την ίδια κυτταρική θέση, μια κοινή προσέγγιση για βιολογική (γεωργία, βιολογία και ιατρική) και βιοτεχνολογική χρήση και φυσικά για εξελικτική και βιομηχανική έρευνα. Για να ενσωματωθούν οι εξελικτικές μελέτες με τη βιοτεχνολογία, η RGDtrip μπορεί να συνδεθεί ή να ενισχυθεί με μία ή περισσότερες βάσεις δεδομένων σε επίπεδο κώδικα, όπου το τριπεπτίδιο θα κατανεμηθεί στις αντίστοιχες κωδικοποιητικές αλληλουχίες έτσι ώστε να επιλεγεί μια πιο βελτιωμένη προσέγγιση, σύμφωνα με την επιθυμητή εφαρμογή και τους διαθέσιμους πόρους τόσο σε αναλυτικά όσο και σε αναπτυξιακά έργα.

Επίσης, η ιδέα της εξατομικευμένης ιατρικής ειδικότερα αλλά και η προσέγγιση φαρμακογονιδιωματικής στο σύνολό της μπορούν να προωθηθούν και να βοηθηθούν από τέτοια αναδρομικά ερευνητικά πρότυπα, όπου οι παρενέργειες μπορούν να προβλεφθούν ή να αποκλειστούν από τον έλεγχο του δραστικού κινήτρου χωρίς προσφυγή (ή τουλάχιστον, πριν από την προσφυγή) στη μοριακή μεθοδολογία. Όσον αφορά τους φαρμακευτικούς στόχους, μπορεί να αντιμετωπιστούν τόσο οι λοιμώξεις όσο και η αδυναμία και η ανεπάρκεια, μετά από αυτή την προσέγγιση.

Επιπλέον, με την RGDtrip παρουσιάζεται η ενσωμάτωση ενός νέου εργαλείου απεικόνισης μέσα στο περιβάλλον συλλογής και ανάκτησης δεδομένων, για πρωτεΐνες που περιέχουν το τριπεπτίδιο RGD. Αυτή η προσέγγιση επιτρέπει την αναζήτηση δεδομένων αιχμής σε συνδυασμό με ένα προηγμένο, φιλικό προς το χρήστη περιβάλλον απεικόνισης. Οι μελλοντικές

προοπτικές περιλαμβάνουν επιπλέον προσεγγίσεις ερωτημάτων και λύσεις οπτικοποίησης που επιτρέπουν το χειρισμό πολυδιάστατων δεδομένων ώστε να δημιουργηθεί ένα παγκόσμιο δίκτυο απεικόνισης, το οποίο θα επιτρέπει στους χρήστες να εκτελούν πολύπλευρες συγκρίσεις των αποτελεσμάτων αναζήτησης και φυσικά, εναλλακτικές λύσεις προς την τεχνολογία Silverlight PivotViewer. Το τελευταίο έχει περιορισμούς χωρητικότητας όσον αφορά τα εμφανιζόμενα στοιχεία προέλευσης μιας συλλογής, με ένα ανώτατο όριο περίπου 6.000 στοιχείων δεδομένων, ενώ η συλλογή δεδομένων RGDtrip περιλαμβάνει σήμερα περίπου 32.000 (31.537) πρωτεΐνες. Επιπλέον, οι τεχνικές εξόρυξης δεδομένων είναι πιθανό να ενσωματωθούν για την περαιτέρω ενίσχυση και αυτοματοποίηση της ανακάλυψης πολύτιμων πληροφοριών που είναι κρυμμένες σε μεγάλα βιολογικά σύνολα δεδομένων.

Τέλος, δεδομένου ότι η HTML5 τείνει να γίνει ένα πρότυπο για την ανάπτυξη εφαρμογών πολυμέσων Web, μπορεί να χρησιμοποιηθεί για τη μετατροπή ολόκληρης της εφαρμογής σ΄ αυτό το πρότυπο.

Αποτίμηση MS SQL και MySQL σε βιολογικές εφαρμογές

Στις βιολογικές εφαρμογές που αναπτύχθηκαν, στο πλαίσιο της παρούσας ερευνητικής εργασίας, χρησιμοποιήθηκαν δύο διαφορετικές πλατφόρμες βάσης δεδομένων SQL, που τη χρησιμοποιούν αλλά με μικρές παραλλαγές και τείνουν να έχουν μια ελαφρώς διαφορετική σύνταξη. Είναι δύο από τις πιο κοινές πλατφόρμες βάσης δεδομένων στον παγκόσμιο ιστό, η Microsoft SQL και MySQL. Αν και έχουν πολλά κοινά, μπορεί να είναι πολύ δύσκολο να γίνει μετάπτωση από το ένα σύστημα στο άλλο. Αυτό οφείλεται στο γεγονός ότι η πλατφόρμα βάσης δεδομένων που θα επιλεγεί, θα καταλήξει να είναι ο πυρήνας της εφαρμογής, σαν δυναμικό περιεχόμενο που κινείται προς τα εμπρός. Αποθηκεύει, ασφαλίζει και ανακτά όλα τα δεδομένα για τις εφαρμογές. Συνεπώς, πρόκειται για μία σημαντική απόφαση και πολύ πιθανό να είναι αμετάκλητη.

Κοινά χαρακτηριστικά

Οι δύο περισσότερο διαδεδομένες πλατφόρμες για χρήση σε εφαρμογές διαδικτύου, παρουσιάζουν πολλά κοινά χαρακτηριστικά.

- Και οι δύο δίνουν τη δυνατότητα να φιλοξενηθούν πολλές βάσεις δεδομένων σε ένα διακομιστή.
- Χρησιμοποιούν πίνακες για την αποθήκευση δεδομένων.
- Υποστηρίζουν πρωτογενή (primary) και ξένα (foreign) κλειδιά για τη συσχέτιση των εγγραφών.
- Χρησιμοποιούν ευρετήρια (indexes) για την ταξινόμηση των δεδομένων με σκοπό να επιταχύνουν την απόδοση, ενώ και οι δύο υποστηρίζουν desktop και web εφαρμογές.
- Το συντακτικό που χρησιμοποιείται για τη δημιουργία των ερωτημάτων είναι κατά βάση παρόμοιο, αν και υπάρχουν ορισμένες διαφορές στη σύνταξη ερωτημάτων CRUD (create, read, update, delete).
- Υπάρχει λογισμικό σύνδεσης (connection drivers) για όλες τις διαδεδομένες γλώσσες προγραμματισμού στο διαδίκτυο. Κατά συνέπεια η γλώσσα προγραμματισμού που θα επιλεγεί δεν αποτελεί δέσμευση στην επιλογή της πλατφόρμας βάσης δεδομένων.

Ωστόσο, ο Microsoft SQL Server απαιτεί περισσότερο αποθηκευτικό χώρο από τον MySQL. Ο Microsoft SQL Server εισήχθη το 1989 και MySQL εισήχθη το 1995 ως ένα έργο ανοικτού κώδικα. Σήμερα ωστόσο και οι δύο έχουν πλέον εδραιωθεί στην αγορά. Ο MySQL εκτελείται είτε σε Windows είτε σε Linux, ενώ ο SQL Server εκτελείται σε Windows.

Και οι δύο πλατφόρμες χειρίζονται από αρκετά μικρές μέχρι και μεγάλες βάσεις δεδομένων και οι επιδόσεις τους είναι παρόμοιες, εφόσον η σχεδίαση της βάσης δεδομένων και ο προγραμματιστής είναι εξοικειωμένοι με το σωστό τρόπο ανάπτυξης και βελτιστοποίησης σύνταξης των SQL ερωτημάτων και του αντίστοιχου προγραμματιστικού κώδικα.

Σημαντικές διαφορές

Εγγενής συμβατότητα: Οι δύο πλατφόρμες λειτουργούν εγγενώς σε συγκεκριμένα λειτουργικά συστήματα. Συγκεκριμένα, αν και μπορεί να χρησιμοποιηθούν οι βάσεις δεδομένων είτε σε Windows είτε σε Linux, ωστόσο η MySQL κυρίως συνεργάζεται με PHP στο Linux ενώ η MS SQL χρησιμοποιείται με το .NET σε Windows.

MyISAM και InnoDB: Η MySQL χρησιμοποιεί δύο διαφορετικές μηχανές για τη διαμόρφωση και διαχείριση των δεδομένων ο οποίες επιτρέπουν στον προγραμματιστή να εκτελέσει πολύ διαφορετικό σχεδιασμό και προγραμματισμό. Αντίθετα η MS SQL, μπορεί να δημιουργήσει μια βάση δεδομένων αποκλειστικά με το δικό της τρόπο.

Εργαλεία διαχείρισης: Και οι δύο πλατφόρμες έχουν εργαλεία διαχείρισης (IDE), αλλά χρειάζεται να συνδυαστούν με το σωστό εργαλείο και το σωστό διακομιστή. Ο MS SQL χρησιμοποιεί το Management Studio ενώ ο MySQL έχει τον Enterprise Manager. Τα εργαλεία αυτά επιτρέπουν τη σύνδεση με το διακομιστή και τη διαχείριση ρυθμίσεων που αφορούν την ασφάλεια, την αρχιτεκτονική και σχεδίαση της βάσης δεδομένων.

Κόστος: Το οικονομικό κόστος είναι μια σημαντική διαφορά μεταξύ τους. Ο SQL Server είναι γενικά ακριβός για να λειτουργήσει, γιατί απαιτούνται άδειες χρήσης για το διακομιστή που εκτελεί το λογισμικό. Αντίθετα ο MySQL παρέχεται δωρεάν και είναι ανοικτού κώδικα (open-source). Ωστόσο, θα απαιτηθεί πληρωμή για υποστήριξη αν χρειαστεί.

Εν κατακλείδι

Η βάση δεδομένων που θα επιλεγεί εξαρτάται κυρίως από το περιβάλλον φιλοξενίας που θα επιλεγεί. Πάροχοι φιλοξενίας που προσφέρουν Linux προσφέρουν συνήθως και MySQL. Δεδομένου ότι η MySQL είναι ανοιχτού κώδικα και δωρεάν, δίνεται η δυνατότητα να δημιουργηθούν όσες βάσεις δεδομένων χρειάζονται χωρίς οικονομικό κόστος.

Αντίθετα, επειδή οι άδειες χρήσης του MS SQL είναι εμπορικά διαθέσιμες και όχι ελεύθερα προσβάσιμες, οι πάροχοι φιλοξενίας που προσφέρουν Windows, χρεώνουν κάθε βάση δεδομένων MS SQL ξεχωριστά. Συνολικά, αυτό κάνει εμφανώς τον SQL Server περισσότερο δαπανηρή λύση από τον MySQL.

Ωστόσο, όπως ήδη έχει αναφερθεί ο SQL Server λειτουργεί εγγενώς με εφαρμογές .NET, είναι κατά συνέπεια η επιλογή για το λογισμικό που εκτελείται σε ένα διακομιστή των Windows, όπου τα εργαλεία ανάπτυξης είναι δωρεάν, αλλά το περιβάλλον παραγωγής δεν είναι δωρεάν.

Επίσης, συχνά σημαντικό ρόλο στην απόφαση επιλογής έχει η προτίμηση του προγραμματιστή, ανάλογα με την εμπειρία του. Οι περισσότεροι προγραμματιστές των Windows επιλέγουν MS SQL και οι προγραμματιστές του Linux MySQL.

Η επιλογή του περιβάλλοντος φιλοξενίας και της τεχνολογίας του λογισμικού που θα απαιτηθεί, θα καθορίσει κυρίως την τελική επιλογή.

Οπτικοποίηση δεδομένων με το PivotViewer και τη D3.js

Κατά την ανάπτυξη των δύο εφαρμογών χρησιμοποιήθηκαν δύο διαφορετικών εργαλείων οπτικοποίησης δεδομένων τα οποία στηρίζονται σε διαφορετικές τεχνολογίες και εμφανίζουν τελικά δύο διαφορετικά περιβάλλοντα.

Στην πρώτη εφαρμογή της RGDtrip, η οποία αποτελεί και το βασικό κομμάτι της διατριβής, η υλοποίηση πραγματοποιήθηκε με χρήση των εργαλείων Microsoft Pivot Viewer και Silverlight και είχε σαν στόχο τη μαζική αναπαράσταση των δεδομένων, το φιλτράρισμα και ταξινόμησή τους.

Στη δεύτερη υλοποίηση, αναπτύχθηκε με χρήση των εργαλείων D3,js και third party βιβλιοθηκών οπτικοποίησης και στοχεύει στην ανάδειξη των αλληλεξαρτήσεων των οντοτήτων.

Όπως έχει ήδη αναφερθεί, η ανάπτυξη εργαλείων οπτικοποίησης δεδομένων είναι ραγδαία. Σ' αυτό το περιβάλλον έρευνας και ανάπτυξης

εργαλείων έχουν εμπλακεί και πολλές επιχειρήσεις οι οποίες εκμεταλλεύονται εμπορικά μερικά ισχυρά εργαλεία που ενσωματώνουν βιβλιοθήκες απεικόνισης, όπως το Tableau, το ClickView και το Kibana. Τα προϊόντα αυτά παρέχουν συγκεκριμένα αποτελέσματα στους χρήστες τους, ενώ ταυτόχρονα παρέχουν και συγκεκριμένη υποστηρίξη από τις εταιρείες. Στοιχείο το οποίο συχνά είναι ένας σημαντικός παράγοντας όταν ζητείται από τους χρήστες συγκεκριμένο αποτέλεσμα.

Εντούτοις υπάρχουν περιπτώσεις, για παράδειγμα σε ερευνητικά project, όπου οι ερευνητές καταφεύγουν σε λύσεις ανοικτού κώδικα, όπως η D3.js ή οποιαδήποτε άλλη από τις παρόμοιες βιβλιοθήκες. Δύο περιπτώσεις μπορεί να τους οδηγήσουν σε αυτή την επιλογή:

Α. Όταν απαιτείται να ενσωματωθεί κώδικας από τις βιβλιοθήκες απεικόνισης απευθείας στον κώδικα της εφαρμογής, ώστε να παρέχει ενσωματωμένες δυνατότητες διασύνδεσης χρήστη, χαρτογράφησης ή/και ανάλυσης.

Β. Όταν χρειάζεται να δημιουργήσουν προσαρμοσμένες απεικονίσεις που υπερβαίνουν τα διαθέσιμα "προσυσκευασμένα" εργαλεία. Σ' αυτή την περίπτωση, τα εργαλεία ανοικτού κώδικα, όπως η D3.js, επιτρέπει στον προγραμματιστή να εξαντλεί τη δημιουργικότητά του ώστε να δημιουργήσει το είδος των απεικονίσεων που πιστεύει ότι θα απεικονίσει καλύτερα τα δεδομένα.

Σε ότι το PivotViewer σε σχέση με άλλα εργαλεία οπτικοποίησης, η υπάρχουσα τεχνολογική βάση του δε δίνει δυνατότητες εξέλιξης και ταυτόχρονα διάδοσης. Το Microsoft Live Labs Pivot έχει χρησιμοποιηθεί ήδη με μεγάλη επιτυχία απεικονίσεις βιολογικού ενδιαφέροντος. Όπως η έκδοση Silverlight, αν και ισχυρή, απαιτεί μια συγκεκριμένη προσθήκη στο πρόγραμμα πλοήγησης και δεν ήταν προσβάσιμη σε όλες τις πλατφόρμες. Επιπλέον οι δυνατότητες επέκτασης ήταν περιορισμένες. Σε όλα αυτά έρχεται να προστεθεί και η πρόσφατη κατάργηση της υποστήριξης για το Silverlight από τη Microsoft.

Διέξοδος στα παραπάνω προβλήματα που αντιμετωπίζει η έκδοση του Silverlight Pivotviewer, δίνεται από την προσπάθεια ανάπτυξης ενός

εκτεταμένου προγράμματος προβολής ανοιχτού κώδικα, που σχεδιάζεται ειδικά με βάση την HTML5 και τεχνολογίες JavaScript[1]. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν δυναμικές και διαδραστικές οπτικοποιήσεις εικόνων ή μεγάλων συνόλων δεδομένων, παρέχοντας ένα ισχυρό, αλλά απλό και διαισθητικό περιβάλλον (σχεδόν πανομοιότυπο με του Silverlight Pivotviewer) που λειτουργεί σε οποιοδήποτε σύγχρονο πρόγραμμα περιήγησης ιστού. Επιτρέπει στους χρήστες να βλέπουν τα δεδομένα τους, να φιλτράρουν, να ταξινομούν και να αναγνωρίζουν σχέσεις βάσει των μεταδεδομένων που παρέχονται για κάθε εικόνα. Επειδή η τεχνολογία βασίζεται σε ανοικτά πρότυπα, υπάρχει δυνατότητα ενσωμάτωσης με άλλες βιβλιοθήκες που βασίζονται σε HTML5, όπως D3, iCanplot για στατιστική απεικόνιση και Scribl για γονιδιωματική οπτικοποίηση πολλαπλών περιοχών.

[1] Taylor, S., & Noble, R. (2014). HTML5 PivotViewer: high-throughput visualization and querying of image data on the web. Bioinformatics, 30(18), 2691-2692. doi:10.1093/bioinformatics/btu349

Δημοσιεύσεις

Περιοδικά

Yiannis Hatzis, Trias Thireou, Emmanouil Viennas, Vassilis Atlamazoglou, George K. Papadopoulos, Konstantinos Poulas, Elias Eliopoulos and Giannis Tzimas, RGDtrip: A Database for the Investigation of Proteins Containing the RGD Tripeptide, Current Bioinformatics, volume 12, pages 1-11, 2017, issn 1574-8936/2212-392X, doi 10.2174/1574893612666170711153356

Συνέδρια

Y. Hatzis, T. Thireou, V. Atlamazoglou, E. Viennas, Y. Tzimas, K. Poulas, G.K. Papadopoulos and E. Eliopoulos, RGDtrip: a database for the investigation of proteins containing the RGD tripeptide, 6th Conference of the Hellenic Society for Computational Biology & Bioinformatics, October 7-9, 2011

RGDtrip: a database for the investigation of proteins containing the RGD tripeptide

T. Thireou[1,*,#], Y. Hatzis[2,#], V. Atlamazoglou[1], E. Viennas[2], Y. Tzimas[3], K. Poulas[2], G.K. Papadopoulos[4] and E. Eliopoulos[1]

[1]Department of Agricultural Biotechnology, Agricultural University of Athens, Iera Odos 75, 11855 Athens, Greece, [2]Department of Pharmacy, University of Patras, Patras 26504, Greece, [3] Department of Applied Informatics in Management & Finance, T.E.I. of Messolonghi, Messolonghi 30200, Greece, [4]Department of Floriculture and Landscape Architecture, Epirus Institute of Technology, Kostakioi, 47100 Arta, Greece,

**Correspondence to: Dr. Trias Thireou, thireou.trias@gmail.com, # equal contribution*

Motivation

The Arg-Gly-Asp (RGD) motif is apparently an ancient tripeptide motif (seen in microbial, plant and animal kingdoms) linked to cell adhesion interactions [1, 2]. The sequence is found in collagen, fibrinogen and several other proteins, while the ligands for the RGD tripeptide are found in several integrins as well as other proteins [3-5]. Such interactions have been well-studied in certain cell adhesion proteins regarding their biochemical and structural requirements. In order to be active in cell adhesion the RGD sequence has invariably been found to form a loop or part of one [6]. Consequently, drugs have been designed to block these interactions, mostly taking advantage of the accumulated knowledge about the structure of ligand-receptor at such an interaction [7]. These drugs (e.g. tirofiban, eptifibatide) have been in use in the clinic for over a decade for prevention and treatment of coronary artery thrombosis and their usefulness in comparative drug trials to prevent cardiovascular events has been shown repeatedly [8]. On the other hand, it seems that several animal poisons (mostly from snakes) function by also blocking such interactions, with the protein agents responsible called disintegrins [9]. We noted some 14 years ago that the RGD sequence was found in the extracellular domain of several cell membrane receptors, not apparently linked to cell adhesion (e.g. major histocompatibility proteins class II) [10]. In several such proteins subsequent structural work has shown that indeed the RGD sequence was part of a loop, but only in a few cases has subsequent biochemical work shown an adhesion function in them [11].
We also noted in the originally submitted manuscript then that several a-adrenergic receptors, and a few other receptors, had RGD sequences in their intra-cytoplasmic domains. As there had been no evidence for any function of such sequences we agreed with the reviewer to take it out. Lo and behold, a year later RGD sequences in a number of proteins involved in apoptosis (e.g. caspase-3) were shown to be of fundamental importance, as injection of cyclical RGD-containign peptides into cells blocked the apoptosis process [12].
Our original call for inclusion of information on RGD sequences of proteins in the structural information package of various data banks has gone unheeded. We have thus proceeded to design a dedicated database, called *RGDTrip*, that will not only include all known sequences and structural information on them, but will further provide assessments into the likelihood of participation of the given protein in cell adhesion or apoptosis interactions.

Methods

RGDtrip is a comprehensive database integrating sequence, structural and functional data collected from Uniprot and PDB.
The procedure followed to populate the database is depicted below:

```
Design the RGDtrip database entities model (Figure1)
Search UniProt for RGD tripeptide
For each relevant UniProt Record
    Check if there are any PDB codes (containing the RGD tripeptide)
    Rank PDB records based on resolution
    Extract data from UniProt, PDB and DSSP
Populate the database
```

Figure 1: RGDtrip database entities model

Microsoft Silverlight technology was used to develop RGDtrip web interface, since it offers useful tools for querying large datasets in multiple ways. Massive amounts of data can be presented at once, while offering the ability to explore the relationships that connect them. The end-user can visualize, sort, organize and categorize data dynamically. Additional filtering allows users to zoom in particular details.

Results

Interested researchers can find RGD-linked information for a given protein into a single environment. An RGDtrip record includes general annotation (*gene and protein names, the name and the superkingdom of the source organism, the description of protein's function and the family that the protein belongs to*), ontology, sequence and structure data including the secondary structure of the RGD tripeptide assigned by DSSP, and database cross-references (Figure 2).
RGDtrip currently hosts 31537 records, 628 of which include structural data. The distribution of the database records in the different superkingdoms is shown at Figure 3. RGDtrip is periodically updated in order to include new sequences and structures of interest and also take into account entry changes of the two source databases. It can be accessed at: http://www.biodata.gr/rgddb/management

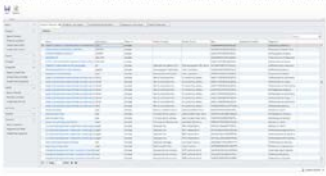


Figure 2: RGDtrip database management web interface

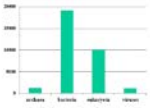


Figure 3: Distribution of RGDtrip records in the different superkingdoms

Discussion

RGDTrip is a new database integrating various data and providing a powerful user friendly web-based multimedia interface for data retrieval and visualization. It may greatly facilitate research into the utility of RGD sequence in various biochemical functions, with cell adhesion and apoptosis being the first two of importance.

References

1. S. E. D'Souza, M. H. Ginsberg, E. F. Plow Arginyl-glycyl-aspartic acid (RGD): a cell adhesion motif. *Trends in Biochemical Sciences*, 16, 246-250, 1991.
2. B.B. Finlay Cell adhesion and invasion mechanisms in microbial pathogenesis *Current Opinion in Cell Biology*, 2, 815-820, 1990.
3. J. Takagi, T. A. Springer. Integrin activation and structural rearrangement. *Immunol Rev.* 186, 141–163, 2002.
4. J. Takagi, B. M. Petre, T. Walz, T. A. Springer. Global conformational rearrangements in integrin extracellular domains in outside-in and inside-out signaling. *Cell* 110, 599–611, 2002.
5. J. Takagi, K. Strokovich, T. A. Springer, T. Walz. Structure of integrin aIIb1 in complex with fibronectin. *EMBO J.* 22, 4607–4615, 2003.
6. Xiong, J. P. et al. Crystal structure of the extracellular segment of integrin aVb3 in complex with an Arg-Gly-Asp ligand. *Science* 296, 151–155 (2002).
7. B. S. Coller. Platelet GPIIb/IIIa antagonists:the first anti-integrin receptor therapeutics. *J. Clin. Invest.* 99, 1467–1471 1997.
8. C. Geequinge , R. Wilcox , P. M. Rath , Triple antiplatelet therapy for preventing vascular events. a systematic review and meta-analysis. *BMC Med.* 2010 Jun 16;8:36.
9. Y. Fujii, D. Okuda, Z. Fujimoto, K. Horii, T. Morita, H. Mizuno. Crystal Structure of Trimestatin, a Disintegrin Containing a Cell Adhesion Recognition Motif RGD Original Research Article. *Journal of Molecular Biology*, 332, 1115-1122, 2003.
10. G.K.Papadopoulos, C. Ozaeounis, E. Eliopoulos. RGD sequences in several receptor proteins: novel cell adhesion function of receptors? *International Journal of Biological Macromolecules*, 22, 51–57, 1998.
11. K. H. Lee, K. W. Wucherpfennig, D. C. Wiley. Structure of a human insulin peptide—HLA-DQ8 complex and susceptibility to type 1 diabetes. *Nat Immunol* 2:501–507, 2001.
12. C. D. Buckley, D. Pilling, N. V. Henriquez, G. Parsonage, K. Threlfall, D. Scheel-Toellner, D. L. Simmons, A. N. Akbar, J. M. Lord, M. Salmon. RGD peptides induce apoptosis by direct caspase-3 activation. *Nature* 397, 534-539, 1999.
13. *Microsoft SilverLight:* ***http://www.silverlight.net***

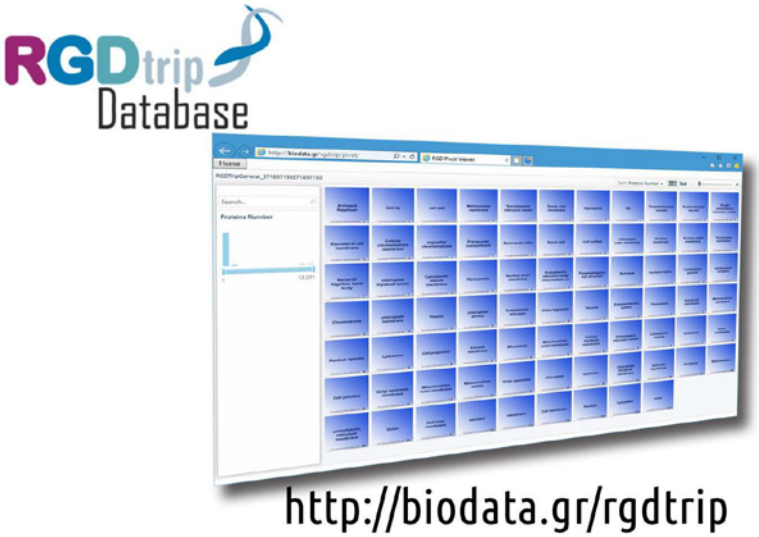
Graphical Abstract

Current Bioinformatics, **2017**, *Vol. 12, No. 00* **000**

RGDtrip: A Database for the Investigation of Proteins Containing the RGD Tripeptide

Yiannis Hatzis, Trias Thireou, Emmanouil Viennas, Vassilis Atlamazoglou, George K. Papadopoulos, Konstantinos Poulas, Elias Eliopoulos* and Giannis Tzimas*

**Department of Biotechnology, Agricultural University of Athens, Iera Odos 75, 11855 Athens, Greece and Department of Computer & Informatics Engineering, Technological Educational Institute of Western Greece, M. Alexandrou 1, Koukouli Patras, 26334, Greece*



Send Orders for Reprints to reprints@benthamscience.ae

Current Bioinformatics, 2017, *12*, 000-000 1

RESEARCH ARTICLE

RGDtrip: A Database for the Investigation of Proteins Containing the RGD Tripeptide

Yiannis Hatzis[1,#], Trias Thireou[2,#], Emmanouil Viennas[3,#], Vassilis Atlamazoglou[2], George K. Papadopoulos[4], Konstantinos Poulas[1], Elias Eliopoulos[2,*] and Giannis Tzimas[5,*]

[1]Department of Pharmacy, University of Patras, Rio Patras, 26500, Greece; [2]Department of Biotechnology, Agricultural University of Athens, Iera Odos 75, 11855 Athens, Greece; [3]Department of Computer Engineering and Informatics, University of Patras, Rio Patras, 26500, Greece; [4]Faculty of Agricultural Technology, Technological Educational Institute of Epirus, Kostakioi, 47100 Arta, Greece and [5]Department of Computer & Informatics Engineering, Technological Educational Institute of Western Greece, M. Alexandrou 1, Koukouli Patras, 26334, Greece

ARTICLE HISTORY

Received: March 25, 2017
Revised: June 3, 2017
Accepted: June 27, 2017

DOI:
10.2174/1574893612666170711153356

Abstract: ***Background:*** The sequence Arginine-Glycine-Aspartic acid (RGD tripeptide) has been identified in most proteins implicated in cell adhesion and signal transduction. Moreover, the RGD paradigm extends to the plant and microbial kingdoms. Investigating this field can be facilitated by combining data from multiple databases into a single one. The RGD tripeptide database is a comprehensive resource with records including general annotation, ontology, database cross-references, sequence and structure data.

Objective: In this work, we present the integration of a novel visualization tool within the RGDtrip 1.0 version data collection and retrieval environment for proteins containing the RGD tripeptide. This approach allows state-of-the-art data querying combined with an advanced, user-friendly visualization environment.

Method: The overall system architecture is based on a three-tier client-server model, thus comprising three main components: the client application, the application server and the database server. The underlying structure of RGDtrip is a relational database developed with Microsoft SQL Server. All the data compiled in RGDtrip were originally scattered in other data bases, such as UNIProt, PDBdb, etc. has been incorporated into a visualization tool based on the Microsoft's PivotViewer software. The tool enables users to see data under many different perspectives and thus to gain a better aspect and understanding of them.

Results: The RGDtrip database may be used for the investigation of proteins containing the RGD tripeptide and the shaping of meaningful conclusions regarding, among other things, evolution, phylogenesis and pharmacological interactions with disease- implicated entities and possible loci of side-effects. The RGDtrip database offers the following main advantages: (i) a collection of about 32,000 proteins containing the RGD tripeptide in just one database and through a unique user interface; (ii) the utilization of state-of-the-art technologies to deliver new data querying and visualization tools for scientists, thus allowing Visual Data Mining, for both basic and applied research on the abovementioned proteins.

Conclusion: This paper describes the integration of existing information with advanced visualization and querying tools, in a dedicated database to implement Visual Data Mining, for basic and applied research on RGD-containing proteins.

Keywords: RGD tripeptide, Querying tools, Visualization tools, Data mining, Cell Adhesion.

*Address correspondence to these authors at the (Giannis Tzimas) Department of Computer & Informatics Engineering, Technological Educational Institute of Western Greece, M. Alexandrou 1, Koukouli Patras, 26334, Greece; E-mail: tzimas@cti.gr
(Elias Eliopoulos) Department of Biotechnology, Agricultural University of Athens, Iera Odos 75, 11855 Athens, Greece; E-mail: eliop@aua.gr
#The authors wish it to be known that, in their opinion, the first 3 authors should be regarded as joint First Authors (in alphabetical order)

1. INTRODUCTION

The tripeptide Arginine-Glycine-Aspartic acid (RGD) is more than a random sequence of aminoacids. It contains one small and neutral of charge aminoacid (Glycine-G) between two larger ones of opposite charges: Aspartic acid (D) with a (-) carboxylate group side chain and Arginine (R), with a (+)

1574-8936/17 $58.00+.00

charged guanidine group. The most common conformation of the RGD tripeptide is a loop and that can be found in all life forms, from virus to human [1], as the constituting aminoacids are coded by the same triplets in all main varieties of the genetic code (by "main" meaning the varieties of the central and not the peripheral genomes, wherever the latter exist, as in plastids or mitochondria). In homologous proteins, the RGD tripeptide may or may not be conserved, the latter case usually showing a change or loss of functionality of the altered homologue [1, 2].

Its first emergence to prominence was through proteins implicated to cell adhesion -a task most common throughout the living organisms and entities [3, 4] but this tripeptide may be more than that.

The loop formation, with the activity-conferring side chains of D and R looking outwards and away of each other and separated by the least conformationally restrained aminoacid residue, Glycine, forming a very active and very distinct electrochemical 3-D entity, is recognizable and contains a positive and a negative charge, at pH=7 [4]; even though the loop as a whole is neutral [5, 6]. This combination allows a wide range of binding, signaling and anchoring interactions, which explains the role of RGD loops in signal transduction as it may stabilize the topology of different ligand entities. It is routinely found in several transmembrane proteins, but also in peripheral membrane proteins, usually facing outwards (extracellularly) for seemingly obvious adhesion purposes, yet cyclical RGD peptides injected into cells at nM concentrations greatly affect cellular transformation, suggesting that intracellular RGD loops must also exist and function [6]. In secreted form, it may assist the binding of the carrier protein onto its target, as in the case of the soluble IGF-1 binding protein-1 (IGFBP-1) [1].

When we presented the first literature review and assessment of all RGD sequences and loops in receptors across species in 1998 [1], we had wondered whether the occurrence of such sequences in receptors might mean that these sequences would be in loop/loop-like structures and whether these loop structures might entail a cell-adhesion function for receptors, as well. In the ensuing time, this has been verified in a number of cases, adding further to the intriguing hypothesis that RGD loops may add a cell adhesion function to receptors. For example, the HLA-DQβ167-169RGD sequence, found in several alleles, was predicted by molecular simulation to be in a loop [7, 8], and was subsequently shown to be in one such loop by crystallography of four different alleles [9-12], with two of the allele structures determined twice by different groups [10, 13] and another two showing the same conformation when bound to a cognate T cell receptor [13, 14]. A function for this well-defined RGD loop in HLA-DQ or their different animal homologues has yet to be found. The cell-adhesion function of RGD sequences in receptors has been shown in one remarkable case: the nucleotide receptor $P2Y_2$ that is up-regulated in the brain by IL-1β-mediated inflammation possesses an RGD sequence in its first extracellular loop; upon activation by UTP, the $P2Y_2$ receptor increases the expression of $\alpha_v\beta_{v3}$ integrins in astrocytes that in turn bind directly to the $P2Y_2$ receptor using this RGD sequence [15]. This interaction is required for downstream astrocyte migration.

The elapsed time has witnessed a much better understanding of the structure of RGD-binding proteins and the intricate mode of the respective ligand-receptor interactions [2, 16-18]. It has become apparent that the RGD-ligand interaction is operative in many fields in biology beyond cell adhesion. The recent determination of the structure of the integrin $\alpha_v\beta_6$-proTGFβ complex leading to the binding of proTGFβ via its RGD sequence in the pro-region, the subsequent processing by furin and the eventual force generation by this integrin for the release of mature TGFβ into the medium for the generation, among other things, of peripheral regulatory T cells, is a case in point [19, 20]. In parallel, we have witnessed the exploitation of the RGD interaction in several ways that have led to successful clinical applications in cardiovascular medicine [21], human tumor imaging [22, 23], cytokine targeting to human tumor sites [24] and surface modification of long-bone implants in small and large animal models [25]. A useful general and recent review is in reference [26]. There are also several clinical trials in Phase II and Phase III for RGD-based anti-tumor pharmaceuticals targeted to specific human tumors, although this field is fraught with difficulties as one failed Stage III trial has shown [27]. The difficulties with RGD-based anti-tumor pharmaceuticals could possibly be overcome in the manner achieved in recent experiments, suggested by the authors for translation into the clinic [28].

This view might be shortsighted, or be only part of the story. The conformation adopted by the tripeptide might well contribute to the distortion or termination a protein strand or helix, as a flexible windle head in mechanical engineering [1]. The applicability of this kind of structure may well be as a recognition/binding site [29] (as in the case of many receptors, to name only the epidermal growth factor receptor [3] and the-soluble- IGF-1 binding protein 1 in humans [1]), as a flexible joint for changing a strand's or helix's direction - perhaps in the case of pyruvate kinase [1] -or as a grappling device for secure binding, as in Fibronectin [4] and in the Tat protein of HIV virus [1]. All of these proposed or attested functions are regulated by neighboring amino acids, which may provide either flexibility or rigidity. The former enlarges the interaction/recognition envelope while the latter limits and contains it [1].

In the original submitted review, but not the published version, [1] we had also pointed out the existence of RGD sequences in the intracellular portions of cell membrane receptors or receptor-associated intracellular proteins [e.g. α-adrenergic receptors and human Grb2 protein). Since then the functionality of the RGD sequence in intracellular processes has been verified, specifically the induction of apoptosis by short RGD-containing peptides acting intracellularly [6]. Furthermore, the list of intracellular proteins or protein segments containing RGD peptides has expanded considerably (the RGDtrip database). It is nonetheless remarkable that the intracellular signal transducing adaptor protein human Grb2 contains two RGD sequences, one each in its N-terminal and C-terminal SH3 domains, the latter RGD in a loop next to residues shown to be essential for interactions with other proteins, but without any test of the RGD loop's functionality [30].

The extreme dispersion of this peptidic pattern throughout the biosphere [1, 29] and the multitude of

individual tasks and applications inherent in such an adaptable and flexible structure [6, 9] create the need for massive data comparison and mining, if it is for its role in biological systems and biomechanics to be elucidated. Phylogenetic and comparative functional research, which substantiate evolutionary tendencies (conservation, convergence and divergence) of functional patterns and detect alternatives, needs robust yet intuitive and user-friendly massive comparison trials to reach correlations initially and, ultimately, results. The subject matter of RGD tripeptides and their uses is the object of many reviews (10 in the first four months of 2017, and 297 in the last ten years [31]). It is worth noting that in spite of so many years of fruitful research on RGD loops, we are not aware of any protein/gene databases that globally report on the presence of the RGD tripeptides.

In order to assist their research in the above mentioned field, investigators are currently collecting the relative information in large spreadsheets which cannot provide any option to explore, analyze and understand data in a user-friendly manner. As a result, there is an imperative need to represent information in a visual form, enabling such interested parties to drill down into the data and gain meaningful insights by revealing underlying patterns and, potentially, previously unseen correlations among large datasets in cumbersome datasheets, which offer limited choices for intuitive data exploration. Thus a need is identified to visualize data while offering potential users the ability to mine data collections from different aspects and thus detect patterns and correlations [32]. This was the motive behind the compilation of the RGDtrip database (available at http://www.biodata.gr/rgdtrip). The resulting data visualization environment for the RGDtrip proteins data collection offers to the user, intuitive interaction with the data so as to handle high data volume while retaining a perspective.

The original RGD tripeptide database (RGDtrip) version included some basic data visualization functionality, which allowed data querying to be coupled to data visualization. In this work, we present the development and implementation of a mature and sophisticated interactive web-based data visualization and querying tool, which allows users to combine large groups of similar elements and identify hidden relationships among individual pieces of information. This tool meets one of the significant challenges of large and complex datasets, that is the effective presentation of and interaction with the data. More specifically, we have built an elegant, web-based multimedia web front-end, based on a software tool launched by Microsoft, namely the PivotViewer. Available: https://www.microsoft.com/silverlight/pivotviewer, 15/5/2017), in order to support a high level visualization of the data collection and of the mining process. It supports dynamic data visualization, sorting, organization and categorization.

The RGDtrip is striving, through successive updates and upgrades, to provide a standardized visualization software implementing Visual Data Mining, in the hope of offering, in addition to the above, a modern educational and diagnostic visualization paradigm to the worldwide community of database developers, curators and engineers.

2. MATERIALS AND METHOD

2.1. System Architecture and Database Structure

The underlying structure of RGDtrip is a relational database developed with Microsoft SQL Server, a flexible software product offering advanced capabilities in database development, manageability, and data warehousing. The application is based on database records such as the protein name, the respective organism, codes from other databases for this protein such UNIprot, PDBdb, Gene3D, SUPfam, Pfam, PIRSF, InterPro and other significant data, such as subcellular location, Gene Ontology (GO) terms and natural variants. The database schema is depicted in Fig. **1**. The overall system architecture is based on a three-tier client-server model [33], thus comprising three main components: the client application, the application server and the database server. The three-tier architecture is intended to allow any of the tiers to be upgraded or replaced independently, in response to changes in requirements or technology. The client application contains only the presentation logic. As a result, less resources are required from the part of the client workstation and no client modification is needed should the database location change. Changes to business logic are automatically enforced by the server and possible future changes are restricted to the application server software that will have to be installed. The three-tier architecture is a robust model, flexible enough to aggregate multiple information sources and integrate modular development [34]. The user interface of the client application tier, the functional process logic ("business logic") of the application server tier, and the computer data storage and the data access (both of the database server tier) are developed and maintained as independent modules.

The primary compilation of the database is by (sub)cellular/viral location of the protein entries, since subcellular-level location is fundamental for the protein function. A sophisticated array of filtering criteria based on the information of an RGDtrip record described in section 2.4 allows further manipulations of the contained proteins.

2.2. Technologies Used

RGDtrip queries can be performed utilizing the PivotViewer control [35, 36], a Silverlight web browser plug-in. Microsoft Silverlight is an application framework for writing and running Rich Internet applications, with features and purposes similar to those of Adobe Flash. PivotViewer was used to implement the main querying interface, since it leverages Deep Zoom which is the fastest, smoothest zooming technology on the Web. As a result, it displays full, high-resolution content without long loading times, while the animations and natural transitions provide context and prevent users from feeling overwhelmed by large quantities of information. The PivotViewer enables users to interact with thousands of objects at once, and sort and browse data in a way that helps them see trends and quickly find what they are looking for. Visualization in data mining is a novel and promising approach for data explanatory analysis, known as Visual Data Mining; it emerged from the technological coupling of automated data mining algorithms and visualization techniques.

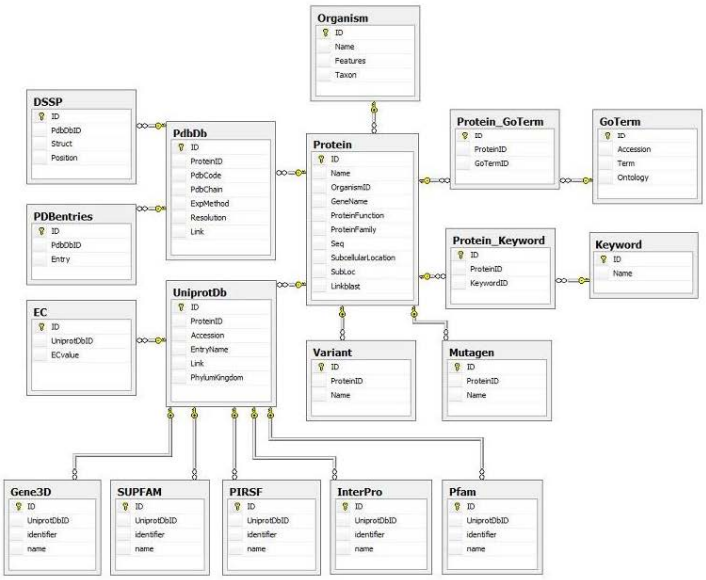


Fig. (1). The overall database schema of RGDtrip.

2.3. Importing Data Environment

The importing tool is another aspect of great importance. With this tool the RGDtrip is up-to-date continuously. Administrators have full access rights to all database functionalities and data and are responsible for updating the data repository. The file containing the data collected from different databases is an ASCII file in a specified format. The data entry is conducted via uploading the ASCII file and a web-based editor subsequently extracts the data from it and feeds it into the database. Modification in RGDtrip will be possible -only for authenticated users- in a next phase. Currently we are also implementing an interface enabling authenticated users to on-line modify the RGDtrip records. The data entry process is at the moment under revision to become more user-friendly.

2.4. Database Compilation and Data Collection

All the data compiled in RGDtrip were originally scattered in other data bases, such as UNIProt, PDBdb, etc. To build the RGDtrip, a straightforward approach was followed, starting from the data collection and the definition/identification of requirements from the users' point of view and resulting in the application prototype. The data collected from the publicly available databases Uniprot and PDB were at first cross-referenced and linked in the RGDtrip database entities model. A significant number of locations (73), both cellular and viral, allow primary search for proteins of interest while retaining a perspective of the whole database. However, a considerable number of entries (12,271) remain unallocated at their source databases and are presented collectively as a single locus card. The large volume and complexity of the RGDtrip protein data collection, as well as the diverse and numerous relationships amongst the data, make it hard for researchers to maintain a global view of the whole dataset. Driven by the need to assist scientists in obtaining a broader picture of the underlying datasets so as to identify and extract hidden correlations, special attention was paid to the development of the visualization of both the data and the querying procedure; thus researchers can interact directly with the huge amount of the available data in an intuitive and meaningful way.

The basic information of an RGDtrip record includes general annotation, ontology, database cross-references, sequence and 3D structure data. These records are derived from UniProtKB [37] sequences that contain the RGD tripeptide. The general annotation features contain the gene and protein names, the name and the super kingdom/domain of the source organism, the Enzyme Commission number,

the description of protein function and family and the protein subcellular location. Ontology data consist of a list of keywords and the selection of GO terms [38] retrieved from UniProtKB, while cross-references to family and domain databases include Gene3D [35], PIRSF [39], Pfam [40], InterPro [41] and SUPFAM [42] data. For position-dependent sequence annotation, information is stored only if the amino-acid position corresponds to the RGD tripeptide. Related records comprise the sites which have been experimentally altered by mutagenesis and natural variants of the protein sequence. If the PDB entries of a UniProtKB record contain the RGD tripeptide, they are sorted based on resolution and structural information is stored for the best resolution PDB code. Data are retrieved from the Protein Data Bank (PDB) [34, 43] and also from the DSSP database [44, 45] in cases of secondary structure assignments.

Sequence similarity searching is one of the most important bioinformatics tasks and often provides the first evidence for the functional, structural, or evolutionary relationships between sequences. Basic Local Alignment Search Tool (BLAST) [46] is probably the most popular similarity search tool. We have used stand-alone BLAST to search each RGDtrip sequence against a locally created database from the FASTA file of all RGDtrip sequences. A link to the corresponding output file is displayed on each protein card. Therefore, the user is able to readily investigate the relationships between the RGD containing protein sequences.

2.5. PivotViewer Visualization Tool

On this basis, we have built a visualization tool based on the Microsoft's PivotViewer software. The tool enables users to see

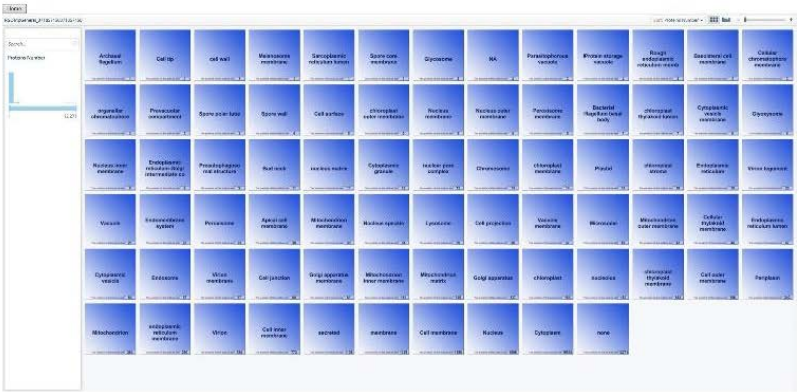


Fig. (2). The first level of the entire RGDtrip data collection, as produced by the visualization tool, with 74 available cards/ subloc s.

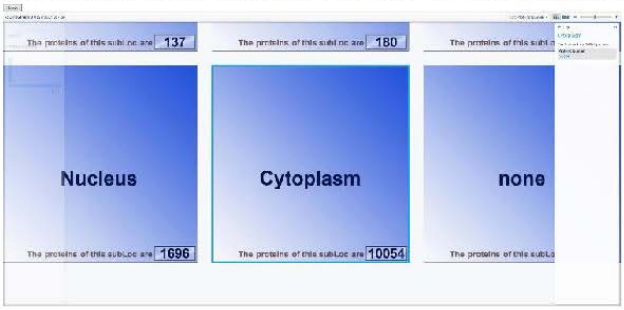


Fig. (3). The investigation of the subloc "Cytoplasm".

data under many different perspectives and thus to gain a better aspect and understanding of them. Moreover, zoom-in options are available from the extensive RGDtrip collective datasets to particular ones. This way the user can detect any interconnections among distant data items and handle them in a way that intuitively reflects their semantic proximity. Once within a subLoc (subcellular location) the search may be conducted with one or more of the available filters which will return proteins of the subLoc meeting the set criteria.

Queries can be performed based on a variety of filtering depicting entry specifics (such as protein name and function, protein family, sequence, respective organism, and source database directory data such as calling/accession numbers and codes) as well as the 3D structure of a particular protein. It is noteworthy here that while a user is experimenting with different querying scenarios, incidental discoveries of potentially high biological importance may be realized.

3. UTILITY AND DISCUSSION

3.1. Entering and Exploring the RGDtrip

This is the first level of the entire RGDtrip data collection, as produced by the visualization tool (Fig. **2**). Each one of the 74 available cards contains proteins grouped by their subcellular location (subLoc), in order to provide a more human-centric visualization approach and displays the name of the subLoc.

After zooming-in for a closer look in each card, the number of proteins contained in the subLoc group and a panel with the card specifics may be seen displayed on the right side (Fig. **3**).

By entering in a data collection (a double click on the card suffices) the visualization tool displays all the proteins of the subLoc group. Each card in the interface represents a protein and the color of the card depends on the "Organism Taxon". Blue cards are for Eukaryota, yellow are for Bacteria, green for Archea and orange cards for Viruses (Figs. **4**, **5**).

A data filtering panel is available, offering a range of 24 filtering criteria to be applied on the underlying data collection (Fig. **5**). PivotViewer application enables users to smoothly and quickly explore the underlying datasets and include, or exclude, specific items by applying filters, whereas the users can simultaneously change the way the resulting set of cards is displayed by choosing between the grid and the graph view by clicking on the corresponding button on upper right corner of the page (Fig. **6**).

This way, users can sort, organize and categorize data dynamically according to characteristics from the data query menu and then zoom-in for a closer look, by either filtering further the collection to get to a subset of interest or by clicking on a particular card. Allowing users to focus on a specific area, or zoom-out to have an overall view of the data, short or long-distance relationships can be uncovered.

The card used for the description of each protein consists of the protein name and family, the organism and the UNIProt Accession data; the color of the card depends on the domain/superkingdom of the organism which produces the protein. Any data available from the PDB database, the PDB Code and the PDB Chain are also displayed, as well as the protein thumbnail on the upper right corner if available (Fig. **7**). A sidebar information panel appears on the right side, when users zoom-in and click on the card (Fig. **8**). The panel provides in-depth information about the protein.

Moreover, for each protein accompanied with structural data from the PDB database, a structural view of it is available: one can use (through a double click on the thumbnail) the Jmol viewer, showing chemical structures in 3D with the position of the RGD tripeptide highlighted in yellow color.

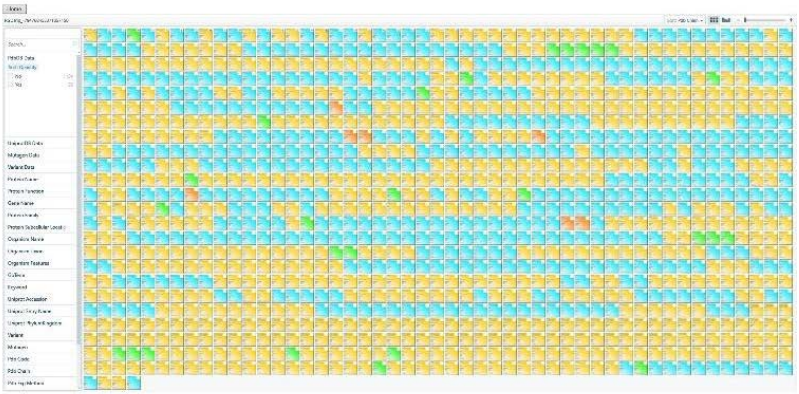


Fig. (4). Each card in the interface represents a protein and the color of the card depends on the "Organism Taxon".

Fig. (5). Change the way the resulting set of cards is displayed by choosing between the grid and the graph view.

More complicated searches could be performed on RGDtrip by acting directly on the database tables using the specified format and table structure.

Fig. (6). A data filtering panel is available, with 24 filtering criteria.

3.2. Demo Query

By exploiting the above functionalities, users have the opportunity to experiment with specific scenarios which may guide them to discover new trends, previously unseen, or perform compound queries. For example, the investigation of the subloc "Cytoplasm" (Fig. **3**) contains 10054 proteins, which are further grouped based on the "phylumKingdom" attribute (Fig. **9**). When applying the corresponding filtering criteria, only 3 of the 1050 "Metazoa" proteins are associated with experimental mutations (Fig. **10**) and 6 with natural variants (Fig. **11**).

Thus, mutations of the RGD tripeptide in specific organisms could be easily located and their effects on protein function could be studied. Moreover, Blast results help analyze conservation of the RGD motif in other organisms and design of experiments to test the impact of RGD mutation on the corresponding protein function.

CONCLUSION

Analyzing high volumes of data becomes more challenging by the day; a way to tackle the issue seems to be the combination of data mining applications with advanced visualization concepts. This approach combines the IT sector's analytical amenities with the human factor by implicating the latter in the data exploration in real time; the visualization of stored data and of the process of data mining allow the user an active part in the procedure while he maintains a vintage point of view and a better appreciation of the data set. This combination of speed and flexibility allows digesting and exploring high volumes of collected data [23].

The core feature of the database is the existence of the RGD tripeptide; this facilitates the deducing of conservation among a number of biological entries -a number ever increasing. The degree and the existence of divergence up-

Fig. (7). The PDB Chain is displayed, as well as the protein thumbnail on the upper right corner if available.

Fig. (8). When users zoom-in on the card the panel on the right-hand side, in-depth information about the protein is provided.

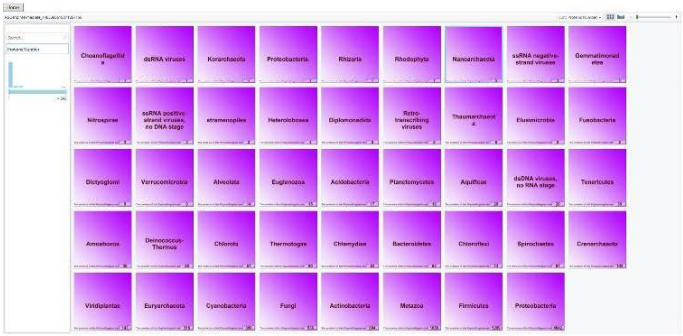


Fig. (9). Proteins of the subloc "Cytoplasm" which are further grouped based on the "phylumKingdom" attribute.

Fig. (10). When applying the corresponding filtering criteria, only 3 proteins are associated with experimental mutations.

Fig. (11). When applying the corresponding filtering criteria, only 6 proteins are associated with natural variants.

or down-stream of the tripeptide are also rather easy to assess. Evolutionary convergence is trickier; still, different but analogous proteins can be detected, too. Things are more challenging with evolutionary divergence. If the RGD is modified or replaced by another functional pattern, the alternative structure would *not* be in the database, irrespectively of location and prevalence in the biosphere. In such cases the divergence must be deduced *ex silencio* and verified by other means; still, for known and sequenced proteins, a first idea or a suspicion-at the very least- is attainable by RGDtrip alone, as already mentioned.

Chemically active and structurally polyvalent aminoacid sequences are important to understand many biological interactions, but also to predict common results in specific challenges. Thus, the RGDtrip, which is devoted to only one such pattern, can be used in many ways and strategies: exclusion strategies for effectors designed to affect one protein/group of residues and not another of the same cellular location, common factor approaches for biological (agricultural, biological and medical) and biotechnological use, and of course for evolutionary and biomechanics research. To intergrate the evolutionary studies with bioengineering, the RGDtrip might be connected or enhanced with one or more code-level databases, where the tripeptide would be allotted to the respective coding sequences so as to select a more streamlined approach, according to the desired application and the available resources in both analytical and developmental projects.

The concept of individualized medicine in particular, but also the pharmacogenomics approach as a whole may be furthered and assisted by such retrospective research patterns, where side-effects might be predicted or ruled out by checking for the/a drastic motive without resorting (or, at least, before resorting) to molecular methodology. As far as pharmaceutical targets are concerned, both infection and inability/inadequacy/degradation syndromes may be dealt with, following this approach [4, 21, 29].

In this work, we present the integration of a novel visualization tool within the RGDtrip 1.0 version data collection and retrieval environment for proteins containing the RGD tripeptide. This approach allows state-of-the-art data querying combined with an advanced, user-friendly visualization environment. Near future prospects include additional querying and visualization approaches; solutions allowing manipulation of multidimensional data so as to create a global visualization network, which will enable users to perform multifaceted comparisons of the querying results and, of course, alternatives to Silverlight PivotViewer technology [32, 36]. The latter has capacity limitations concerning the displayed source items of a collection, with an upper limit of approximately 6,000 data items, while the RGDtrip data collection includes almost 31,537 proteins at the moment. Moreover, data mining techniques are likely to be incorporated to further enhance and automate the discovery of valuable information hidden in large biological datasets. Additionally, an interface enabling authenticated users to on-line modify the RGDtrip records is currently implemented. Lastly, given that the HTML-5 tends to become a standard for the development of Web multimedia applications, it might be used for the visualization of the whole web application.

LIST OF ABBREVIATIONS

RGD = Arginine-Glycine-Aspartic acid tripeptide.

HTML = Hypertext Markup Language.

DECLARATIONS

Availability of Data and Materials

The RGDtrip is a web application for organizing proteins containing the RGD tripeptide. It is public and there are no registration requirements for data querying. RGDtrip can be accessed at http://www.biodata.gr/rgdtrip. The user needs to use one of the browsers Internet Explorer (is not available in the Microsoft Edge) or Firefox, because it is necessary to install Microsoft Silverlight (free web-browser plug-in that enables interactive media experiences) that doesn't work with all available browsers.

Authors' Contributions

Y.H., E.V., K.P., E.E. and G.T. were responsible for the database design, Y.H., E.V. and G.T. were responsible for the software development, T.T, V.A. and G.K.P. were responsible for the data acquisition. Y.H., T.T. and G.K.P. were responsible for manuscript preparation. All authors reviewed and approved the manuscript before being submitted for publication.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

No Animals/Humans were used for studies that are base of this research.

CONSENT FOR PUBLICATION

Not applicable.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

We are indebted to all RGDtrip users worldwide for their valuable comments and suggestions, which helped us to keep the information as updated and complete as possible.

REFERENCES

[1] Papadopoulos GK, Ouzounis C, Eliopoulos E. RGD sequences in several receptor proteins: novel cell adhesion function of receptors? Int J Biol Macromol 1998; 22: 51-7.

[2] Xiong JP, Stehle T, Zhang R, *et al.* Crystal structure of the extracellular segment of integrin aVb3 in complex with an Arg-Gly-Asp ligand. Science 2002; 296: 151-5.

[3] Takagi J, Springer TA. Integrin activation and structural rearrangement. Immunol Rev 2002; 186: 141-63.

[4] D'Souza SE, Ginsberg MH, Plow EF. Arginyl-glycyl-aspartic acid (RGD): a cell adhesion motif. Trends Biochem Sci 1991; 16: 246-50.

[5] Fujii Y, Okuda D, Fujimoto Z, Horii K, Morita T, Mizuno H. Crystal Structure of Trimestatin, a Disintegrin Containing a Cell Adhesion Recognition Motif RGD. J Mol Biol 2003; 332: 1115-22.

[6] Buckley CD, Pilling D, Henriquez NV, *et al.* RGD peptides induce apoptosis by direct caspase-3 activation. Nature 1999; 397: 534-39. See also commentary article in same issue by Ruoslahti E. and Reed J., pp. 479-480.

[7] Routsias J, Papadopoulos GK. Polymorphic structural features of modelled HLA-DQ molecules segregate according to susceptibility or resistance to IDDM. Diabetologia 1995; 38; 1251-61.

[8] Paliakasis K, Routsias J, Petratos K, Ouzounis C, Kokkinidis M, Papadopoulos GK. Novel structural features of the human histocompatibility molecules HLA-DQ as revealed by modelling based on the published structure of the related molecule HLA-DR1. J Struct Biol 1996; 117: 145-63.

[9] Lee KH, Wucherpfennig KW, Wiley DC. Structure of a human insulin peptide—HLA-DQ8 complex and susceptibility to type 1 diabetes. Nat Immunol 2001; 2: 501-7.

[10] Kim CY, Quarsten H, Bergseng E, Khosla C, Sollid LM. Structural basis for HLA-DQ2-mediated presentation of gluten epitopes in celiac disease. Proc Nat Acad Sci USA 2004; 101: 4175-9.

[11] Sethi DK, Schubert DA, Anders AK, *et al.* A highly tilted binding mode by a self-reactive T cell receptor results in altered engagement of peptide and MHC. J Exp Med 2011; 208: 91-102.

[12] Siebold C, Hansen BE, Wyer JR, *et al.* Crystal structure of HLA-DQ0602 that protects against type 1 diabetes and confers strong susceptibility to narcolepsy. Proc Natl Acad Sci U S A 2004; 101: 1999-2004.

[13] Henderson KN, Tye-Din JA, Reid HH, *et al.* A Structural and Immunological Basis for the Role of Human Leukocyte Antigen DQ8 in Celiac Disease. Immunity 2007; 27: 23-34.

[14] Broughton SE, Petersen J, Theodossis A, *et al.* Biased T Cell Receptor Usage Directed against Human Leukocyte Antigen DQ8-Restricted Gliadin Peptides Is Associated with Celiac Disease. Immunity 2012; 37: 1-11.

[15] Peterson TS, Camden JM, Wang Y. P2Y2 Nucleotide Receptor-Mediated Responses in Brain Cells. Mol Neurobiol 2010; 41: 356-66.

[16] Humphries MJ. Insights into integrin-ligand binding and activation from the first crystal structure. Arthritis Res 2002; 4(Suppl 3): S69-78.

[17] Xiong JP, Goodman SL, Arnaout MA. Purification, analysis, and crystal structure of integrins. Methods Enzymol 2007; 426: 307-36.

[18] Luo BH, Carman CV, Springer TA. Structural basis of integrin regulation and signaling. Annu Rev Immunol 2007; 25: 619-47.

[19] Dong X, Zhao B, Iacob RE, *et al.* Force interacts with macromolecular structure in activation of TGF-β. Nature 2017; 542: 55-9.

[20] Ohkura N, Hamaguchi M, Sakaguchi S. FOXP3+ regulatory T cells control of FOXP3 expression by pharmacological agents. Trends Pharmacol Sci 2011; 32: 158-66.

[21] Geeganage C, Wilcox R, Bath PM. Triple antiplatelet therapy for preventing vascular events: a systematic review and meta-analysis. BMC Med 2010; 8: 36.

[22] Meyer A, Auernheimer J, Modlinger A, Kessler H. Targeting RGD-recognizing integrins: drug development, biomaterial research, tumor imaging and targeting. Curr Pharm Des 2006; 12: 2723-47.

[23] Li Z, Huang P, Zhang X, *et al.* RGD-conjugated dendrimer-modified gold nanorods for in vivo tumor targeting and photothermal therapy. Mol Pharm 2010; 7: 94-104.

[24] Corti A, Curnis F, Rossoni G, Marcucci F, Gregorc V. Peptide-Mediated Targeting of Cytokines to Tumor Vasculature: The NGR-hTNF Example. BioDrugs 2013; 27: 591-603.

[25] Förster Y, Rentsch C, Schneiders W. *et al.* Surface modification of implants in long bone. Biomatter 2012; 2: 149-57.

[26] Sun CC, Qu XJ, Gao ZH. Arginine-Glycine-Aspartate-Binding Integrins as Therapeutic and Diagnostic Targets. Am J Ther 2016; 23(1): e198-207.

[27] Marelli UK, Rechenmacher F, Sobahi TRA, Mas-Moruno C, Kessler H. Tumor targeting via integrin ligands. Front Oncol 2013; 3: 1-12.

[28] Kwan BH, Zhu EF, Tzeng A, *et al.* Integrin-targeted cancer immunotherapy elicits protective adaptive immune responses. J Exp Med 2017; 214: 1679-90.

[29] Finlay BB. Cell adhesion and invasion mechanisms in microbial pathogenesis. Curr Opin Cell Biol 1990; 2: 815-20.

[30] Harkiolaki M, Tsirka T, Lewitzky M, *et al.* Distinct binding modes of two epitopes in Grb2 that interact with the SH3C domain of Grb2. Structure 2009; 17: 809-22.

[31] PubMed accessed on 19th May, 2017, 20.18 Eastern European Time, using the terms "RGD" and "review".

[32] Viennas E, Gkantouna V, Ioannou M, *et al.* Population-ethnic group specific genome variation allele frequency data: A querying and visualization journey. Genomics 2012; 100(2): 93-101.

[33] Eckerson WW. Three tier client/server architecture: achieving scalability, performance, and efficiency in client server applications. Open Inform Syst 1995; 3: 20.

[34] Berman HM, Westbrook J, Feng Z, *et al.* The Protein Data Bank. Nucleic Acids Res 2000; 28: 235-42.

[35] Yeats C, Lees J, Reid A, *et al.* Gene3D: comprehensive structural and functional annotation of genomes. Nucleic Acids Res 2008; 36(Database issue): D414-8.

[36] Papadopoulos P, Viennas E, Gkantouna V, *et al.* Developments in FINDbase worldwide database for clinically relevant genomic variation allele frequencies. Nucleic Acids Res 2014; 42(Database issue): D1020-6.

[37] UniProt Consortium. Reorganizing the protein space at the Universal Protein Resource (UniProt). Nucleic Acids Res 2012; 40: D71-D5.

[38] Ashburner M, Ball CA, Blake JA, *et al.* Gene ontology: tool for the unification of biology. Nat Genet 2000; 25(1): 25-9.

[39] Wu CH, Nikolskaya A, Huang H, *et al.* PIRSF: family classification system at the Protein Information Resource. Nucleic Acids Res 2004; 32: D112-D4.

[40] Punta M, Coggill PC, Eberhardt RY, *et al.* The Pfam protein families database. Nucleic Acids Res 2014; 42(Database Issue): D222-D30.

[41] Hunter S, Jones P, Mitchell A, *et al.* InterPro in 2011: new developments in the family and domain prediction database. Nucleic Acids Res 2012; 40(Database issue): D306-12.

[42] Wilson D, Pethica R, Zhou Y, *et al.* SUPERFAMILY - Comparative Genomics, Datamining and Sophisticated Visualisation. Nucleic Acids Res 2009; 37(Database issue): D380-D6.

[43] Rose PW, Beran B, Bi C, *et al.* The RCSB Protein Data Bank: redesigned web site and web services. Nucleic Acids Res 2011; 39(Suppl 1): D392-D401.

[44] Joosten RP, Te Beek TAH, Krieger E, *et al.* A series of PDB related databases for everyday needs. Nucleic Acids Res 2011; 39(Suppl 1): D411-D9.

[45] Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 1983; 22: 2577-637.

[46] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990; 215: 403-10.

Παράρτημα 1

Λογισμικό εισαγωγής δεδομένων στην RGDtrip

Εισαγωγή αρχικών δεδομένων πρωτεϊνών

```
using System;
using System.IO;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using System.Collections;
using System.Collections.Generic;
using System.Threading;

namespace RGD_db_parser
{
 public class keywords
 {
  public string keyword;
 }

 public class variants
 {
  public string variant;
 }

 public class mutagens
 {
  public string mutagen;
 }

 public class DSSP
 {
```

```
 public string structs;
 public string position;
}

public class PDBentries
{
 public string pdbEntry;
}

public class GO_terms
{
 public string accession;
 public string term;
 public string ontology;
}

public static class RGD_insert_data
{
 public static void Main()
 {
  DateTime StartTime = DateTime.Now;

  string connectionString = "Data Source=YHATZIS-STATION;"
   + "Initial Catalog=RGD;"
   + "Integrated Security=True";

  SqlConnection db_connect = null;

  String path = @".\data4import.txt";

  String line;
  String field_name = null;
  String field_data = null;

  // Variables for Table organism
  String organismName = "";
  String organismFeatures = "";
  String taxon = "";

  // Variables for Table protein
  String proteinName = "";
  String geneName = "";
  String proteinFunction = "";
  String proteinFamily = "";
  String seq = "";
  String subcellularLocation = "";

  // Variables for Table PDB_db
  String pdbCode = "";
  String pdbChain = "";
  String expMethod = "";
  float resolution = 0;
```

```
String pdbLink = "";

// Variables for Table UNIPROT_db
String uniprotAccession = "";
String uniprotEntryName = "";
String uniprotLink = "";

// Queue for Table keywords
keywords recordkeywords = null;
recordkeywords = new keywords();
Queue keywordsQueue = new Queue();

// Queue for Table variants
variants recordvariants = null;
recordvariants = new variants();
Queue variantsQueue = new Queue();

// Queue for Table mutagens
mutagens recordmutagens = null;
recordmutagens = new mutagens();
Queue mutagensQueue = new Queue();

// Queue for Table DSSP
DSSP recordDSSP = null;
recordDSSP = new DSSP();
Queue DSSPQueue = new Queue();

// Queue for Table PDBentries
PDBentries recordPDBentries = null;
recordPDBentries = new PDBentries();
Queue PDBentriesQueue = new Queue();

// Queue for Table GO_terms
GO_terms recordGOterms = null;
recordGOterms = new GO_terms();
Queue GOtermsQueue = new Queue();

int first_tab = 0;
int second_tab = 0;

// Counters
int new_organism = 0;
int new_keyword = 0;
int new_GoTermID = 0;
int PDB_data = 0;
int UNIPROT_data = 0;
int both_data = 0;

// table IDs
int organismID = 0;
int proteinID = 1;
int PdbDbID = 0;
```

```
int UniprotDbID = 0;
int keywordID = 0;
int Protein_KeywordID = 0;
int VariantID = 0;
int MutagenID = 0;
int DSSPID = 0;
int PDBEntryID = 0;
int GoTermID = 0;
int Protein_GoTermID = 0;

// SQL commands for inserting data into Tables

String sql_find_organismID = "SELECT ID FROM dbo.Organism "
 + "WHERE Name = @organismName "
 + "AND Features = @organismFeatures "
 + "AND Taxon = @taxon";

String sql_organism = "INSERT INTO dbo.Organism VALUES ("
 + "@organismID, "
 + "@organismName, "
 + "@organismFeatures, "
 + "@taxon)";

 String sql_protein = "INSERT INTO dbo.Protein VALUES ("
 + "@proteinID, "
 + "@proteinName, "
 + "@organismID, "
 + "@geneName, "
 + "@proteinFunction, "
 + "@proteinFamily, "
 + "@seq, "
 + "@subcellularLocation)";

String sql_PDB_db = "INSERT INTO dbo.PdbDb VALUES ("
 + "@PdbDbID, "
 + "@proteinID, "
 + "@pdbCode, "
 + "@pdbChain, "
 + "@expMethod, "
 + "@resolution, "
 + "@pdbLink)";

String sql_UNIPROT_db = "INSERT INTO dbo.UniprotDb VALUES ("
 + "@UniprotDbID, "
 + "@proteinID, "
 + "@uniprotAccession, "
 + "@uniprotEntryName, "
 + "@uniprotLink)";

String sql_find_keywordID = "SELECT ID FROM dbo.Keyword "
 + "WHERE Name = @keyword";
```

```
String sql_keywords = "INSERT INTO dbo.Keyword VALUES ("
 + "@keywordID, "
 + "@keyword)";

String sql_J_keywords = "INSERT INTO dbo.Protein_Keyword VALUES ("
 + "@Protein_KeywordID, "
 + "@proteinID, "
 + "@keywordID)";

String sql_variants = "INSERT INTO dbo.Variant VALUES ("
 + "@VariantID, "
 + "@proteinID, "
 + "@variant)";

String sql_mutagens = "INSERT INTO dbo.Mutagen VALUES ("
 + "@MutagenID, "
 + "@proteinID, "
 + "@mutagen)";

String sql_DSSP = "INSERT INTO dbo.DSSP VALUES ("
 + "@DSSPID, "
 + "@PdbDbID, "
 + "@structs, "
 + "@position)";

String sql_PDBentries = "INSERT INTO dbo.PDBentries VALUES ("
 + "@PDBEntryID, "
 + "@PdbDbID, "
 + "@pdbEntry)";

String sql_find_GOterms = "SELECT ID FROM dbo.GoTerm "
 + "WHERE Accession = @accession";

String sql_GOterms = "INSERT INTO dbo.GoTerm VALUES ("
 + "@GoTermID, "
 + "@accession, "
 + "@term, "
 + "@ontology)";

String sql_J_GOterms = "INSERT INTO dbo.Protein_GoTerm VALUES ("
 + "@Protein_GoTermID, "
 + "@proteinID, "
 + "@GoTermID)";

// Create and open the database connection
try
{
 db_connect = new SqlConnection(connectionString);
 db_connect.Open();
}
catch (Exception err_opendb)
{
```

```
    // Let user know if anything go wrong with open db
    if (db_connect != null)
    {
     Console.WriteLine(err_opendb.Message);
     db_connect.Dispose();
    }
   }

   try
   {
    // Create an instance of StreamReader to read from a file.
    // The using statement also closes the StreamReader.
    using (StreamReader datafile = new StreamReader(path))
    {
     // Read lines from the file until the end of the file is reached.
     while ((line = datafile.ReadLine()) != null)
     {
      if (line == "$$$$")
      {
       SqlCommand find_organismID = new SqlCommand(sql_find_organismID,
db_connect);
       find_organismID.Parameters.AddWithValue("@organismName", organismName);
       find_organismID.Parameters.AddWithValue("@organismFeatures",
organismFeatures);
       find_organismID.Parameters.AddWithValue("@taxon", taxon);
       object found_organism = find_organismID.ExecuteScalar();

       if (found_organism == null)
       {
        organismID = ++new_organism;
        SqlCommand into_organism = new SqlCommand(sql_organism, db_connect);
        into_organism.Parameters.AddWithValue("@organismID", organismID);
        into_organism.Parameters.AddWithValue("@organismName", organismName);
        into_organism.Parameters.AddWithValue("@organismFeatures",
organismFeatures);
        into_organism.Parameters.AddWithValue("@taxon", taxon);
        into_organism.ExecuteNonQuery();
       }
       else
        organismID = Convert.ToInt32(found_organism);

       SqlCommand into_protein = new SqlCommand(sql_protein, db_connect);
       into_protein.Parameters.AddWithValue("@proteinID", proteinID);
       into_protein.Parameters.AddWithValue("@proteinName", proteinName);
       into_protein.Parameters.AddWithValue("@organismID", organismID);
       into_protein.Parameters.AddWithValue("@geneName", geneName);
       into_protein.Parameters.AddWithValue("@proteinFunction", proteinFunction);
       into_protein.Parameters.AddWithValue("@proteinFamily", proteinFamily);
       into_protein.Parameters.AddWithValue("@seq", seq);
       into_protein.Parameters.AddWithValue("@subcellularLocation",
subcellularLocation);
       into_protein.ExecuteNonQuery();
```

```
        if (pdbCode != "")
        {
         PdbDbID++;
         SqlCommand into_PDB_db = new SqlCommand(sql_PDB_db, db_connect);
         into_PDB_db.Parameters.AddWithValue("@PdbDbID", PdbDbID);
         into_PDB_db.Parameters.AddWithValue("@proteinID", proteinID);
         into_PDB_db.Parameters.AddWithValue("@pdbCode", pdbCode);
         into_PDB_db.Parameters.AddWithValue("@pdbChain", pdbChain);
         into_PDB_db.Parameters.AddWithValue("@expMethod", expMethod);
         into_PDB_db.Parameters.AddWithValue("@resolution", resolution);
         into_PDB_db.Parameters.AddWithValue("@pdbLink", pdbLink);
         into_PDB_db.ExecuteNonQuery();
         PDB_data++;
        }

        if (uniprotAccession != "")
        {
         UniprotDbID++;
         SqlCommand into_UNIPROT_db = new SqlCommand(sql_UNIPROT_db, db_connect);
         into_UNIPROT_db.Parameters.AddWithValue("@UniprotDbID", UniprotDbID);
         into_UNIPROT_db.Parameters.AddWithValue("@proteinID", proteinID);
         into_UNIPROT_db.Parameters.AddWithValue("@uniprotAccession",
uniprotAccession);
         into_UNIPROT_db.Parameters.AddWithValue("@uniprotEntryName",
uniprotEntryName);
         into_UNIPROT_db.Parameters.AddWithValue("@uniprotLink", uniprotLink);
         into_UNIPROT_db.ExecuteNonQuery();
         UNIPROT_data++;
        }

        while (keywordsQueue.Count != 0)
        {
         recordkeywords = (keywords)keywordsQueue.Dequeue();
         SqlCommand find_keywordID = new SqlCommand(sql_find_keywordID,
db_connect);
         find_keywordID.Parameters.AddWithValue("@keyword",
recordkeywords.keyword);
         object found_keyword = find_keywordID.ExecuteScalar();

         if (found_keyword == null)
         {
          keywordID = ++new_keyword;
          SqlCommand into_keywords = new SqlCommand(sql_keywords, db_connect);
          into_keywords.Parameters.AddWithValue("@keywordID", keywordID);
          into_keywords.Parameters.AddWithValue("@keyword",
recordkeywords.keyword);
          into_keywords.ExecuteNonQuery();
         }
         else
          keywordID = Convert.ToInt32(found_keyword);
```

```
        Protein_KeywordID++;
        SqlCommand into_J_keywords = new SqlCommand(sql_J_keywords, db_connect);
        into_J_keywords.Parameters.AddWithValue("@Protein_KeywordID",
Protein_KeywordID);
        into_J_keywords.Parameters.AddWithValue("@proteinID", proteinID);
        into_J_keywords.Parameters.AddWithValue("@keywordID", keywordID);
        into_J_keywords.ExecuteNonQuery();
       }

       while (variantsQueue.Count != 0)
       {
        recordvariants = (variants)variantsQueue.Dequeue();
        VariantID++;
        SqlCommand into_variants = new SqlCommand(sql_variants, db_connect);
        into_variants.Parameters.AddWithValue("@VariantID", VariantID);
        into_variants.Parameters.AddWithValue("@proteinID", proteinID);
        into_variants.Parameters.AddWithValue("@variant",
recordvariants.variant);
        into_variants.ExecuteNonQuery();
       }

       while (mutagensQueue.Count != 0)
       {
        recordmutagens = (mutagens)mutagensQueue.Dequeue();
        MutagenID++;
        SqlCommand into_mutagens = new SqlCommand(sql_mutagens, db_connect);
        into_mutagens.Parameters.AddWithValue("@MutagenID", MutagenID);
        into_mutagens.Parameters.AddWithValue("@proteinID", proteinID);
        into_mutagens.Parameters.AddWithValue("@mutagen",
recordmutagens.mutagen);
        into_mutagens.ExecuteNonQuery();
       }

       while (DSSPQueue.Count != 0)
       {
        recordDSSP = (DSSP)DSSPQueue.Dequeue();
        DSSPID++;
        SqlCommand into_DSSP = new SqlCommand(sql_DSSP, db_connect);
        into_DSSP.Parameters.AddWithValue("@DSSPID", DSSPID);
        into_DSSP.Parameters.AddWithValue("@PdbDbID", PdbDbID);
        into_DSSP.Parameters.AddWithValue("@structs", recordDSSP.structs);
        into_DSSP.Parameters.AddWithValue("@position", recordDSSP.position);
        into_DSSP.ExecuteNonQuery();
       }

       while (PDBentriesQueue.Count != 0)
       {
        recordPDBentries = (PDBentries)PDBentriesQueue.Dequeue();
        PDBEntryID++;
        SqlCommand into_PDBentries = new SqlCommand(sql_PDBentries, db_connect);
        into_PDBentries.Parameters.AddWithValue("@PDBEntryID", PDBEntryID);
        into_PDBentries.Parameters.AddWithValue("@PdbDbID", PdbDbID);
```

```
        into_PDBentries.Parameters.AddWithValue("@pdbEntry",
recordPDBentries.pdbEntry);
        into_PDBentries.ExecuteNonQuery();
       }

       while (GOtermsQueue.Count != 0)
       {
        recordGOterms = (GO_terms)GOtermsQueue.Dequeue();
        SqlCommand find_GOterms = new SqlCommand(sql_find_GOterms, db_connect);
        find_GOterms.Parameters.AddWithValue("@accession",
recordGOterms.accession);
        object found_GoTerm = find_GOterms.ExecuteScalar();

        if (found_GoTerm == null)
        {
         GoTermID = ++new_GoTermID;
         SqlCommand into_GOterms = new SqlCommand(sql_GOterms, db_connect);
         into_GOterms.Parameters.AddWithValue("@GoTermID", GoTermID);
         into_GOterms.Parameters.AddWithValue("@accession",
recordGOterms.accession);
         into_GOterms.Parameters.AddWithValue("@term", recordGOterms.term);
         into_GOterms.Parameters.AddWithValue("@ontology",
recordGOterms.ontology);
         into_GOterms.ExecuteNonQuery();
        }
        else
         GoTermID = Convert.ToInt32(found_GoTerm);

        Protein_GoTermID++;
        SqlCommand into_J_GOterms = new SqlCommand(sql_J_GOterms, db_connect);
        into_J_GOterms.Parameters.AddWithValue("@Protein_GoTermID",
Protein_GoTermID);
        into_J_GOterms.Parameters.AddWithValue("@proteinID", proteinID);
        into_J_GOterms.Parameters.AddWithValue("@GoTermID", GoTermID);
        into_J_GOterms.ExecuteNonQuery();
       }

       if ((pdbCode != "") && (uniprotAccession != ""))
        both_data++;

       Console.Write("\r{0} records processed", proteinID);

       proteinID++;

       pdbCode = "";
       uniprotAccession = "";

       field_name = null;
       field_data = null;
      }
      else
      {
```

```
// Check line if is field name or data
if (line.Substring(0, 1) == ">")
 field_name = line.Substring(1);
else
{
 field_data = line;

 // Check field_data if has value
 if (field_data == "NA")
  field_data = "";

 // Create record for protein table
 if (field_name == "proteinName")
  proteinName = field_data;
 else if (field_name == "geneName")
  geneName = field_data;
 else if (field_name == "taxon")
  taxon = field_data;
 else if (field_name == "function")
  proteinFunction = field_data;
 else if (field_name == "proteinFamily")
  proteinFamily = field_data;
 else if (field_name.Substring(0, 3) == "seq")
  seq = field_data;
 else if (field_name == "subLoc")
  subcellularLocation = field_data;

 // Create record for organism
 else if (field_name == "organism")
  organismName = field_data;
 else if (field_name == "organismFeatures")
  organismFeatures = field_data;

 // Create record for PDB_db
 else if (field_name == "pdbCode")
  pdbCode = field_data;
 else if (field_name == "pdbChain")
  pdbChain = field_data;
 else if (field_name == "expMethod")
  expMethod = field_data;
 else if (field_name == "resolution")
  if (field_data != "")
   resolution = float.Parse(field_data.Replace('.', ','));
  else
   resolution = 0;
 else if (field_name == "pdbLink")
  pdbLink = field_data;

 // Create record for UNIPROT_db
 else if (field_name == "uniprotAccession")
  uniprotAccession = field_data;
 else if (field_name == "uniprotEntryName")
```

```
         uniprotEntryName = field_data;
        else if (field_name == "uniprotLink")
         uniprotLink = field_data;

        // Create record for keywords
        else if (field_name == "keywords")
        {
         if (field_data != "")
         {
          recordkeywords = new keywords();
          recordkeywords.keyword = field_data;
          keywordsQueue.Enqueue(recordkeywords);
         }
        }

        // Create record for variants
        else if (field_name == "variant")
        {
         if (field_data != "")
         {
          recordvariants = new variants();
          recordvariants.variant = field_data;
          variantsQueue.Enqueue(recordvariants);
         }
        }

       // Create record for mutagens
        else if (field_name == "mutagen")
        {
         if (field_data != "")
         {
          recordmutagens = new mutagens();
          recordmutagens.mutagen = field_data;
          mutagensQueue.Enqueue(recordmutagens);
         }
        }

        else if (field_name == "DSSP")
        {
         if (field_data != "")
         {
          first_tab = field_data.IndexOf('\t');
          second_tab = field_data.LastIndexOf('\t');
          recordDSSP = new DSSP();
          recordDSSP.structs = field_data.Substring(first_tab + 1, second_tab -
first_tab - 1);
          recordDSSP.position = field_data.Substring(second_tab + 1);
          DSSPQueue.Enqueue(recordDSSP);
         }
        }

        else if (field_name == "pdbEntries")
```

```
        {
         if (field_data != "")
         {
          recordPDBentries = new PDBentries();
          recordPDBentries.pdbEntry = field_data;
          PDBentriesQueue.Enqueue(recordPDBentries);
         }
        }

        else if (field_name == "GO_terms")
        {
         if (field_data != "")
         {
          first_tab = field_data.IndexOf('\t');
          second_tab = field_data.LastIndexOf('\t');
          recordGOterms = new GO_terms();
          recordGOterms.accession = field_data.Substring(0, first_tab);
          recordGOterms.term = field_data.Substring(first_tab + 1, second_tab -
first_tab - 1);
          recordGOterms.ontology = field_data.Substring(second_tab + 1);
          GOtermsQueue.Enqueue(recordGOterms);
         }
        }
       }
      }
     }
    }
   }
   catch (Exception err_txt)
   {
    // Let user know if anything go wrong with txt
    Console.WriteLine("The file could not be read:");
    Console.WriteLine(err_txt.Message);
   }

   // Close the database connection
   try
   {
    db_connect.Close();
   }
   // Let user know if anything go wrong with close db
   catch (Exception err_closedb)
   {
    Console.WriteLine(err_closedb.ToString());
   }

   DateTime StopTime = DateTime.Now;
   TimeSpan duration = StopTime - StartTime;

   Console.WriteLine("\n\n{0} proteins with only PDB data inserted", PDB_data);
   Console.WriteLine("{0} proteins with only UNIPROT data inserted",
UNIPROT_data);
```

```
   Console.WriteLine("{0} proteins with PDB and UNIPROT data inserted",
both_data);

   Console.WriteLine("Total time of process: {0} hours {1} minutes {2} seconds
{3} ms",
    + duration.Hours, duration.Minutes, duration.Seconds, duration.Milliseconds);

   Console.WriteLine("Average time for one record process: {0} ms",
    +duration.TotalMilliseconds/proteinID);
  }
 }
}
```

Εισαγωγή πρόσθετων δεδομένων

```
using System;
using System.IO;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using System.Collections;
using System.Collections.Generic;
using System.Threading;

namespace RGD_more_dataDB
{

 public class dataDB
 {
  public string db_identifier;
  public string db_name;
 }

 public static class RGD_insert_more_dataDB
 {
  public static void Main()
  {

   string connectionString = "Data Source=YHATZIS-STATION;"
    + "Initial Catalog=RGD;"
    + "Integrated Security=True";

   SqlConnection db_connect = null;

   Console.WriteLine("Give filename. Filename must the same with TableDB name");
   string filename = Console.In.ReadLine();

   String path = @".\more_data\" + filename +".csv";

   String line;
```

```
int lines = 0;
int file_false = 0;
String uniprotAccession;
int pos = 0;

// Queue for Table
dataDB record_dataDB = null;
record_dataDB = new dataDB();
Queue dataDB_Queue = new Queue();

// Table IDs
int table_ID = 0;
int uniprot_ID = 0;

// SQL command for inserting data into Table
String sql_find_UniprotDbID = "SELECT ID FROM dbo.UniprotDb "
 + "WHERE Accession = @Accession ";

String sql_insert = "INSERT INTO dbo." + filename + " VALUES ("
 + "@ID, "
 + "@UniprotDbID, "
 + "@identifier, "
 + "@name)";

// Create and open the database connection
try
{
 db_connect = new SqlConnection(connectionString);
 db_connect.Open();
}
catch (Exception err_opendb)
{
 // Let user know if anything go wrong with open db
 if (db_connect != null)
 {
  Console.WriteLine(err_opendb.Message);
  db_connect.Dispose();
 }
}

try
{
 // Create an instance of StreamReader to read from a file.
 // The using statement also closes the StreamReader.
 using (StreamReader datafile = new StreamReader(path))
 {
  // Check file for final ';'.
  while ((line = datafile.ReadLine()) != null)
  {
   lines++;
   if (line.Substring(line.Length - 1, 1) == ";")
   {
```

```
       file_false = 1;
       Console.WriteLine("Problem! Line {0} has ';' at the end of line", lines);
      }
      int commas = 0;
      foreach (char c in line)
      {
       if (c == ';')
        commas++;
      }
      if (commas > 1 && commas % 2 != 0)
      {
       file_false = 1;
       Console.WriteLine("Problem! In line {0} missing identifier or name",
lines);
      }
     }

     if (file_false == 0)
     {
      Console.WriteLine("File is OK! Begin inserting data to DB");
      datafile.DiscardBufferedData();
      datafile.BaseStream.Seek(0, SeekOrigin.Begin);
      datafile.BaseStream.Position = 0;
     }

     // Read lines from the file until the end of the file is reached.
     while ((line = datafile.ReadLine()) != null)
     {
      pos = line.IndexOf(';');
      uniprotAccession = line.Substring(0, pos);
      line = line.Substring(pos + 1);

      SqlCommand find_UniprotDbID = new SqlCommand(sql_find_UniprotDbID,
db_connect);
      find_UniprotDbID.Parameters.AddWithValue("@Accession", uniprotAccession);
      object found_UniprotDbID = find_UniprotDbID.ExecuteScalar();

      if (found_UniprotDbID == null)
       Console.WriteLine("Problem! Protein with Accession {0} not found. EC
values not inserted.", uniprotAccession);
      else
       uniprot_ID = Convert.ToInt32(found_UniprotDbID);

      if (line == "NA")
       Console.WriteLine("{0} | {1} | EMPTY - Not inserted", uniprot_ID,
uniprotAccession);
      else
      {
       do
       {
        record_dataDB = new dataDB();
```

```
        pos = line.IndexOf(';');
        record_dataDB.db_identifier = line.Substring(0, pos);
        line = line.Substring(pos + 1);
        pos = line.IndexOf(';');
        if (pos == -1)
         record_dataDB.db_name = line;
        else
        {
         record_dataDB.db_name = line.Substring(0, pos);
         line = line.Substring(pos + 1);
        }
        dataDB_Queue.Enqueue(record_dataDB);
       }
       while (pos != -1);

       while (dataDB_Queue.Count != 0)
       {
        record_dataDB = (dataDB)dataDB_Queue.Dequeue();

        table_ID++;

        Console.WriteLine("OK >>> {0} | {1} | {2} | {3}", table_ID, uniprot_ID,
record_dataDB.db_identifier, record_dataDB.db_name);

        SqlCommand into_table = new SqlCommand(sql_insert, db_connect);
        into_table.Parameters.AddWithValue("@ID", table_ID);
        into_table.Parameters.AddWithValue("@UniprotDbID", uniprot_ID);
        into_table.Parameters.AddWithValue("@identifier",
record_dataDB.db_identifier);
        into_table.Parameters.AddWithValue("@name", record_dataDB.db_name);
        into_table.ExecuteNonQuery();
       }
      }
     }
    }
   }
   catch (Exception err_txt)
   {
    // Let user know if anything go wrong with txt
    Console.WriteLine("The file could not be read:");
    Console.WriteLine(err_txt.Message);
   }

   // Close the database connection
   try
   {
    db_connect.Close();
   }
   // Let user know if anything go wrong with close db
   catch (Exception err_closedb)
   {
    Console.WriteLine(err_closedb.ToString());
```

```
    }
   }
  }
 }
```

Εισαγωγή στοιχείου ενζυμικής ταξινόμησης (EC)

```
using System;
using System.IO;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using System.Collections;
using System.Collections.Generic;
using System.Threading;

namespace RGD_EC_data
{

 public class EC
 {
  public string ec_value;
 }

 public static class RGD_insert_EC_data
 {
  public static void Main()
  {

   string connectionString = "Data Source=YHATZIS-STATION;"
    + "Initial Catalog=RGD;"
    + "Integrated Security=True";

   SqlConnection db_connect = null;

   String path = @".\more_data\EC.csv";

   String line;
   int lines = 0;
   int file_false = 0;
   String uniprotAccession;
   int pos = 0;

   // Queue for Table EC
   EC recordEC = null;
   recordEC = new EC();
   Queue ECQueue = new Queue();

   // table ID
   int ECID = 0;
```

```
int uniprot_ID = 0;

// SQL command for inserting data into Table
String sql_find_UniprotDbID = "SELECT ID FROM dbo.UniprotDb "
 + "WHERE Accession = @Accession ";

String sql_EC = "INSERT INTO dbo.EC VALUES ("
 + "@ECID, "
 + "@UniprotDbID, "
 + "@ECvalue)";

// Create and open the database connection
try
{
 db_connect = new SqlConnection(connectionString);
 db_connect.Open();
}
catch (Exception err_opendb)
{
 // Let user know if anything go wrong with open db
 if (db_connect != null)
 {
  Console.WriteLine(err_opendb.Message);
  db_connect.Dispose();
 }
}

try
{
 // Create an instance of StreamReader to read from a file.
 // The using statement also closes the StreamReader.
 using (StreamReader datafile = new StreamReader(path))
 {
  // Check file for final ';'.
  while ((line = datafile.ReadLine()) != null)
  {
   lines++;
   if (line.Substring(line.Length - 1, 1) == ";")
   {
    file_false = 1;
    Console.WriteLine("Problem! Line {0} has ';' at the end of line", lines);
   }
  }

  if (file_false == 0)
  {
   Console.WriteLine("File is OK! Begin inserting data to DB");
   datafile.DiscardBufferedData();
   datafile.BaseStream.Seek(0, SeekOrigin.Begin);
   datafile.BaseStream.Position = 0;
  }
```

```
     // Read lines from the file until the end of the file is reached.
     while ((line = datafile.ReadLine()) != null)
     {
      pos = line.IndexOf(';');
      uniprotAccession = line.Substring(0, pos);
      line = line.Substring(pos + 1);

      SqlCommand find_UniprotDbID = new SqlCommand(sql_find_UniprotDbID,
db_connect);
      find_UniprotDbID.Parameters.AddWithValue("@Accession", uniprotAccession);
      object found_UniprotDbID = find_UniprotDbID.ExecuteScalar();

      if (found_UniprotDbID == null)
       Console.WriteLine("Problem! Protein with Accession {0} not found. EC
values not inserted.", uniprotAccession);
      else
       uniprot_ID = Convert.ToInt32(found_UniprotDbID);

      do
      {
       recordEC = new EC();

       pos = line.IndexOf(';');

       if (pos == -1)
       {
        recordEC.ec_value = line;
        if (recordEC.ec_value == "NA")
         recordEC.ec_value = "";
       }
       else
       {
        recordEC.ec_value = line.Substring(0, pos);
        line = line.Substring(pos + 1);
       }

       ECQueue.Enqueue(recordEC);
      }
      while (pos != -1);

      while (ECQueue.Count != 0)
      {
       recordEC = (EC)ECQueue.Dequeue();

       if (recordEC.ec_value != "")
       {
        ECID++;

        Console.WriteLine("OK >>> {0} | {1} | {2}", ECID, uniprot_ID,
recordEC.ec_value);

        SqlCommand into_EC = new SqlCommand(sql_EC, db_connect);
```

```
        into_EC.Parameters.AddWithValue("@ECID", ECID);
        into_EC.Parameters.AddWithValue("@UniprotDbID", uniprot_ID);
        into_EC.Parameters.AddWithValue("@ECvalue", recordEC.ec_value);
        into_EC.ExecuteNonQuery();
       }
       else
        Console.WriteLine("{0} | {1} | EMPTY - Not inserted", uniprot_ID,
uniprotAccession);
      }
     }
    }
   }
   catch (Exception err_txt)
   {
    // Let user know if anything go wrong with txt
    Console.WriteLine("The file could not be read:");
    Console.WriteLine(err_txt.Message);
   }

   // Close the database connection
   try
   {
    db_connect.Close();
   }
   // Let user know if anything go wrong with close db
   catch (Exception err_closedb)
   {
    Console.WriteLine(err_closedb.ToString());
   }
  }
 }
}
```

Ενημέρωση πεδίου phylumKingdom

```
using System;
using System.IO;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using System.Collections;
using System.Collections.Generic;
using System.Threading;

namespace RGD_db_more_data
{
 public static class RGD_insert_more_data
 {
  public static void Main()
  {
```

```
string connectionString = "Data Source=YHATZIS-STATION;"
 + "Initial Catalog=RGD;"
 + "Integrated Security=True";

SqlConnection db_connect = null;

String path = @".\more_data\phylumKingdom.csv";

String line;
int pos_comma = 0;
int counter = 0;

String uniprotAccession = "";
String phylumKingdom = "";

// SQL commands for inserting data into Tables

String sql_UNIPROT_db = "UPDATE dbo.UniprotDb "
 + "SET PhylumKingdom = @phylumKingdom "
 + "WHERE Accession=@uniprotAccession";

// Create and open the database connection
try
{
 db_connect = new SqlConnection(connectionString);
 db_connect.Open();
}
catch (Exception err_opendb)
{
 // Let user know if anything go wrong with open db
 if (db_connect != null)
 {
  Console.WriteLine(err_opendb.Message);
  db_connect.Dispose();
 }
}

try
{
 // Create an instance of StreamReader to read from a file.
 // The using statement also closes the StreamReader.
 using (StreamReader datafile = new StreamReader(path))
 {
  // Read lines from the file until the end of the file is reached.
  while ((line = datafile.ReadLine()) != null)
  {
   counter++;
   pos_comma = line.IndexOf(';');

   uniprotAccession = line.Substring(0, pos_comma);
   phylumKingdom = line.Substring(pos_comma + 1);
```

```
      //Console.WriteLine("{0} - {1}", uniprotAccession, phylumKingdom);

      Console.Write("\r{0} records updated", counter);

      SqlCommand set_phylumKingdom = new SqlCommand(sql_UNIPROT_db, db_connect);
      set_phylumKingdom.Parameters.AddWithValue("@phylumKingdom", phylumKingdom);
      set_phylumKingdom.Parameters.AddWithValue("@uniprotAccession",
uniprotAccession);
      set_phylumKingdom.ExecuteNonQuery();
     }
    }
   }

   catch (Exception err_txt)
   {
    // Let user know if anything go wrong with txt
    Console.WriteLine("The file could not be read:");
    Console.WriteLine(err_txt.Message);
   }

   // Close the database connection
   try
   {
    db_connect.Close();
   }

   // Let user know if anything go wrong with close db
   catch (Exception err_closedb)
   {
    Console.WriteLine(err_closedb.ToString());
   }

   Console.WriteLine("\n\nAll UNIPROTdb records updated");
  }
 }
}
```

Ενημέρωση των εγγραφών με τα PDB files

```
using System;
using System.IO;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using System.Collections;
using System.Collections.Generic;
using System.Threading;
using System.Net;

namespace RGD_pdb_files
```

```
{
 public static class RGD_DL_pdb_files
 {
  public static void Main()
  {
   string connectionString = "Data Source=YHATZIS-STATION;"
    + "Initial Catalog=RGD;"
    + "Integrated Security=True";

   SqlConnection db_connect = null;

   // Variables
   int PdbDbID = 0;
   int pbd_records = 0;
                        String pdbCode = "";
                        String remote_file = "";
   String local_file = "";
   String remote_image = "";
   String local_image = "";

                        String sql_count_records = "SELECT count(*) FROM
dbo.PdbDb";

   String sql_find_pdbCode = "SELECT pdbCode FROM dbo.PdbDb "
                                + "WHERE ID = @PdbDbID";

   try
   {
    db_connect = new SqlConnection(connectionString);
    db_connect.Open();
   }
   catch (Exception err_opendb)
   {
    // Let user know if anything go wrong with open db
    if (db_connect != null)
    {
     Console.WriteLine(err_opendb.Message);
     db_connect.Dispose();
    }
   }

   SqlCommand count_records = new SqlCommand(sql_count_records, db_connect);
   object records = count_records.ExecuteScalar();
   pbd_records = Convert.ToInt32(records);

   for (PdbDbID = 1; PdbDbID <= pbd_records; PdbDbID++)
                        {
                                WebClient webClient = new WebClient();

    SqlCommand find_pdbCode = new SqlCommand(sql_find_pdbCode, db_connect);
                                find_pdbCode.Parameters.AddWithValue("@PdbDbID",
PdbDbID);
```

```
                                object found_pdbCode =
find_pdbCode.ExecuteScalar();
    pdbCode = Convert.ToString(found_pdbCode);

    remote_file = "http://www.rcsb.org/pdb/files/" + pdbCode + ".pdb.gz";
    local_file = @"F:\Διδακτορικό\RGD-database\pdb_files\" + pdbCode + ".pdb.gz";

    remote_image = "http://www.rcsb.org/pdb/images/" + pdbCode +
"_bio_r_500.jpg";
    local_image = @"F:\Διδακτορικό\RGD-database\pdb_images\" + pdbCode + ".jpg";

    Console.WriteLine("{0}. Downloading files for {1} protein", PdbDbID,
pdbCode);

    Console.WriteLine("Downloading pdb file...");
    webClient.DownloadFile(remote_file, local_file);

    Console.WriteLine("Downloading image file...");
    webClient.DownloadFile(remote_image, local_image);

    FileInfo f_info = new FileInfo(local_file);
    long f_size = f_info.Length;
    FileInfo img_info = new FileInfo(local_image);
    long img_size = img_info.Length;

    Console.WriteLine("Downloaded pdb file {0}KB and image {1}KB\n / / / / / / /
/ / / / / / /", f_size/1024, img_size/1024);

                        }
                }
 }
}
```

Παράρτημα 2

Λογισμικό οπτικοποίησης δεδομένων fungibase

Σχεδίαση ραβδογράμματος

```
<?php
session_start();
        include_once 'config.php';
        $con=mysqli_connect(DB_HOST,DB_USER,DB_PASS,DB_NAME);
        mysqli_set_charset($con,"utf8");

        if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_error();
        }

        if (isset($_SESSION['userSession'])) {
                $query = $con->query("SELECT * FROM users WHERE
id=".$_SESSION['userSession']);
                $userRow = $query->fetch_array();
                $user = $_SESSION['userSession'];
                $role_id = $userRow['role_id'];
        } else {
                $user = 0;
                $role_id = 0;
        }
?>
<!DOCTYPE html>
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>fungibase | The science database of things Fungal</title>
        <link rel="shortcut icon" href="images/favicon.ico">
        <link rel="icon" type="image/png" href="images/favicon.png">
        <link rel="stylesheet" type="text/css" href="css/style.css">
```

```
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <script src="js/jquery.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <link
href="https://fonts.googleapis.com/css?family=Ubuntu&amp;subset=greek,greek-ext"
rel="stylesheet">
</head>

<body>

<div class="container-fluid" style="max-width:75%">
<div style="width:200px;margin:0 auto"><div style="float:left;margin-
top:7px"><img src="images/logo_s.png"></div></div>
<?php
        include 'menu.php';
        include 'visuals/bar_tsv.php';
?>

<style>

.bar {
  fill: steelblue;
}

.bar:hover {
  fill: brown;
}

.axis--x path {
  display: none;
}

</style>
<svg width="960" height="500"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>

var svg = d3.select("svg"),
    margin = {top: 20, right: 20, bottom: 30, left: 40},
    width = +svg.attr("width") - margin.left - margin.right,
    height = +svg.attr("height") - margin.top - margin.bottom;

var x = d3.scaleBand().rangeRound([0, width]).padding(0.1),
    y = d3.scaleLinear().rangeRound([height, 0]);

var g = svg.append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

d3.tsv("visuals/bar.tsv", function(d) {
  d.num = +d.num;
  return d;
}, function(error, data) {
```

```
  if (error) throw error;

  x.domain(data.map(function(d) { return d.method; }));
  y.domain([0, d3.max(data, function(d) { return d.num; })]);

  g.append("g")
      .attr("class", "axis axis--x")
      .attr("transform", "translate(0," + height + ")")
      .call(d3.axisBottom(x));

  g.append("g")
      .attr("class", "axis axis--y")
      .call(d3.axisLeft(y).ticks(20, "d"))
    .append("text")
      .attr("transform", "rotate(-90)")
      .attr("y", 6)
      .attr("dy", "0.71em")
      .attr("text-anchor", "end")
      .text("Total Records");

  g.selectAll(".bar")
    .data(data)
    .enter().append("rect")
      .attr("class", "bar")
      .attr("x", function(d) { return x(d.method); })
      .attr("y", function(d) { return y(d.num); })
      .attr("width", x.bandwidth())
      .attr("height", function(d) { return height - y(d.num); });
});

</script>
```

Δημιουργία αρχείου δεδομένων για ραβδόγραμμα

```
<?php
        $myfile = fopen("visuals/bar.tsv", "w") or die("Unable to open file!");
        fwrite($myfile, "method\tnum\n");

        $full_record = mysqli_query( $con,"SELECT name_method AS name,
count(id_fungus) AS num FROM fung_rec AS fr
                                                                INNER JOIN
records AS r on fr.id_record = r.id_record INNER JOIN methods AS md on
r.id_method = md.id_method GROUP BY md.id_method" );

        while($row = mysqli_fetch_array($full_record))
                fwrite($myfile, $row['name'] . "\t" . $row['num'] . "\n");

        fclose($myfile);
?>
```

Σχεδίαση πίτας

```
<?php
session_start();
        include_once 'config.php';
        $con=mysqli_connect(DB_HOST,DB_USER,DB_PASS,DB_NAME);
        mysqli_set_charset($con,"utf8");

        if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_error();
        }

        if (isset($_SESSION['userSession'])) {
                $query = $con->query("SELECT * FROM users WHERE
id=".$_SESSION['userSession']);
                $userRow = $query->fetch_array();
                $user = $_SESSION['userSession'];
                $role_id = $userRow['role_id'];
        } else {
                $user = 0;
                $role_id = 0;
        }
?>
<!DOCTYPE html>
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>fungibase | The science database of things Fungal</title>
        <link rel="shortcut icon" href="images/favicon.ico">
        <link rel="icon" type="image/png" href="images/favicon.png">
        <link rel="stylesheet" type="text/css" href="css/style.css">
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <script src="js/jquery.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <link
href="https://fonts.googleapis.com/css?family=Ubuntu&amp;subset=greek,greek-ext"
rel="stylesheet">
</head>

<body>

<div class="container-fluid" style="max-width:75%">
<div style="width:200px;margin:0 auto"><div style="float:left;margin-
top:7px"><img src="images/logo_s.png"></div></div>
<?php
        include 'menu.php';
        include 'visuals/pie_csv.php';
?>

<style>
```

```
.arc text {
  font: 14px sans-serif;
  text-anchor: middle;
}

.arc path {
  stroke: #fff;
}

</style>
<svg width="960" height="500"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>

var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height"),
    radius = Math.min(width, height) / 2,
    g = svg.append("g").attr("transform", "translate(" + width / 2 + "," + height
/ 2 + ")");

var color = d3.scaleOrdinal(["#98abc5", "#8a89a6", "#7b6888", "#6b486b",
"#a05d56", "#d0743c", "#ff8c00"]);

var pie = d3.pie()
    .sort(null)
    .value(function(d) { return d.num; });

var path = d3.arc()
    .outerRadius(radius - 10)
    .innerRadius(0);

var label = d3.arc()
    .outerRadius(radius - 40)
    .innerRadius(radius - 40);

d3.csv("visuals/pie.csv", function(d) {
  d.num = +d.num;
  return d;
}, function(error, data) {
  if (error) throw error;

  var arc = g.selectAll(".arc")
    .data(pie(data))
    .enter().append("g")
      .attr("class", "arc");

  arc.append("path")
      .attr("d", path)
      .attr("fill", function(d) { return color(d.data.method); });

  arc.append("text")
```

```
      .attr("transform", function(d) { return "translate(" + label.centroid(d) +
")"; })
      .attr("dy", "0.35em")
      .text(function(d) { return d.data.method; });
});

</script>
```

Δημιουργία αρχείου δεδομένων πίτας

```
<?php
        $myfile = fopen("visuals/pie.csv", "w") or die("Unable to open file!");
        fwrite($myfile, "method,num\n");

        $full_record = mysqli_query( $con,"SELECT name_method AS name,
count(id_fungus) AS num FROM fung_rec AS fr
                                                                INNER JOIN
records AS r on fr.id_record = r.id_record INNER JOIN methods AS md on
r.id_method = md.id_method GROUP BY md.id_method" );

        while($row = mysqli_fetch_array($full_record))
                fwrite($myfile, $row['name'] . "," . $row['num'] . "\n");

        fclose($myfile);
?>
```

Σχεδίαση διαγράμματος δέντρου

```
<?php
session_start();
        include_once 'config.php';
        $con=mysqli_connect(DB_HOST,DB_USER,DB_PASS,DB_NAME);
        mysqli_set_charset($con,"utf8");

        if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_error();
        }

        if (isset($_SESSION['userSession'])) {
                $query = $con->query("SELECT * FROM users WHERE
id=".$_SESSION['userSession']);
                $userRow = $query->fetch_array();
                $user = $_SESSION['userSession'];
                $role_id = $userRow['role_id'];
        } else {
                $user = 0;
                $role_id = 0;
        }
```

```
?>
<!DOCTYPE html>
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>fungibase | The science database of things Fungal</title>
        <link rel="shortcut icon" href="images/favicon.ico">
        <link rel="icon" type="image/png" href="images/favicon.png">
        <link rel="stylesheet" type="text/css" href="css/style.css">
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <script src="js/jquery.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <link
href="https://fonts.googleapis.com/css?family=Ubuntu&amp;subset=greek,greek-ext"
rel="stylesheet">
</head>

<body>

<div class="container-fluid" style="max-width:75%">
<div style="width:200px;margin:0 auto"><div style="float:left;margin-
top:7px"><img src="images/logo_s.png"></div></div>
<?php
        include 'menu.php';
        include 'visuals/flare_csv.php';
?>

<style>

.node circle {
  fill: #999;
}

.node text {
  font: 14px sans-serif;
}

.node--internal circle {
  fill: #555;
}

.node--internal text {
  text-shadow: 0 1px 0 #fff, 0 -1px 0 #fff, 1px 0 0 #fff, -1px 0 0 #fff;
}

.link {
  fill: none;
  stroke: #555;
  stroke-opacity: 0.4;
  stroke-width: 1.5px;
}
```

```
</style>
<svg width="960" height="2000"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>

var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height"),
    g = svg.append("g").attr("transform", "translate(100,0)");

var tree = d3.tree()
    .size([height, width - 300]);

var stratify = d3.stratify()
    .parentId(function(d) { return d.id.substring(0, d.id.lastIndexOf(".")); });

d3.csv("visuals/flare.csv", function(error, data) {
  if (error) throw error;

  var root = stratify(data)
      .sort(function(a, b) { return (a.height - b.height) ||
a.id.localeCompare(b.id); });

  var link = g.selectAll(".link")
    .data(tree(root).links())
    .enter().append("path")
      .attr("class", "link")
      .attr("d", d3.linkHorizontal()
          .x(function(d) { return d.y; })
          .y(function(d) { return d.x; }));

  var node = g.selectAll(".node")
    .data(root.descendants())
    .enter().append("g")
      .attr("class", function(d) { return "node" + (d.children ? " node--
internal" : " node--leaf"); })
      .attr("transform", function(d) { return "translate(" + d.y + "," + d.x +
")"; })

  node.append("circle")
      .attr("r", 2.5);

  node.append("text")
      .attr("dy", 3)
      .attr("x", function(d) { return d.children ? -8 : 8; })
      .style("text-anchor", function(d) { return d.children ? "end" : "start"; })
      .text(function(d) { return d.id.substring(d.id.lastIndexOf(".") + 1); });
});

</script>
```

Σχεδίαση διαγράμματος κυκλικού δέντρου

```
<?php
session_start();
        include_once 'config.php';
        $con=mysqli_connect(DB_HOST,DB_USER,DB_PASS,DB_NAME);
        mysqli_set_charset($con,"utf8");

        if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_error();
        }

        if (isset($_SESSION['userSession'])) {
                $query = $con->query("SELECT * FROM users WHERE
id=".$_SESSION['userSession']);
                $userRow = $query->fetch_array();
                $user = $_SESSION['userSession'];
                $role_id = $userRow['role_id'];
        } else {
                $user = 0;
                $role_id = 0;
        }
?>
<!DOCTYPE html>
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>fungibase | The science database of things Fungal</title>
        <link rel="shortcut icon" href="images/favicon.ico">
        <link rel="icon" type="image/png" href="images/favicon.png">
        <link rel="stylesheet" type="text/css" href="css/style.css">
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <script src="js/jquery.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <link
href="https://fonts.googleapis.com/css?family=Ubuntu&amp;subset=greek,greek-ext"
rel="stylesheet">
</head>

<body>

<div class="container-fluid" style="max-width:75%">
<div style="width:200px;margin:0 auto"><div style="float:left;margin-
top:7px"><img src="images/logo_s.png"></div></div>
<?php
        include 'menu.php';
        include 'visuals/flare_csv.php';
?>

<style>
```

```
.node circle {
  fill: #999;
}

.node text {
  font: 14px sans-serif;
}

.node--internal circle {
  fill: #555;
}

.node--internal text {
  text-shadow: 0 1px 0 #fff, 0 -1px 0 #fff, 1px 0 0 #fff, -1px 0 0 #fff;
}

.link {
  fill: none;
  stroke: #555;
  stroke-opacity: 0.4;
  stroke-width: 1.5px;
}

</style>
<svg width="960" height="1060"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>

var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height"),
    g = svg.append("g").attr("transform", "translate(" + (width / 2 + 40) + "," +
(height / 2 + 10) + ")");

var stratify = d3.stratify()
    .parentId(function(d) { return d.id.substring(0, d.id.lastIndexOf(".")); });

var tree = d3.tree()
    .size([2 * Math.PI, 250])
    .separation(function(a, b) { return (a.parent == b.parent ? 1 : 2) / a.depth;
});

d3.csv("visuals/flare.csv", function(error, data) {
  if (error) throw error;

  var root = tree(stratify(data));

  var link = g.selectAll(".link")
    .data(root.links())
    .enter().append("path")
      .attr("class", "link")
```

```
      .attr("d", d3.linkRadial()
          .angle(function(d) { return d.x; })
          .radius(function(d) { return d.y; }));

  var node = g.selectAll(".node")
    .data(root.descendants())
    .enter().append("g")
      .attr("class", function(d) { return "node" + (d.children ? " node--
internal" : " node--leaf"); })
      .attr("transform", function(d) { return "translate(" + radialPoint(d.x,
d.y) + ")"; });

  node.append("circle")
      .attr("r", 2.5);

  node.append("text")
      .attr("dy", "0.31em")
      .attr("x", function(d) { return d.x < Math.PI === !d.children ? 6 : -6; })
      .attr("text-anchor", function(d) { return d.x < Math.PI === !d.children ?
"start" : "end"; })
      .attr("transform", function(d) { return "rotate(" + (d.x < Math.PI ? d.x -
Math.PI / 2 : d.x + Math.PI / 2) * 180 / Math.PI + ")"; })
      .text(function(d) { return d.id.substring(d.id.lastIndexOf(".") + 1); });
});

function radialPoint(x, y) {
  return [(y = +y) * Math.cos(x -= Math.PI / 2), y * Math.sin(x)];
}

</script>
```

Σχεδίαση διαγράμματος φυσσαλίδων

```
<?php
session_start();
        include_once 'config.php';
        $con=mysqli_connect(DB_HOST,DB_USER,DB_PASS,DB_NAME);
        mysqli_set_charset($con,"utf8");

        if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_error();
        }

        if (isset($_SESSION['userSession'])) {
                $query = $con->query("SELECT * FROM users WHERE
id=".$_SESSION['userSession']);
                $userRow = $query->fetch_array();
                $user = $_SESSION['userSession'];
                $role_id = $userRow['role_id'];
```

```
        } else {
                $user = 0;
                $role_id = 0;
        }
?>
<!DOCTYPE html>
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>fungibase | The science database of things Fungal</title>
        <link rel="shortcut icon" href="images/favicon.ico">
        <link rel="icon" type="image/png" href="images/favicon.png">
        <link rel="stylesheet" type="text/css" href="css/style.css">
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <script src="js/jquery.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <link
href="https://fonts.googleapis.com/css?family=Ubuntu&amp;subset=greek,greek-ext"
rel="stylesheet">
</head>

<body>

<div class="container-fluid" style="max-width:75%">
<div style="width:200px;margin:0 auto"><div style="float:left;margin-
top:7px"><img src="images/logo_s.png"></div></div>
<?php
        include 'menu.php';
        include 'visuals/flare_csv.php';
?>

<svg width="960" height="960" font-family="sans-serif" font-size="13" text-
anchor="middle"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>

var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height");

var format = d3.format(",d");

var color = d3.scaleOrdinal(d3.schemeCategory20c);

var pack = d3.pack()
    .size([width, height])
    .padding(1.5);

d3.csv("visuals/flare.csv", function(d) {
  d.value = +d.value;
  if (d.value) return d;
```

```
}, function(error, classes) {
  if (error) throw error;

  var root = d3.hierarchy({children: classes})
      .sum(function(d) { return d.value; })
      .each(function(d) {
        if (id = d.data.id) {
          var id, i = id.lastIndexOf(".");
          d.id = id;
          d.package = id.slice(0, i);
          d.class = id.slice(i + 1);
        }
      });

  var node = svg.selectAll(".node")
    .data(pack(root).leaves())
    .enter().append("g")
      .attr("class", "node")
      .attr("transform", function(d) { return "translate(" + d.x + "," + d.y +
")"; });

  node.append("circle")
      .attr("id", function(d) { return d.id; })
      .attr("r", function(d) { return d.r; })
      .style("fill", function(d) { return color(d.package); });

  node.append("clipPath")
      .attr("id", function(d) { return "clip-" + d.id; })
    .append("use")
      .attr("xlink:href", function(d) { return "#" + d.id; });

  node.append("text")
      .attr("clip-path", function(d) { return "url(#clip-" + d.id + ")"; })
    .selectAll("tspan")
    .data(function(d) { return d.class.split(/(?=[A-Z][^A-Z])/g); })
    .enter().append("tspan")
      .attr("x", 0)
      .attr("y", function(d, i, nodes) { return 13 + (i - nodes.length / 2 - 0.5)
* 10; })
      .text(function(d) { return d; });

  node.append("title")
      .text(function(d) { return d.id + "\n" + format(d.value); });
});

</script>
```

Δημιουργία αρχείου δεδομένων για διαγραμματα δέντρων και φυσσαλίδας

```
<?php
        $myfile = fopen("visuals/flare.csv", "w") or die("Unable to open file!");
        fwrite($myfile, "id,value\n");
        fwrite($myfile, "Methodologies,\n");

        $full_record = mysqli_query( $con,"SELECT methodology_name AS
ml,name_method AS md,f.fungus_name AS fn,count(f.id_fungus) as x
                                        FROM methodologies AS ml INNER JOIN
methods AS md ON ml.id_methodology=md.id_methodology INNER JOIN records AS rec ON
md.id_method=rec.id_method
                                        INNER JOIN fung_rec AS fr ON
rec.id_record=fr.id_record INNER JOIN fungus AS f ON fr.id_fungus=f.id_fungus
                                        GROUP BY
methodology_name,name_method,fungus_name" );

        $lastml = "";
        $lastmd = "";

        while($row = mysqli_fetch_array($full_record)) {
                if ($row['ml'] != $lastml) {
                        fwrite($myfile, "Methodologies." . $row['ml'] . ",\n");
                        fwrite($myfile, "Methodologies." . $row['ml'] . "." .
$row['md'] . ",\n");
                        fwrite($myfile, "Methodologies." . $row['ml'] . "." .
$row['md'] . "." . $row['fn'] . "," . $row['x'] . "\n");
                }
                elseif ($row['md'] != $lastmd) {
                        fwrite($myfile, "Methodologies." . $row['ml'] . "." .
$row['md'] . ",\n");
                        fwrite($myfile, "Methodologies." . $row['ml'] . "." .
$row['md'] . "." . $row['fn'] . "," . $row['x'] . "\n");
                }
                else {
                        fwrite($myfile, "Methodologies." . $row['ml'] . "." .
$row['md'] . "." . $row['fn'] . "," . $row['x'] . "\n");
                }
                $lastml = $row['ml'];
                $lastmd = $row['md'];
        }
        fclose($myfile);
?>
```

Σχεδίαση δενδροδιαγράμματος

```
<?php
session_start();
        include_once 'config.php';
        $con=mysqli_connect(DB_HOST,DB_USER,DB_PASS,DB_NAME);
        mysqli_set_charset($con,"utf8");

        if (mysqli_connect_errno()) {
                echo "Failed to connect to MySQL: " . mysqli_connect_error();
        }

        if (isset($_SESSION['userSession'])) {
                $query = $con->query("SELECT * FROM users WHERE
id=".$_SESSION['userSession']);
                $userRow = $query->fetch_array();
                $user = $_SESSION['userSession'];
                $role_id = $userRow['role_id'];
        } else {
                $user = 0;
                $role_id = 0;
        }
?>
<!DOCTYPE html>
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>fungibase | The science database of things Fungal</title>
        <link rel="shortcut icon" href="images/favicon.ico">
        <link rel="icon" type="image/png" href="images/favicon.png">
        <link rel="stylesheet" type="text/css" href="css/style.css">
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <script src="js/jquery.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
        <link
href="https://fonts.googleapis.com/css?family=Ubuntu&amp;subset=greek,greek-ext"
rel="stylesheet">
</head>

<body>

<div class="container-fluid" style="max-width:75%">
<div style="width:200px;margin:0 auto"><div style="float:left;margin-
top:7px"><img src="images/logo_s.png"></div></div>
<?php
        include 'menu.php';
        include 'visuals/flare_json.php';
?>

<style>
```

```
form {
  font-family: 'Ubuntu', sans-serif;
}

svg {
  font: 15px sans-serif;
}

</style>
<svg width="960" height="570"></svg>
<form>
  <label><input type="radio" name="mode" value="sumBySize" checked> Size</label>
  <label><input type="radio" name="mode" value="sumByCount"> Count</label>
</form>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>

var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height");

var fader = function(color) { return d3.interpolateRgb(color, "#fff")(0.2); },
    color = d3.scaleOrdinal(d3.schemeCategory20.map(fader)),
    format = d3.format(",d");

var treemap = d3.treemap()
    .tile(d3.treemapResquarify)
    .size([width, height])
    .round(true)
    .paddingInner(1);

d3.json("visuals/flare.json", function(error, data) {
  if (error) throw error;

  var root = d3.hierarchy(data)
      .eachBefore(function(d) { d.data.id = (d.parent ? d.parent.data.id + "." :
"") + d.data.name; })
      .sum(sumBySize)
      .sort(function(a, b) { return b.height - a.height || b.value - a.value; });

  treemap(root);

  var cell = svg.selectAll("g")
    .data(root.leaves())
    .enter().append("g")
      .attr("transform", function(d) { return "translate(" + d.x0 + "," + d.y0 +
")"; });

  cell.append("rect")
      .attr("id", function(d) { return d.data.id; })
      .attr("width", function(d) { return d.x1 - d.x0; })
```

```
      .attr("height", function(d) { return d.y1 - d.y0; })
      .attr("fill", function(d) { return color(d.parent.data.id); });

  cell.append("clipPath")
      .attr("id", function(d) { return "clip-" + d.data.id; })
    .append("use")
      .attr("xlink:href", function(d) { return "#" + d.data.id; });

  cell.append("text")
      .attr("clip-path", function(d) { return "url(#clip-" + d.data.id + ")"; })
    .selectAll("tspan")
      .data(function(d) { return d.data.name.split(/(?=[A-Z][^A-Z])/g); })
    .enter().append("tspan")
      .attr("x", 4)
      .attr("y", function(d, i) { return 13 + i * 10; })
      .text(function(d) { return d; });

  cell.append("title")
      .text(function(d) { return d.data.id + "\n" + format(d.value); });

  d3.selectAll("input")
      .data([sumBySize, sumByCount], function(d) { return d ? d.name :
this.value; })
      .on("change", changed);

  var timeout = d3.timeout(function() {
    d3.select("input[value=\"sumByCount\"]")
        .property("checked", true)
        .dispatch("change");
  }, 2000);

  function changed(sum) {
    timeout.stop();

    treemap(root.sum(sum));

    cell.transition()
        .duration(750)
        .attr("transform", function(d) { return "translate(" + d.x0 + "," + d.y0
+ ")"; })
      .select("rect")
        .attr("width", function(d) { return d.x1 - d.x0; })
        .attr("height", function(d) { return d.y1 - d.y0; });
  }
});

function sumByCount(d) {
  return d.children ? 0 : 1;
}

function sumBySize(d) {
  return d.size;
```

```
}

</script>
```

Δημιουργία αρχείου δεδομένων δενδροδιαγράμματος

```
<?php
        $myfile = fopen("visuals/flare.json", "w") or die("Unable to open
file!");

        fwrite($myfile, "{\n");
        fwrite($myfile, " \"name\": \"Methodologies\",\n");
        fwrite($myfile, " \"children\": [\n");

        $full_record = mysqli_query( $con,"SELECT methodology_name AS
ml,name_method AS md,count(f.id_fungus) as x
                                        FROM methodologies AS ml INNER JOIN
methods AS md ON ml.id_methodology=md.id_methodology INNER JOIN records AS rec ON
md.id_method=rec.id_method
                                        INNER JOIN fung_rec AS fr ON
rec.id_record=fr.id_record INNER JOIN fungus AS f ON fr.id_fungus=f.id_fungus
                                        GROUP BY methodology_name,name_method"
);

        $lastml = "";

        while($row = mysqli_fetch_array($full_record)) {
                if ($row['ml'] != $lastml) {
                        if ($lastml != "") {
                                fwrite($myfile, "\n");
                                fwrite($myfile, "   ]\n");
                                fwrite($myfile, "  },\n");
                        }
                        fwrite($myfile, "  {\n");
                        fwrite($myfile, "   \"name\": \"" . $row['ml'] .
"\",\n");
                        fwrite($myfile, "   \"children\": [\n");
                        fwrite($myfile, "    {\"name\": \"" . $row['md'] . "\",
\"size\": " . $row['x'] . "}");
                }
                else {
                        fwrite($myfile, ",\n");
                        fwrite($myfile, "    {\"name\": \"" . $row['md'] . "\",
\"size\": " . $row['x'] . "}");
                }
                $lastml = $row['ml'];
        }

        fwrite($myfile, "\n");
        fwrite($myfile, "   ]\n");
```

```
        fwrite($myfile, "  }");
        fwrite($myfile, "\n");
        fwrite($myfile, " ]\n");
        fwrite($myfile, "}");
        fclose($myfile);
?>
```

Παράρτημα 3

Βιογραφικό σημείωμα

Ο Ιωάννης (Γιάννης) Χατζής είναι μόνιμος εκπαιδευτικός στη δευτεροβάθμια εκπαίδευση από το 2001. Γεννήθηκε το Σεπτέμβριο του 1969, έζησε και μεγάλωσε στο Μεσολόγγι.

Αποφοίτησε από το τμήμα Φυσικής του Πανεπιστημίου Πατρών το 1992 και το 2008 έλαβε το μεταπτυχιακό τίτλο από το Πανεπιστήμιο Αθηνών, με τίτλο "Τεχνολογίες της πληροφορίας και της επικοινωνίας για την εκπαίδευση".

Επιπλέον, σπούδασε προγραμματιστής ηλεκτρονικών υπολογιστών και το 1997 έλαβε επαγγελματικό πτυχίο Δευτεροβάθμιας Τεχνικής Σχολής. Σήμερα, συνεχίζει τις σπουδές του στο Τμήμα Διοίκησης Επιχειρήσεων του ΤΕΙ Δυτικής Ελλάδας στην κατεύθυνση "Διοίκηση Πληροφοριακών Συστημάτων".

Εργάστηκε στο Τεχνολογικό Εκπαιδευτικό Ίδρυμα (ΤΕΙ) Μεσολογγίου, από τον Οκτώβριο του 1997 μέχρι τον Αύγουστο του 2001, ως υπεύθυνος του συστήματος αυτοματοποίησης της Βιβλιοθήκης και του δικτύου υπολογιστών, έχοντας συμμετάσχει στο σχεδιασμό και την υλοποίηση διαφόρων έργων έρευνας και ανάπτυξης. Στη συνέχεια, από τον Σεπτέμβριο του 2008 μέχρι τον Αύγουστο του 2010 εργάστηκε στο Κέντρο Διαχείρισης Δικτύου του ΤΕΙ Μεσολογγίου, έχοντας καθήκοντα διαχείρισης και υποστήριξης των συστημάτων σύγχρονης και ασύγχρονης ηλεκτρονικής μάθησης.

Έχει διδάξει μαθήματα Πληροφορικής ως εργαστηριακός συνεργάτης του ΤΕΙ Μεσολογγίου από το 1996 έως το 2011.

Παράλληλα, από τον Οκτώβριο 1994 μέχρι τον Ιούνιο 2002 και από τον Οκτώβριο 2004 μέχρι τον Φεβρουάριο 2006 έχει διδάξει σε δημόσια Ινστιτούτα Επαγγελματικής Κατάρτισης (ΙΕΚ) στο Μεσολόγγι, το Αγρίνιο και την Άμφισσα, καθώς και σε ένα πλήθος προγραμμάτων Εκπαίδευσης Ενηλίκων.

Από το Νοέμβριο του 2013 μέχρι τον Σεπτέμβριο του 2016 διετέλεσε Διευθυντής του δημόσιου ΙΕΚ Αγρινίου και στη συνέχεια, μέχρι σήμερα, κατέχει τη θέση του Διευθυντή στο δημόσιο ΙΕΚ Μεσολογγίου.

Τα ερευνητικά του ενδιαφέροντα βρίσκονται σε δύο κύριους τομείς. Ο πρώτος αφορά τις βάσεις δεδομένων και τα εργαλεία οπτικοποίησης και ανάλυσης των δεδομένων τους, τις τεχνολογίες του Διαδικτύου και τις εφαρμογές στο διαδίκτυο. Ιδιαίτερα, από τον Οκτώβριο του 2010 το ερευνητικό του ενδιαφέρον έχει επικεντρωθεί σε εφαρμογές Βιοπληροφορικής. Παράλληλα, τον απασχολούν και οι μεθοδολογικές προσεγγίσεις στην εκπαίδευση ενηλίκων, που υπηρετεί τα τελευταία χρόνια.

Τον ελεύθερο χρόνο, ασχολείται με τη γραφιστική και τη δημιουργία ιστοσελίδων.

Περίληψη

Τα τελευταία χρόνια είναι έντονο το φαινόμενο της αύξησης και συσσώρευσης βιολογικών δεδομένων σε διεθνείς βάσεις δεδομένων. Ο μεγάλος αριθμός των καταχωρημένων νουκλεοτιδικών αλληλουχιών (π.χ. ολόκληρα γονιδιώματα, ESTs DNA και mRNA) και πρωτεϊνικών αλληλουχιών, των τρισδιάστατων δομών των βιομακρομορίων, καθώς επίσης και η χαρτογράφηση και αλληλούχηση ολόκληρων γονιδιωμάτων, είχε ως αποτέλεσμα τη συγκέντρωση ενός τεράστιου όγκου ακατέργαστων δεδομένων εκτεταμένης πολυπλοκότητας. Οι ερευνητές για να ανταποκριθούν σ' αυτή την πρόκληση έκριναν απαραίτητη τη χρήση ηλεκτρονικών υπολογιστών για την αποτελεσματική αποθήκευση, οργάνωση, ανάλυση και ερμηνεία αυτής της πληθώρας βιολογικών δεδομένων. Με την ταυτόχρονη έντονη ανάπτυξη της Πληροφορικής, δημιουργήθηκαν οι κατάλληλες προϋποθέσεις για την ανάπτυξη ενός νέου και συνεχώς εξελισσόμενου επιστημονικού πεδίου, της Βιοπληροφορικής.

Η πρώτη εφαρμογή δημιουργήθηκε χρησιμοποιώντας τεχνολογία και εργαλεία που αναπτύχθηκαν και υποστηρίζονται από τη Microsoft. Πρόκειται για την RGDtrip, μια εφαρμογή που σχεδιάστηκε και υλοποιήθηκε, με σκοπό την αποθήκευση, διαχείριση και επεξεργασία δεδομένων πρωτεϊνών που περιέχουν το τριπεπτίδιο RGD. Το ερευνητικό ενδιαφέρον εμφανίστηκε από την πρώτη βιβλιογραφική ανασκόπηση και αξιολόγηση όλων των αλληλουχιών RGD και βρόχων σε υποδοχείς μεταξύ των ειδών το 1998 και την επαλήθευση ότι η εμφάνιση τέτοιων αλληλουχιών σε υποδοχείς θα μπορούσε να σημαίνει ότι αυτές οι αλληλουχίες θα ήταν σε δομές τύπου βρόχου / βρόχου και ότι αυτές οι δομές θηλειάς θα μπορούσαν να συνεπάγονται επίσης λειτουργία κυτταρικής

προσκόλλησης για υποδοχείς. Αυτό δημιούργησε μια ενδιαφέρουσα υπόθεση, ότι οι βρόχοι RGD μπορούν να προσθέσουν μια συνάρτηση κυτταρικής προσκόλλησης στους υποδοχείς. Η ανάγκη φυλογενετικής και συγκριτικής λειτουργικής έρευνας, απαιτεί ισχυρές αλλά διαισθητικές και φιλικές προς τον χρήστη μαζικές δοκιμές σύγκρισης για την επίτευξη αρχικών αποτελεσμάτων συσχετισμού. Ως αποτέλεσμα, υπάρχει μια επιτακτική ανάγκη να αντιπροσωπεύονται οι πληροφορίες σε οπτική μορφή, επιτρέποντας στους ερευνητές να αναλύουν τα δεδομένα και να αποκτούν ουσιαστικές γνώσεις, αποκαλύπτοντας τα υποκείμενα μοντέλα και ενδεχομένως, προηγούμενες αόρατες συσχετίσεις μεταξύ μεγάλων συνόλων δεδομένων.

Για τη δημιουργία της δεύτερης εφαρμογής χρησιμοποιήθηκαν εργαλεία ελεύθερου λογισμικού και λογισμικό ανοικτού κώδικα (ΕΛ/ΛΑΚ). Η εφαρμογή **fungibase** δημιουργήθηκε με σκοπό την καταγραφή και αποθήκευση των διαφόρων ψηφιακών καταθετηρίων, που επικεντρώνονταν σε χαρακτηριστικά του μικροοργανισμού ενδιαφέροντος (αλληλουχίες, μεταβολικές οδούς, φαινοτύπος) και όχι στον υποκείμενο μεθοδολογικό ιστό. Πιο συγκεκριμένα, η αλματώδης εξέλιξη της μοριακής (αλλά και της συμβατικής) μυκητολογίας στη Φαρμακευτική έρευνα, σε ιατρικό, φαρμακευτικό και βιοτεχνολογικό επίπεδο δημιούργησε ένα χάος ως προς την πρόσβαση στα αποτελέσματα της αντίστοιχης έρευνας που υπήρξε ογκώδης, άναρχη, πολυδιασπασμένη και ευκαιριακή.

Ωστόσο, ο αρχικός σχεδιασμός της fungibase δεν αφορά σε σύνθετες λειτουργίες και πολύπλοκη επεξεργασία ή εξόρυξη δεδομένων. Περιλαμβάνει στοιχειώδη εργαλεία αναζήτησης και οπτικοποίησης των δεδομένων της βάσης δεδομένων. Δίνει τη δυνατότητα στον ερευνητή να αναζητήσει πληροφορίες με βάση τη μεθοδολογία, πεδία μεθοδολογίας ή/και είδη μυκήτων, προκειμένου να ελεγχθεί αν μια τεχνική έχει εφαρμοστεί σε συγκεκριμένο μύκητα και τι αποτελέσματα είχε, ή το σε ποια είδη έχει εφαρμοστεί μια μέθοδος ή παράμετρος μεθόδου και τα αποτελέσματα, μαζί με τις μεθοδολογικές αποκλίσεις (πχ εκκινητές PCR σε διαφορετικά είδη μυκήτων με διαφορετικά προγράμματα θερμικών κυκλοποιητών).

Και στις δύο περιπτώσεις, έγινε προσπάθεια εφαρμογής της νέα και πολλά υποσχόμενης προσέγγισης για την επεξήγηση των δεδομένων, γνωστή ως οπτική εξόρυξη δεδομένων, που προέκυψε από την τεχνολογική σύζευξη των αυτοματοποιημένων αλγορίθμων εξόρυξης δεδομένων και των τεχνικών οπτικοποίησης. Η αξιοποίηση τόσο των μεθόδων αυτόματης ανάλυσης όσο και της ανθρώπινης αντίληψης υπόσχεται πιο αποτελεσματική επιθεώρηση, κατανόηση και αλληλεπίδραση με τεράστιες συλλογές δεδομένων.

Summary

In recent years, the increase and accumulation of biological data in international databases has been intense. The large number of registered nucleotides sequences (e.g. whole genomes, ESTs DNA and MRNA) and protein sequences, three-dimensional structures of biomacromolecular, as well as the mapping and sequencing of entire genomes, It has resulted in the concentration of a huge volume of crude data of extensive complexity. Researchers to respond to this challenge deemed it necessary to use computers for the efficient storage, organization, analysis and interpretation of this plethora of biological data. With the simultaneous intense development of informatics, the appropriate conditions were created for the development of a new and constantly evolving scientific field, bioinformatics.

The purpose of this work is the development, application and study of tools, different technologies with main objectives, (a) the efficient organization of existing biological data and access to them as well as the accumulation of new data, (b) the Development of methods and computational tools for data visualization in order to extract information from data and (c) the use of these tools for the analysis and interpretation of data in a biologically acceptable way (Reichhardt, 1999).

The first application was created using technology and tools developed and supported by Microsoft. This is RGDtrip, an application designed and implemented, for the purpose of storing, managing and processing protein data containing the tripeptide RGD. The research interest appeared from the first bibliographic review and evaluation of all RGD and loop sequences in receptors between species in 1998 and the verification that the occurrence of such sequences in receptors could mean that these The sequences would be in loop/loop-like structures and that

these loop structures might entail a cell-adhesion function for receptors. This created an interesting hypothesis that RGD loops can add a cell adhesion function to the receptors. The need for phylogenetic and comparative functional research requires strong but intuitive and user-friendly massive comparison tests to achieve initial correlation results. As a result, there is an urgent need to represent the information in a visual format, allowing researchers to analyse the data and gain substantial knowledge, revealing the underlying models and possibly previous Invisible correlations between large datasets.

The creation of the second application used free software tools and open source software (EL/LAC). The Fungibase application was created for the purpose of recording and storing the various digital repositories, which focused on characteristics of the micro-organism of interest (sequences, metabolic pathways, phenotype) and not on the underlying Methodological tissue. More specifically, the booming evolution of molecular (but also conventional) mycology in pharmaceutical research, in medical, pharmaceutical and biotechnological level, has created a chaos in access to the results of the corresponding research It has been massive, unregulated, fragmented and opportunistic.

However, the original design of fungibase does not involve complex operations and complex processing or data mining. It includes incremental tools for searching and visualizing database data. It enables the researcher to seek information based on methodology, methodology fields and/or species of fungi, in order to check whether a technique has been applied to a particular fungus and what results it had, or what species has been applied method or parameter of method and the results, along with methodological deviations (eg PCR starters in different types of fungi with different programs of thermal cyclers).

In both cases, an attempt was made to implement the new and promising approach for the explanation of data, known as optical data mining, resulting from the technological coupling of automated data mining algorithms and visualization techniques. The utilization of both automatic analysis methods and human perception promises more effective inspection, understanding and interaction with vast data collections.