ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνικές Διαχείρισης Δεδομένων σε Βιοεπιστήμες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΜΙΧΑΗΛ Α. ΠΑΠΑΔΟΠΟΥΛΟΥ

Επιβλέπων : Σελλής Τιμολέων
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΓΝΩΣΕΩΝ ΚΑΙ ΔΕΔΟΜΕΝΩΝ

Αθήνα, Ιούλιος 2007

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνικές Διαχείρισης Δεδομένων σε Βιοεπιστήμες

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΜΙΧΑΗΛ Α. ΠΑΠΑΔΟΠΟΥΛΟΥ

Επιβλέπων : Σελλής Τιμολέων
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Ιουλίου 2007.

.....	.....	.....
Τιμολέων Σελλής	Ιωάννης Βασιλείου	Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Επικ. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2007

.....

ΜΙΧΑΗΛ Α. ΠΑΠΑΔΟΠΟΥΛΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Παπαδόπουλος Μιχαήλ, Τεχνικές Διαχείρισης Δεδομένων σε Βιοεπιστήμες, Διπλωματική Εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών, Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων, 19 Ιουλίου 2007.

Σελίδες: 174

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας, Καθηγητή Τιμολέοντα Σελλή καθώς και τον Διδάκτορα Θοδωρή Δαλαμάγκα, για την άψογη συνεργασία μας, αλλά και για την ευκαιρία που μου έδωσαν να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.
Επίσης, θέλω να ευχαριστήσω τους γονείς μου Τάσο και Χριστίνα, τα αδέρφια μου Προκόπη και Φλώρα και τον θείο Τάσο, για τη συμπαράστασή τους.
Η προσπάθειά μου αυτή αφιερώνεται στον παππού Προκόπη, τον παππού Μιχάλη και τη γιαγιά Σοφία, που ήρθαν από μακριά και στη γιαγιά Φλώρα.

Περίληψη

Πολύχρονες ερευνητικές προσπάθειες, επιστημόνων από όλο τον κόσμο, οδήγησαν στην ανακάλυψη του καθοριστικού ρόλου που το DNA και οι παραγόμενες από αυτό πρωτεΐνες, διαδραματίζουν στους ζωντανούς οργανισμούς. Το επόμενο επιστημονικό επίτευγμα ήταν η αναγωγή των πολύπλοκων μορίων DNA ή πρωτεϊνών στην ευανάγνωστη και εύκολα απόθηκεύσιμη μορφή των ακολουθιών. Η παρούσα διπλωματική εργασία λοιπόν, αρχικά εισάγει τον αναγνώστη στις απαραίτητες βιολογικές έννοιες και διαδικασίες, δίνοντας έμφαση σε ακολουθίες αμινοξέων και νουκλεοτιδίων. Έπειτα, καταγράφονται τα σημαντικότερα πρότυπα αναπαράστασης βιολογικών ακολουθιών, τα οποία χαρακτηρίζονται από την XML δομή τους. Στη συνέχεια περιγράφεται η τεχνική ευθυγράμμισης βιολογικών ακολουθιών, η οποία στοχεύει στον εντοπισμό ομοιοτήτων ανάμεσα σε αυτές. Συγκεκριμένα γίνεται η μελέτη των προγραμμάτων BLAST, FASTA και CLUSTAL, τόσο σε θεωρητικό, όσο και σε πειραματικό επίπεδο. Ο βαθμός ομοιότητας ακολουθιών που προκύπτει από την τεχνική της ευθυγράμμισης, μπορεί στη συνέχεια να χρησιμοποιηθεί προκειμένου να απεικονιστούν σε μορφή πίνακα οι εξελικτικές σχέσεις μεταξύ ακολουθιών, ή κατ' επέκταση μεταξύ των ειδών. Με βάση έναν τέτοιον πίνακα και με τη χρήση αλγορίθμων ή πακέτων λογισμικού, κατασκευάζονται τα φυλογενετικά δένδρα, που απεικονίζουν εξελικτικές αποστάσεις μεταξύ ακολουθιών ή ειδών. Εν προκειμένω, μελετώνται οι αλγόριθμοι UPGMA και Neighbor-Joining καθώς και το πακέτο φυλογενετικής ανάλυσης PHYLIP. Τέλος η διπλωματική εργασία περιγράφει και αναλύει τον τρόπο με τον οποίο οι σεναριακές γλώσσες προγραμματισμού Perl και Python, συντελούν στην αυτοματοποίηση και διευκόλυνση εργασιών και ερευνών πάνω σε βιολογικά δεδομένα, κάνοντας ιδιαίτερη αναφορά στις κανονικές εκφράσεις και στους τρόπους αναζήτησης μοτίβων σε ακολουθίες.

Λέξεις Κλειδιά:

DNA, νουκλεοτίδια, πρωτεΐνες, αμινοξέα, ακολουθίες, XML πρότυπα, ευθυγράμμιση ακολουθιών, Needleman-Wunch, Smith-Waterman, BLAST, FASTA, CLUSTAL, φυλογενετικά δένδρα, UPGMA, Neighbor-Joining, PHYLIP, Perl, Python, μοτίβα, κανονικές εκφράσεις.

Abstract

Many years of research carried out by scientists from all over the world, have led to the discovery of the vital role that DNA and its resulting proteins play in living organisms. The next scientific achievement was the reduction of the complex DNA or protein molecules to the easily readable and storable sequence format. The present diploma thesis, firstly introduces the reader to the essential biological terms and processes, emphasizing on aminoacid and nucleotide sequences. Then, the most important standards for representation of biological data, characterized by XML structure, are noted. Next, there is a description of the biological sequence alignment technique, which aims at the detection of similarities among sequences. Specifically, BLAST, FASTA, and CLUSTAL programs, are studied both in theoretical and practical level. The level of sequence similarity, resulting from the alignment technique, can then be used to show evolutionary relations among sequences, or among species, in array format. Based on such an array, algorithms and software packages construct phylogenetic trees. The latter, depict evolutionary distances among sequences or species. Therefore, there is a study of UPGMA and Neighbor-Joining algorithms and of the PHYLIP phylogenetic analysis package. Finally, this thesis describes and analyses how the script programming languages, Perl and Python, contribute in the automatization and facilitation of tasks and research on biological data, putting special emphasis on regular expressions and on ways of locating motifs in sequences.

Keywords:

DNA, nucleotides, proteins, aminoacids, translation, sequences, XML standards, sequence alignment, Needleman-Wunch, Smith-Waterman, BLAST, FASTA, CLUSTAL, phylogenetic trees, UPGMA, Neighbor-Joining, PHYLIP, Perl, Python, motifs, regular expressions.

Πίνακας περιεχομένων

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

1

Εισαγωγή

1.1 *Αντικείμενο της διπλωματικής*

Η διπλωματική αυτή εργασία πραγματεύεται τη μελέτη και ανάλυση μερικών εκ των σημαντικότερων τεχνικών διαχείρισης δεδομένων βιοεπιστημών. Τα δεδομένα αυτά είναι κυρίως αμινοξέα και νουκλεοτίδια, οι ακολουθίες των οποίων συνθέτουν τις πρωτεΐνες και τις αλυσίδες DNA αντιστοίχως. Το επίτευγμα της αναπαράστασης των πολύπλοκων μορίων DNA ή πρωτεϊνών είναι εξαιρετικά σημαντικό τόσο για τον καλύτερο χειρισμό και την αποθήκευση τους, όσο και για την ανάπτυξη της έρευνας με απώτερο στόχο την καλύτερη κατανόηση των μηχανισμών λειτουργίας του κυττάρου. Η ανάγκη για την αποθήκευση, αναπαράσταση και τον συμμερισμό δεδομένων βιολογικών ακολουθιών, ώθησε στη δημιουργία προτύπων με κοινή XML δομή. Η XML είναι ιδανική για συστήματα διαχείρισης γνώσης και γίνεται σταδιακά αποδεκτή στο χώρο των Βιοεπιστημών. Εν προκειμένω, καταγράφονται ορισμένα εκ των σημαντικοτέρων προτύπων αναπαράστασης δεδομένων Βιοεπιστημών.

Προκειμένου να ανιχνευθούν δομικής ή λειτουργικής σημασίας ομοιότητες ανάμεσα στις βιολογικές ακολουθίες, γίνεται σύγκριση των ακολουθιών. Από την οπτική γωνία της Βιολογίας το κίνητρο της σύγκρισης ακολουθιών είναι ότι εφόσον όλοι οι ζωντανοί οργανισμοί συσχετίζονται εξελικτικά μεταξύ τους, θα πρέπει τα γονίδια και οι πρωτεΐνες των ειδών που είναι εξελικτικά κοντά να παρουσιάζουν μεγάλες ομοιότητες. Οι προς σύγκριση ακολουθίες πρέπει να είναι του ίδιου τύπου (DNA-DNA ή πρωτεΐνες-πρωτεΐνες) ή αν είναι

διαφορετικού, να μετατραπούν σε ίδιου τύπου ακολουθίες και στη συνέχεια να συγκριθούν. Η τεχνική σύγκρισης που μελετάται στην παρούσα διπλωματική εργασία είναι αυτή της ευθυγράμμισης. Στην ευθυγράμμιση, δύο ή περισσότερες ακολουθίες ευθυγραμμίζονται κατακόρυφα, με τρόπο τέτοιον ώστε να μεγιστοποιείται ένας βαθμός-μέτρο ομοιότητας. Η τεχνική αυτή μελετάται τόσο σε θεωρητικό, όσο και σε πειραματικό επίπεδο, μέσω των δημοφιλέστερων προγραμμάτων.

Οι βαθμοί ομοιότητας που έχουν προκύψει από την τεχνική της ευθυγράμμισης ακολουθιών μπορούν να αναπαρασταθούν γραφικά με τη μορφή δενδρών, τα οποία ονομάζονται φυλογενετικά. Τα δένδρα αυτά απεικονίζουν εξελικτικές αποστάσεις-σχέσεις μεταξύ βιολογικών ακολουθιών και κατ' επέκταση μεταξύ ειδών. Η κατασκευή τους είναι απαραίτητη, καθώς μπορούν να αποκαλύψουν εξελικτικές ασυνέπειες και να καταστήσουν έγκυρες ή άκυρες, υποθέσεις που έχουν γίνει για πιθανούς προγόνους διαφόρων οργανισμών. Στην εργασία αυτή κατασκευάζονται φυλογενετικά δένδρα τόσο μέσω αλγορίθμων, όσο και μέσω έτοιμων πακέτων φυλογενετικής ανάλυσης.

Η επόμενη τεχνική που μελετάται είναι η χρήση σεναριακών γλωσσών προγραμματισμού για χειρισμό δεδομένων Βιοεπιστημών. Οι σεναριακές γλώσσες είναι εξαιρετικά χρήσιμες στο χώρο αυτόν καθώς επιταχύνουν, διευκολύνουν και αυτοματοποιούν ενέργειες και έρευνες, απαλλάσσοντας τον χρήστη-ερευνητή από κοπιαστικές χειρωνακτικές διαδικασίες. Ένα μεγάλο επίσης πλεονέκτημα των γλωσσών αυτών είναι ότι ως σεναριακές είναι αρκετά εύληπτες από τους Βιολόγους. Δύο εκ των δημοφιλέστερων σεναριακών γλωσσών περιγράφονται θεωρητικά, αλλά και πρακτικά μέσω παραδειγμάτων.

Συνοψίζοντας λοιπόν, οι τεχνικές που η διπλωματική αυτή εργασία περιγράφει, έχουν στόχους όπως η αποθήκευση και η αναπαράσταση πληροφοριών βιολογικών ακολουθιών, ο εντοπισμός ομοιοτήτων μεταξύ ακολουθιών, η απεικόνιση εξελικτικών σχέσεων μεταξύ ειδών ή ακολουθιών, καθώς και η αυτοματοποίηση ερευνών πάνω σε βιολογικά δεδομένα. Οι τεχνικές αυτές περιγράφονται σε θεωρητικό αλλά και πειραματικό επίπεδο, μέσα από παραδείγματα, ενώ συγκρίνονται και τα σημαντικότερα χαρακτηριστικά τους.

1.2 *Οργάνωση του τόμου*

Στο πρώτο κεφάλαιο παρατίθεται το αντικείμενο, οι κυριότεροι στόχοι και τα κίνητρα εκπόνησης της διπλωματικής εργασίας, καθώς και η οργάνωση των κεφαλαίων και ενοτήτων του τόμου.

Το δεύτερο κεφάλαιο επιχειρεί μία σύντομη περιγραφή των βιολογικών δεδομένων που χειριζόμαστε στα επόμενα κεφάλαια, ξεκινώντας από το βασικότερο μόριο κάθε ζωντανού οργανισμού, το DNA. Το πολύπλοκο αυτό μόριο περιέχει κωδικοποιημένη όλη την απαραίτητη πληροφορία για τη λειτουργία του κάθε κυττάρου και μεταξύ των άλλων είναι υπεύθυνο

για τη σύνθεση των πρωτεϊνών. Οι πρωτεΐνες είναι κι αυτές πολύ σημαντικά μόρια με ξεχωριστό ρόλο στη ζωή του κυττάρου. Εφόσον τα μόρια του DNA και τα μόρια πρωτεϊνών αποτελούνται από συνδεδεμένες σε μορφή αλυσίδας υπομονάδες, γίνεται δυνατή η αναπαράστασή τους σε μορφή ακολουθιών. Ανακαλύπτοντας μάλιστα, πως ανάμεσα στις υπομονάδες του DNA (νουκλεοτίδια), το μόνο που διαφέρει είναι οι αζωτούχες βάσεις, μπορούμε να αναπαραστήσουμε ένα μόριο DNA με δύο ακολουθίες (επειδή είναι δίκλωνο μόριο) αζωτούχων βάσεων. Αντίστοιχα, μία πρωτεΐνη μπορεί να αναπαρασταθεί με μία ακολουθία αμινοξέων. Η αναπαράσταση σε μορφή ακολουθίας είναι εξαιρετικά σημαντική, γιατί συντελεί στην καλύτερη κατανόηση της λειτουργίας των γονιδίων, των πρωτεϊνών, και κατ' επέκταση των κυττάρων, αλλά και διευκολύνει την έρευνα των βιολογικών δεδομένων.

Στη συνέχεια το κεφάλαιο αυτό καταγράφει πρότυπα αναπαράστασης και μοντελοποίησης πληροφοριών στο χώρο των Βιοεπιστημών. Τα πρότυπα αυτά είναι κυρίως γλώσσες τύπου XML, οι οποίες έχουν αναπτυχθεί από διάφορες επιστημονικές ομάδες, με σκοπό να εξυπηρετηθούν στόχοι όπως η καταγραφή και αποθήκευση πρωτεϊνικών ή νουκλεοτιδικών ακολουθιών συνοδευόμενων από σχόλια και περιγραφές, η αναπαράσταση και περιγραφή βιολογικών πειραμάτων, ο συμμερισμός και η ανταλλαγή βιολογικών μοντέλων. Όλα τα παραπάνω γίνονται δυνατά κυρίως λόγω της δομής της γλώσσας XML, κάτι που την καθιστά ιδανική για συστήματα διαχείρισης γνώσης. Συγκεκριμένα, η δομή της είναι ιεραρχική, με χρήση ετικετών, κάτι που επιτρέπει να συνδυάζονται τα δεδομένα με επιπλέον πληροφορίες για αυτά (μεταδεδομένα).

Στο τρίτο κεφάλαιο περιγράφεται η τεχνική της ευθυγράμμισης πρωτεϊνικών και νουκλεοτιδικών ακολουθιών η οποία έχει ως στόχο την αναζήτηση δομικών ή λειτουργικών ομοιοτήτων ανάμεσα σε δεδομένα που έχουν αποθηκευθεί σε βάσεις δεδομένων. Η ευθυγράμμιση έγκειται στην κατακόρυφη στοίχιση δύο ή περισσοτέρων ακολουθιών, προκειμένου να εντοπιστούν περιοχές σημαντικού βαθμού ομοιότητας, οι οποίες και μας επιτρέπουν να φτάνουμε σε συμπεράσματα για το βαθμό εγγύτητας των εξεταζόμενων ακολουθιών. Το πρόβλημα της ευθυγράμμισης ακολουθιών είναι ένα πρόβλημα βελτιστοποίησης, καθώς υπάρχουν πολλές πιθανές ευθυγραμμίσεις και μία μόνο βέλτιστη. Προκειμένου να βρεθεί η βέλτιστη λύση χρησιμοποιείται ο δυναμικός προγραμματισμός και συγκεκριμένα οι αλγόριθμοι των Needleman-Wunch και Smith-Waterman. Παρά το γεγονός λοιπόν, ότι ο δυναμικός προγραμματισμός μας εγγυάται την εύρεση της βέλτιστης λύσης, είναι εξαιρετικά κοστοβόρος από άποψη χώρου και χρόνου. Για να λυθεί αυτό το πρόβλημα, θυσιάζουμε μέρος της ευαισθησίας του δυναμικού προγραμματισμού, καταφεύγοντας σε ευριστικές μεθόδους. Οι μέθοδοι αυτές, αν και δεν εγγυώνται της εύρεσης της βέλτιστης λύσης, εγγυώνται ότι θα βρουν πολύ γρήγορα, μία λύση αρκετά κοντά στη βέλτιστη. Τέτοιες μεθόδους χρησιμοποιούν και τα προγράμματα BLAST, FASTA και CLUSTAL τα οποία και

μελετούμε. Τα προγράμματα BLAST και FASTA πραγματοποιούν τοπικές ευθυγραμμίσεις ακολουθιών κατά ζεύγη, ενώ το πρόγραμμα CLUSTAL πραγματοποιεί πολλαπλές ευθυγραμμίσεις ακολουθιών καθ' όλο το μήκος τους.

Το τέταρτο κεφάλαιο ασχολείται με την παρουσίαση και κατασκευή εξελικτικών-φυλογενετικών δενδρών. Τα δένδρα αυτά αναπαριστούν εξελικτικές αποστάσεις μεταξύ των διαφόρων ειδών του ζωικού βασιλείου ή γενικότερα μεταξύ βιολογικών ακολουθιών. Η μέθοδος της ευθυγράμμισης ακολουθιών, η οποία και ερευνά το βαθμό ομοιότητας ακολουθιών, μπορεί να χρησιμοποιηθεί στην κατασκευή ενός πίνακα που απεικονίζει εξελικτικές σχέσεις μεταξύ ακολουθιών ή ειδών. Με βάση έναν τέτοιον πίνακα οι αλγόριθμοι UPGMA και Neighbor-Joining κατασκευάζουν φυλογενετικά δένδρα. Οι αλγόριθμοι αυτοί είναι οι δημοφιλέστερες μέθοδοι αποστάσεων και ονομάζονται έτσι, ακριβώς επειδή εκτελούνται με βάση πίνακες εξελικτικών αποστάσεων. Στο κεφάλαιο αυτό λοιπόν παρουσιάζονται οι αλγόριθμοι αυτοί βήμα προς βήμα μέσα από παραδείγματα. Το δημοφιλέστερο πακέτο φυλογενετικής ανάλυσης PHYLIP είναι επίσης αντικείμενο μελέτης σε αυτό το κεφάλαιο. Το πακέτο αυτό εμπεριέχει πολλές μεθόδους φυλογενετικής ανάλυσης και επί μέρους προγράμματα.

Στο πέμπτο κεφάλαιο παρουσιάζονται, από την οπτική γωνία των Βιοεπιστημών, οι σεναριακές γλώσσες προγραμματισμού Perl και Python, οι οποίες είναι οι πλέον εύχρηστες στο χώρο των Βιοεπιστημών. Οι γλώσσες αυτές είναι αρκετά φιλικές προς το χρήστη και τα προγράμματα-σενάριά τους στοχεύουν στην αυτοματοποίηση, διευκόλυνση, επιτάχυνση και αλυσιδωτή εκτέλεση εργασιών σε βιολογικά δεδομένα. Στις γλώσσες αυτές αναπτύσσονται παραδείγματα χειρισμού και έρευνας βιολογικών ακολουθιών, όπως ο υπολογισμός συμπληρωματικών νουκλεοτιδικών ακολουθιών, η αναζήτηση μοτίβων σε ακολουθίες μέσω της χρήσης κανονικών εκφράσεων, ή η χρήση των γλωσσών αυτών για διαδικτυακές εκτελέσεις προγραμμάτων όπως το BLAST.

Το έκτο κεφάλαιο συνοψίζει την εργασία, παρουσιάζοντας κάποια βασικά σημεία και συμπεράσματα και προτείνει πιθανές περιοχές μελλοντικής έρευνας, ενώ το έβδομο κεφάλαιο καταγράφει τη βιβλιογραφία και τους διαδικτυακούς τόπους που χρησιμοποιήθηκαν.

1.3 *Συνεισφορά*

Η διπλωματική εργασία αυτή αποτελεί μία προσπάθεια αναλυτικής περιγραφής και επεξήγησης μερικών εκ των σημαντικότερων τεχνικών διαχείρισης δεδομένων που χρησιμοποιούνται στις Βιοεπιστήμες. Συγκεκριμένα η εν λόγω εργασία:

- Καταγράφει πρότυπα τύπου XML που επιτρέπουν την αποθήκευση, αναπαράσταση ή συμμερισμό βιολογικών μοντέλων και δεδομένων, συγκρίνοντας βασικά χαρακτηριστικά τους.

- Αναλύει την τεχνική της ευθυγράμμισης βιολογικών ακολουθιών μέσω αλγορίθμων και έτοιμων λογισμικών πακέτων, που μελετώνται σε παραδείγματα. Σημειώνονται επίσης, τα βασικά χαρακτηριστικά, τα πλεονεκτήματα και μειονεκτήματα, καθώς και οι διαφορές των μεθόδων αυτών.
- Καταγράφει τους σημαντικότερους και δημοφιλέστερους αλγορίθμους και πακέτα κατασκευής φυλογενετικών δενδρών, αναλύοντας τα μειονεκτήματα και πλεονεκτήματά τους.
- Χρησιμοποιεί σεναριακές γλώσσες προγραμματισμού για χειρισμό δεδομένων Βιοεπιστημών παραθέτοντας παραδείγματα, συγκρίνοντας τις γλώσσες αυτές και καταγράφοντας τα πλεονεκτήματα και μειονεκτήματά τους.

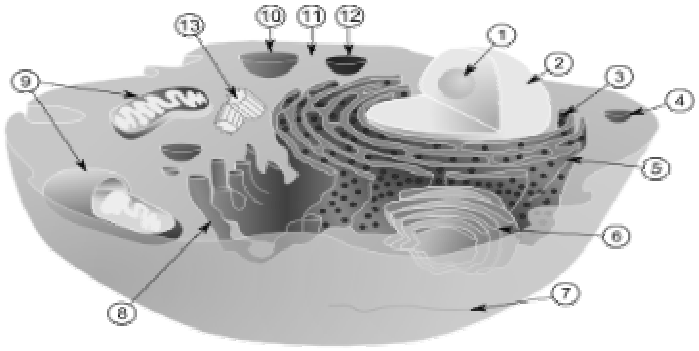
2

Βιολογία: Βασικές έννοιες και αναπαράσταση δεδομένων

2.1 *Το φαινόμενο της ζωής*

Το φαινόμενο της ζωής εκδηλώνεται από τη δομική και λειτουργική μονάδα που ονομάζεται κύτταρο. Όλοι οι οργανισμοί, με εξαίρεση τους ιούς, αποτελούνται από ένα ή περισσότερα κύτταρα. Ως κύτταρο εννοούμε μια συστηματικά οργανωμένη ομάδα μορίων, που βρίσκονται σε δυναμική αλληλεπίδραση μεταξύ τους. Το κύτταρο διαθέτει μορφολογική, φυσική και χημική οργάνωση και την ικανότητα της αφομοίωσης, της ανάπτυξης και της αναπαραγωγής . Η μορφή του και τα δομικά του στοιχεία απεικονίζονται στο Σχήμα 2.1.

Ως οργανισμός, το κύτταρο διαθέτει την ικανότητα να ζει, ακόμη και χωρίς την ύπαρξη άλλων κυττάρων. Η ιδιότητα αυτή προϋποθέτει της ύπαρξης μιας μεταβολικής μηχανής που μπορεί να αντλήσει ενέργεια από το περιβάλλον και να τη χρησιμοποιήσει σε ουσιώδεις βιοχημικές διεργασίες, που περιλαμβάνουν την κίνηση ουσιών, την εκλεκτική μεταφορά μορίων μέσα και έξω από το κύτταρο και την ικανότητα αλλαγής και διαμόρφωσής τους, δηλαδή της προσαρμογής τους στις περιβάλλουσες φυσικές και χημικές συνθήκες. Προκειμένου όμως να γίνουν οι απαραίτητες πολύπλοκες χημικές αντιδράσεις, χρειάζονται ακριβείς οδηγίες, τις οποίες δίνει το γενετικό υλικό από τον πυρήνα του κυττάρου.



Σχήμα 2.1: Ζωικό κύτταρο

(1.Πυρηνίσκος 2.Πυρήνας 3.Ριβόσωμα 4.Κυστίδιο 5.Αδρό ενδοπλασματικό δίκτυο 6.Σωμάτιο Golgi 7.Κυτταρικός σκελετός 8.Λείο ενδοπλασματικό δίκτυο 9.Μιτοχόνδριο 10.Κενοτόπιο 11.Κυτταρόπλασμα 12.Λυσόσωμα 13.Κεντρύλλιο)

Ο πυρήνας λοιπόν είναι ο σημαντικότερος σχηματισμός του κυττάρου και μπορεί να συγκριθεί με τον ανθρώπινο εγκέφαλο. Βρίσκεται συνήθως στο κέντρο του κυττάρου και το σχήμα του είναι σφαιρικό ή ωοειδές. Οι πληροφορίες που περιέχει ονομάζονται γενετικές και συναντούνται με τη μορφή κώδικα, σε ένα μεγάλο και πολύπλοκο μόριο, το DNA (δεοξυριβονουκλεϊκό οξύ). Το μόριο αυτό αποτελείται από μικρότερα μόρια (νουκλεοτίδια) και ενώνεται με πρωτεΐνες για να συγκροτήσει μακριά νημάτια (χρωμονημάτια). Τα νημάτια αυτά συμπυκνώνονται κατά τη διαίρεση του κυττάρου, παίρνουν συγκεκριμένο σχήμα, οπότε και ονομάζονται χρωμοσώματα, συγκεκριμένος αριθμός των οποίων υπάρχει σε κάθε είδος οργανισμού. Στο εσωτερικό του πυρήνα διακρίνονται ένα ή δύο περίπου σφαιρικά σωματίδια που ονομάζονται πυρηνίσκοι. Στους πυρηνίσκους εκτός από DNA υπάρχει και το RNA (ριβονουκλεϊκό οξύ), που συμμετέχει στη σύνθεση των πρωτεϊνών.

2.2 *Ιστορική αναδρομή στην έρευνα του DNA*

Η ανακάλυψη ότι το DNA είναι ο φορέας της γενετικής πληροφορίας είναι το αποτέλεσμα μιας σειράς επιστημονικών ερευνών που διήρκησε πολλά χρόνια. Ενώ η ύπαρξη του στον πυρήνα των κυττάρων πιστοποιήθηκε ήδη από το 1869, ήταν στα μέσα του 20ου αιώνα που οι ερευνητές ξεκίνησαν να υποθέτουν ότι μπορεί να αποθηκεύει γενετική πληροφορία. Τα νουκλεϊκά οξέα ανακαλύφθηκαν το 1869 από τον Φρίντριχ Μίσερ. Ο Μίσερ ανακάλυψε μέσα σε πυρήνες κυττάρων την ύπαρξη μιας ουσίας με συγκεκριμένη όξινη αντίδραση την οποία ονόμασε νουκλεΐνη (από το λατινικό nucleus που σημαίνει πυρήνας). Αργότερα απομόνωσε από το σπέρμα σολομού δείγμα της ουσίας που σήμερα αποκαλούμε DNA και το 1889 ο μαθητής του Ρίτσαρντ Άλτμαν την ονόμασε νουκλεϊκό οξύ. Την ίδια περίπου εποχή ο αυστριακός μοναχός Γκρέγκορ Μέντελ ανακάλυπτε τους νόμους της Γενετικής. Πέρασαν όμως 75 χρόνια προκειμένου να φανεί ότι η ανακάλυψη του Μίσερ αποτελούσε τη μοριακή βάση της ανακάλυψης του Μέντελ.

Σημαντικό ρόλο στην ανακάλυψη του γενετικού ρόλου του DNA είχε το βακτήριο του πνευμονιόκοκκου. Το 1928 ο Φρεντ Γκρίφιθ χρησιμοποίησε δύο στελέχη του συγκεκριμένου βακτηρίου (Diplococcus pneumoniae), τα οποία ξεχωρίζουν μορφολογικά όταν καλλιεργηθούν σε θρεπτικό υλικό. Αυτά είναι το λείο βακτήριο (συμβολίζεται με S από το smooth = λείος) που ονομάζεται έτσι επειδή δημιουργεί λείες αποικίες ενώ ταυτόχρονα είναι παθογόνο, και το αδρό βακτήριο (συμβολίζεται με R από το rough = αδρός) που δημιουργεί αδρές αποικίες και δεν είναι παθογόνο. Ο Γκρίφιθ ανακάλυψε ότι το μη παθογόνο βακτήριο (R) μπορεί να μετατραπεί σε παθογόνο (S), χορηγώντας σε ένα ποντίκι ένα μείγμα βακτηρίων από ζωντανά αδρά βακτήρια και νεκρά λεία βακτήρια. Το μείγμα αποδείχτηκε παθογόνο, ενώ καθένα από τα συστατικά του από μόνο του δεν ήταν. Τόσο τα ζωντανά αδρά βακτήρια, όσο και τα νεκρά λεία βακτήρια από μόνα τους δεν ήταν παθογόνα. Ο Γκρίφιθ συμπέρανε ότι με κάποιο τρόπο μερικά αδρά βακτήρια μετασχηματίστηκαν σε λεία παθογόνα, χωρίς όμως να δώσει ικανοποιητική εξήγηση για τον τρόπο που γίνεται κάτι τέτοιο.

Η απάντηση δόθηκε το 1944, όταν οι Όσβαλντ Άβερι, Κόλιν Μακλέοντ και Μακλίν Μακάρτι επανέλαβαν τα πειράματα του Γκρίφιθ σε δοκιμαστικό σωλήνα εργαστηρίου (in vitro). Ο Άβερι και οι συνεργάτες του διαχώρισαν τα διάφορα συστατικά των νεκρών λείων βακτηρίων σε υδατάνθρακες, πρωτεΐνες, RNA, DNA κ.α. και ερεύνησαν ποιο από αυτά μπορούσε να μετασχηματιστεί. Τα αποτελέσματά τους έδειξαν ότι το συστατικό που προκαλούσε το μετασχηματισμό των αδρών βακτηρίων σε λεία ήταν το DNA. Ένα τέτοιο εύρημα ήταν μία πολύ καλή ένδειξη ότι το DNA αποτελεί το γενετικό υλικό και αποτέλεσε την αρχή μιας επαναστατικής περιόδου για τις βιολογικές επιστήμες.

Σημείο σταθμό σε αυτή τη περίοδο αποτελεί η ανακάλυψη της δομής του DNA που πραγματοποιήθηκε το 1953 από τους Τζέιμς Γουάτσον και Φράνσις Κρικ, δύο Βρετανούς ερευνητές που εργάζονταν στο Πανεπιστήμιο του Καίμπριτζ. Η ανακάλυψη τους όμως, μάλλον θα πρέπει να αντιμετωπίζεται ως το αποτέλεσμα μιας σειράς σχετικών ερευνητικών δεδομένων, παρά ως μια μεμονωμένη επαναστατική ανακάλυψη. Για παράδειγμα, από το 1948 ο Λάινους Πόλινγκ είχε ανακαλύψει ότι αρκετές πρωτεΐνες περιελάμβαναν σχήματα με ελικοειδή δομή, πραγματοποιώντας πειράματα με χρήση ακτίνων Χ. Επίσης, από το 1947 ο Έρβιν Τσάργκαφ είχε παρατηρήσει κάτι χαρακτηριστικό: σε οποιοδήποτε δείγμα DNA, το ποσοστό των νουκλεοτιδίων που έχουν ως αζωτούχο βάση την αδενίνη είναι ίσο με το ποσοστό των νουκλεοτιδίων που έχουν ως αζωτούχο βάση την θυμίνη, ενώ το ποσοστό των νουκλεοτιδίων που έχουν ως αζωτούχο βάση την γουανίνη είναι ίσο με το ποσοστό των νουκλεοτιδίων που έχουν ως αζωτούχο βάση την κυτοσίνη.

Οι Γουάτσον και Κρικ βασίστηκαν ιδιαίτερα στην έρευνα της Ροζαλίν Φράνκλιν. Συγκεκριμένα, στηρίχθηκαν στα εξής:

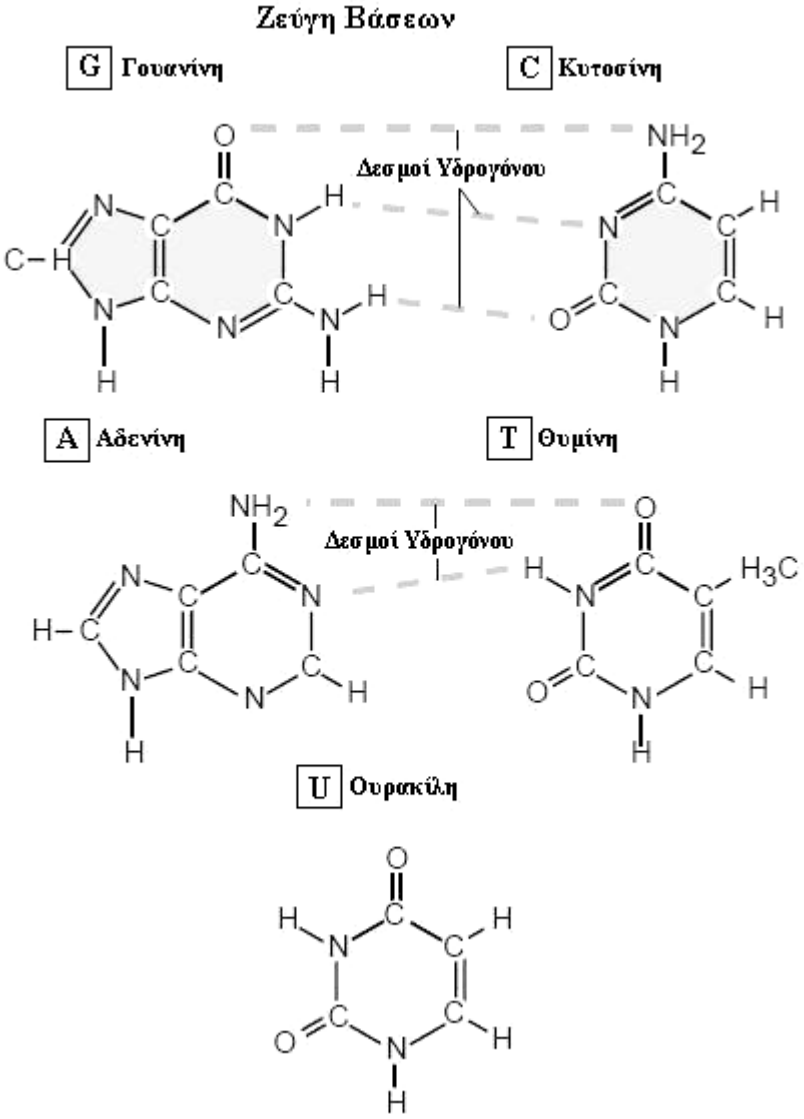
- Σε μια φωτογραφία του DNA που ο Μορίς Γουίλκινς είχε πάρει από το γραφείο της Φράνκλιν και την έδειξε στον Γουάτσον. Εκείνος αναγνώρισε τη διπλή έλικα, κάτι που τον στήριξε στην συνέχιση των ερευνών του.
- Στις μετρήσεις της Φράνκλιν στο κυτταρικό DNA, όπως παρουσιάζονταν σε μια μη δημόσια έκθεση που είδε ο Κρικ. Έτσι αντιλήφθηκε ότι οι δυο έλικες του DNA κινούνται σε αντίθετες κατευθύνσεις (είναι αντιπαράλληλες).

Η προσφορά της Φράνκλιν, που πέθανε νέα από καρκίνο των ωοθηκών λόγω των ραδιενεργών υλικών που χρησιμοποιούσε στη δουλειά της, αναγνωρίστηκε μετά το θάνατό της. Τα αποτελέσματα των εργασιών των Γουάτσον και Κρικ ανακοινώθηκαν στις 25 Απριλίου 1953, στο περιοδικό Nature. Για τη συνεισφορά τους στη μελέτη της δομής του DNA, οι Γουάτσον και Κρικ μοιράστηκαν το 1962 το Βραβείο Νομπέλ με τον Μορίς Γουίλκινς. Το 1957 οι Γουάτσον και Κρικ πρότειναν το κεντρικό δόγμα της μοριακής βιολογίας, στο οποίο περιγράφουν τη διαδικασία με την οποία πρωτεΐνες παράγονται από το DNA του πυρήνα. Σημαντικά επίσης σημεία της έρευνας σχετικά με το DNA, αποτελούν η ανακάλυψη του μηχανισμού σύνθεσης του DNA από τον Άρθουρ Κορνμπεργκ το 1956 και η ανακάλυψη του γενετικού κώδικα από το Μάρσαλ Νίρενμπεργκ το 1961.

2.3 *Νουκλεϊκά οξέα*

Τα είδη των νουκλεϊκών οξέων που υπάρχουν στα κύτταρα είναι δύο: το δεοξυριβονουκλεϊκό οξύ (DNA) και το ριβονουκλεϊκό οξύ (RNA). Οι δομικές μονάδες ενός νουκλεϊκού οξέος είναι τα νουκλεοτίδια που είναι σύνθετα μόρια, αποτελούνται δηλαδή από τρία μόρια συνδεδεμένα μεταξύ τους. Τα μόρια αυτά είναι: μία πεντόζη (σάκχαρο που αντλεί το όνομά του από τα πέντε άτομα άνθρακα που περιέχει), μία οργανική βάση και ένα φωσφορικό οξύ. Τα νουκλεοτίδια του DNA περιέχουν την πεντόζη δεοξυριβόζη, ενώ τα νουκλεοτίδια του RNA την πεντόζη ριβόζη. Οι αζωτούχες βάσεις είναι οι: αδενίνη (A), γουανίνη (G), κυτοσίνη (C), θυμίνη (T) και ουρακίλη (U). Η αδενίνη, η γουανίνη και η κυτοσίνη συναντώνται και στα δύο είδη νουκλεϊκών οξέων, η θυμίνη υπάρχει μόνο στο μόριο του DNA και η ουρακίλη μόνο στο RNA. Τα μόρια του φωσφορικού οξέος είναι ίδια σε όλα τα νουκλεοτίδια.

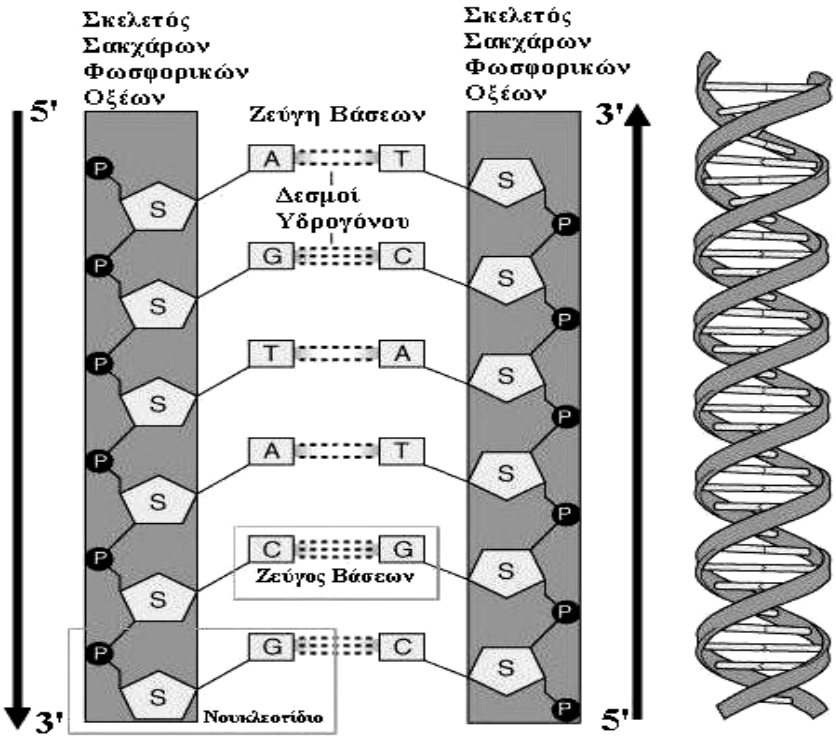
Το DNA αποτελείται από δύο πολυνουκλεοτιδικές αλυσίδες, τους κλώνους, που σχηματίζουν δεξιόστροφη έλικα, ενώ οι αζωτούχες βάσεις σε κάθε κλώνο είναι κάθετες στον κύριο άξονα του μορίου και προεξέχουν προς το εσωτερικό του. Οι δύο κλώνοι συγκρατούνται μεταξύ τους με δεσμούς υδρογόνου που σχηματίζονται μεταξύ των αζωτούχων βάσεών τους. Οι αζωτούχες βάσεις ανάμεσα στις οποίες μπορούν να σχηματιστούν δεσμοί υδρογόνου είναι καθορισμένες και χαρακτηρίζονται συμπληρωματικές. Η αδενίνη σχηματίζει δύο δεσμούς υδρογόνου με τη θυμίνη, ενώ η γουανίνη σχηματίζει τρεις δεσμούς υδρογόνου με την κυτοσίνη (Σχήμα 2.2).



Σχήμα 2.2: Αζωτούχες Βάσεις

Αυτό σημαίνει πως στο δίκλωνο μόριο του DNA απέναντι από κάθε αδενίνη θα βρίσκεται πάντοτε μία θυμίνη και αντίστροφα, ενώ απέναντι από κάθε γουανίνη θα βρίσκεται μία κυτοσίνη και αντίστροφα (στο μονόκλωνο RNA η θυμίνη αντικαθίσταται από την ουρακίλη, Σχήμα 2.2).

Το χρησιμότατο συμπέρασμα που εξάγεται από τα παραπάνω, είναι ότι στις νουκλεοτιδικές αλυσίδες του DNA και του RNA αυτό που μεταβάλλεται είναι μόνο οι αζωτούχες βάσεις. Συνεπώς προκειμένου να αναπαραστήσουμε μία νουκλεοτιδική αλυσίδα αρκεί να την καταγράψουμε ως ακολουθία συμβόλων που αντιστοιχούν στις αζωτούχες βάσεις (A – αδενίνη, T – θυμίνη, G – γουανίνη, C – κυτοσίνη και U – ουρακίλη). Επίσης θα γνωρίζουμε πως η αλυσίδα αυτή (στο DNA) θα συνδέεται με μία άλλη, η οποία θα αποτελείται από τις συμπληρωματικές βάσεις της πρώτης (Σχήμα 2.3).



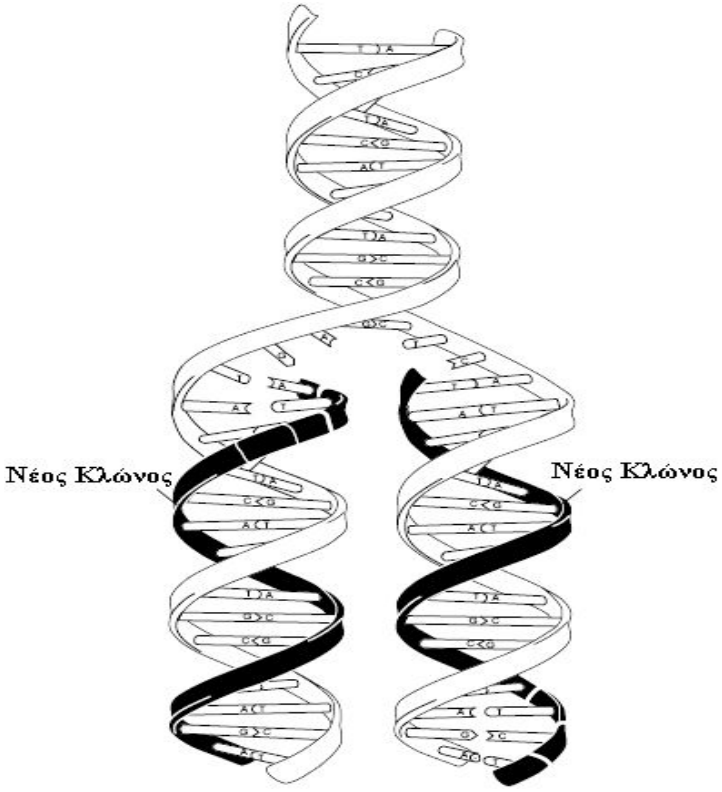
Σχήμα 2.3: Οι δύο κλώνοι του DNA

Μάλιστα οι δύο αλυσίδες έχουν αντίθετη φορά, καθώς η αρχή της μίας είναι απέναντι από το τέλος της άλλης και αποκαλούνται αντιπαράλληλες (και οι δύο έχουν κατεύθυνση από το 5' προς το 3' άκρο τους, Σχήμα 2.3). Η ίδια μέθοδος ισχύει και για την αναπαράσταση μίας νουκλεοτιδικής αλυσίδας ενός μορίου του RNA, το οποίο σε αντίθεση με το μόριο του DNA, είναι μονόκλωνο. Η μόνη διαφορά για μια ακολουθία συμβόλων RNA, είναι ότι αντί για το σύμβολο T (θυμίνη) υπάρχει το σύμβολο U (ουρακίλη).

Οι σημαντικότερες διαδικασίες που παρατηρούνται στο DNA είναι η αντιγραφή, μεταγραφή, μετάφραση και μετάλλαξη.

- **Αντιγραφή (Replication)**

Στη διαδικασία αυτή οι δύο αλυσίδες του DNA ανοίγουν, με αποτέλεσμα να μένουν ελεύθερες οι βάσεις των νουκλεοτιδίων. Τότε κατάλληλα ένζυμα φέρνουν απέναντι από κάθε βάση τη συμπληρωματική της, συνθέτοντας μία νέα αλυσίδα. Έτσι κάθε παλιά αλυσίδα χρησιμεύει σαν χημικό καλούπι για την κατασκευή ενός νέου μορίου, απολύτως όμοιου με το παλιό. Κάθε νέο μόριο αποτελείται από μία αλυσίδα του προηγούμενου μορίου και μία νέα αλυσίδα (ο τρόπος αυτός αυτοδιπλασιασμού ονομάζεται ημισυντηρητικός). Η διαδικασία λοιπόν της αντιγραφής έχει ως αποτέλεσμα τη δημιουργία δύο νέων μορίων, απολύτως ομοίων και μεταξύ τους και με το παλιό μόριο (Σχήμα 2.4).



Σχήμα 2.4: Αντιγραφή του DNA

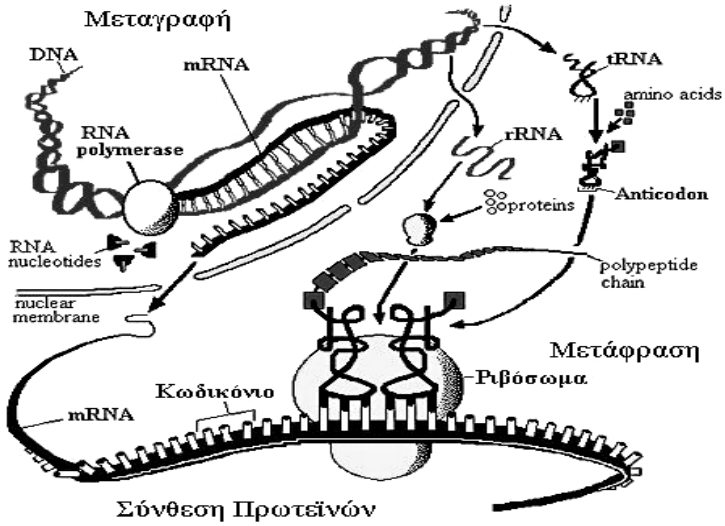
- **Μεταγραφή (Transcription)**

Η μεταγραφή είναι η διαδικασία σύνθεσης του RNA από το DNA. Όπως και στην αντιγραφή οι δύο αλυσίδες του DNA ανοίγουν, αλλά μόνο στην περιοχή που πρόκειται να μεταγραφεί και ονομάζεται γονίδιο. Τότε η μία εκ των δύο αλυσίδων χρησιμοποιείται σαν καλούπι για τη σύνθεση ενός μορίου RNA, καθώς απέναντι από την περιοχή που μεταγράφεται προσκολλώνται οι συμπληρωματικές βάσεις ($A \to U, C \leftrightarrow G, T \to A$). Μόλις ολοκληρωθεί η σύνθεση το RNA αποχωρίζεται από το DNA και το τελευταίο ξαναγίνεται δίκλωνο, αφού οι χωρισμένοι σε μία συγκεκριμένη περιοχή κλώνοι, επανενώνονται. Όλα τα είδη μορίων RNA δημιουργούνται με καλούπι το DNA. Αυτό όμως που μας ενδιαφέρει περισσότερο, είναι το αγγελιοφόρο μόριο RNA, mRNA, που είναι και αυτό που φέρει την αγγελία-μήνυμα για τη σύνθεση πρωτεϊνών. Τα τρία άλλα είδη RNA που συμμετέχουν επίσης, έμμεσα ή άμεσα, στη διαδικασία σύνθεσης πρωτεϊνών (μετάφραση), είναι τα tRNA (μεταφορικό), rRNA (ριβοσωμικό) και snRna (μικρό πυρηνικό).

- **Μετάφραση (Translation)**

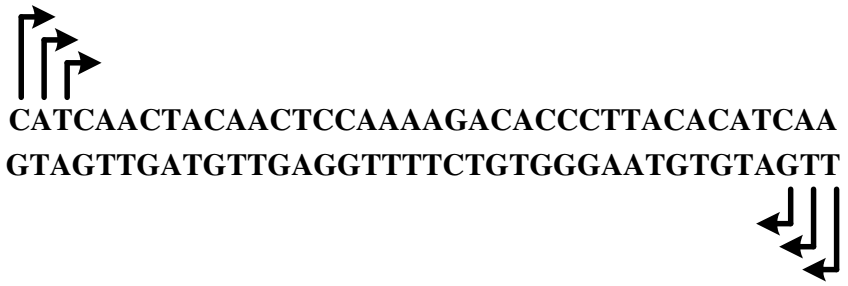
Η μετάφραση είναι η διαδικασία σύνθεσης των πρωτεϊνών που γίνεται με τη συμμετοχή τριών ειδών του RNA. Η συμμετοχή του DNA έγκειται κυρίως στη σύνθεση του mRNA το οποίο και κωδικοποιεί τη δημιουργία των αμινοξέων και κατά συνέπεια των πρωτεϊνών. Το mRNA αμέσως μετά τη σύνθεσή του μεταφέρεται στα ριβοσώματα (σχηματισμοί στη σύνθεση των οποίων υπάρχει το rRNA) του κυτταροπλάσματος. Στο ριβόσωμα έρχεται και το

tRNA, με το αντίστοιχο αμινοξύ, αναγνωρίζοντας ταυτοχρόνως την τριάδα νουκλεοτιδίων του mRNA που κωδικοποιεί το αμινοξύ αυτό. Το ριβόσωμα αλλάζει συνεχώς θέσεις στο mRNA και επομένως και τριπλέτες που αναγνωρίζονται από τα αντίστοιχα tRNA. Έτσι μπαίνουν σε σειρά τα αμινοξέα, σύμφωνα με τις οδηγίες που μεταφέρει το mRNA από τον πυρήνα, ενώνονται μεταξύ τους και σχηματίζουν τις πρωτεΐνες.



Σχήμα 2.5: Σύνθεση Πρωτεϊνών

Οι τριάδες νουκλεοτιδίων του mRNA που κωδικοποιούν αμινοξέα ονομάζονται κωδικόνια. Κωδικόνια ονομάζονται επίσης και οι τριάδες νουκλεοτιδίων της ακολουθίας DNA από την οποία δημιουργήθηκε το mRNA. Ένα μόριο DNA μπορεί θεωρητικά να συνθέσει έξι διαφορετικές ακολουθίες αμινοξέων, αναλόγως με το νουκλεοτίδιο από το οποίο αρχίζει η μετάφραση, και λέμε ότι έχει έξι πλαίσια ανάγνωσης (reading frames).



Σχήμα 2. 6: Πλαίσια ανάγνωσης

Όπως φαίνεται και στο Σχήμα 2.6, κάθε κλώνος του DNA έχει τρία διαφορετικά πλαίσια ανάγνωσης ανάλογα με το αν η μετάφραση ξεκινήσει από το πρώτο, δεύτερο ή τρίτο νουκλεοτίδιο, άρα το δίκλωνο μόριο έχει συνολικά έξι πλαίσια ανάγνωσης.

Πρέπει να σημειωθεί πως λέμε ότι μία ακολουθία DNA κωδικοποιεί θεωρητικά τρεις ακολουθίες αμινοξέων, άσχετα με το εάν η σύνθεση αυτών των ακολουθιών είναι όντως πραγματοποιήσιμη. Προκειμένου μία ακολουθία νουκλεοτιδίων να κωδικοποιεί όντως τη δημιουργία πρωτεΐνης, πρέπει να υπάρχει σε κάποιο σημείο το κωδικόνιο έναρξης (AUG στο

mRNA ή ATG στο DNA) που αντιστοιχεί στο αμινοξύ μεθειονίνη, και σε κάποιο άλλο σημείο στη συνέχεια, να υπάρχει ένα κωδικόνιο λήξης, οπότε και η διαδικασία αποκαλείται μετάφραση.

Μάλιστα οι τρεις αυτές διαδικασίες συνθέτουν το κεντρικό δόγμα της βιολογίας το οποίο αποτελείται από τρία σκέλη: 1)Το DNA αντιγράφει τον εαυτό του, 2)Το DNA παράγει το RNA, 3)Το RNA οδηγεί στην κατασκευή πρωτεϊνών.

Επίσης πρέπει να αναφερθεί πως κάτω από ορισμένες προϋποθέσεις τα μόρια του RNA υφίστανται τη διαδικασία της αντίστροφης μεταγραφής και συνθέτουν μόρια DNA, οπότε το δεύτερο σκέλος του κεντρικού δόγματος μπορεί να γίνει: 2)Το DNA παράγει RNA και το RNA παράγει DNA.

- **Μετάλλαξη (Mutation)**

Μετάλλαξη ή μεταλλαγή ονομάζεται οποιαδήποτε αλλαγή στις βάσεις του DNA. Τέτοιες αλλαγές μπορεί να είναι η αντικατάσταση μιας βάσης από μία άλλη και η προσθήκη ή η αφαίρεση μίας βάσης. Στην πρώτη περίπτωση αλλάζει μία τριπλέτα στο DNA και συνεπώς αλλάζει και ένα αμινοξύ στην πρωτεΐνη. Αν προστεθεί ή αφαιρεθεί μία βάση τότε αλλάζουν όλες οι τριπλέτες από το σημείο αυτό και μετά, οπότε αλλάζουν και όλα τα αμινοξέα. Οι αλλαγές αυτές μπορεί να οφείλονται σε λάθη στην αντιγραφή του DNA, έκθεση σε ακτινοβολία, καθώς και σε χημικές ουσίες. Οι μεταλλάξεις συνήθως οδηγούν σε τροποποιημένες πρωτεΐνες, κάτι που μπορεί να αποβεί ολέθριο για τον οργανισμό. Χαρακτηριστικό παράδειγμα είναι η δρεπανοκυτταρική αναιμία, η οποία προέρχεται από την αντικατάσταση του αμινοξέος γλουταμίνη από το αμινοξύ βαλίνη, λόγω της αντικατάστασης μίας και μόνο βάσης σε μία τριπλέτα στο DNA.

Πρέπει να σημειωθεί πως οι παραπάνω βιολογικές διαδικασίες περιγράφησαν απλοποιημένες, καθώς σε αυτές συμμετέχουν και άλλα μόρια, όπως τα ένζυμα (που είναι και αυτά πρωτεΐνες) ο ρόλος των οποίων είναι καταλυτικός στις εντός κυττάρου διεργασίες, αλλά και άλλα συστατικά του κυττάρου. Μία αλυσίδα μοριακών διεργασιών στις οποίες συμμετέχουν ένζυμα ορίζεται ως μεταβολικό μονοπάτι.

2.4 *Αμινοξέα και Πρωτεΐνες*

Τα αμινοξέα είναι ενώσεις που αποτελούνται από ένα μόριο άνθρακα ενωμένο με ένα άτομο υδρογόνου, μία καρβοξυλική ομάδα, μία αμινομάδα και μία πλευρική ομάδα R. Η χημική σύσταση της τελευταίας είναι αυτή που ποικίλλει στα διάφορα αμινοξέα. Στα κύτταρα έχουν ανιχνευθεί εκατόν εβδομήντα διαφορετικά αμινοξέα, εκ των οποίων μόνον είκοσι αποτελούν συνήθως συστατικά των πρωτεϊνών και απεικονίζονται στον Πίνακα 2.1 (Πρόσφατες έρευνες έχουν δείξει πως περισσότερα των είκοσι αμινοξέα συμμετέχουν στη σύσταση των πρωτεϊνών).

Συντόμευση		Όνομα
Ala	A	Alanine-Αλανίνη
Arg	R	Arginine-Αργινίνη
Asn	N	Asparagine-Ασπαραγγίνη
Asp	D	Aspartic Acid (Aspartate) –Ασπαρτάτη
Cys	C	Cysteine – Κυστεΐνη
Gln	Q	Glutamine – Γλουταμίνη
Glu	E	Glutamic acid (Glutamate) – Γλουταμάτη
Gly	G	Glycine – Γλυκίνη
His	H	Histidine - Ιστιδίνη
Ile	I	Isoleucine - Ισολευκίνη
Leu	L	Leucine – Λευκίνη
Lys	K	Lysine – Λυσίνη
Met	M	Methionine – Μεθειονίνη
Phe	F	Phenylanine – Φαινυλανίνη
Pro	P	Proline – Προλίνη
Ser	S	Serine – Σερίνη
Thr	T	Threonine – Θρεονίνη
Trp	W	Tryptophan – Τρυπτοφάνη
Tyr	Y	Tyrosine – Τυροσίνη
Val	V	Valine – Βαλίνη
Asx	B	Aspartic acid (Asparagine) – Ασπαραγγίνη
Glx	Z	Glutamine or Glutamic acid
Xaa	X	Any amino acid – Οποιοδήποτε αμινοξύ

Πίνακας 2.1: Αμινοξέα που συμμετέχουν στη σύνθεση πρωτεϊνών

Τα αμινοξέα ενώνονται μεταξύ τους με ένα δεσμό που ονομάζεται πεπτιδικός, σχηματίζοντας έτσι τις πολυπεπτιδικές αλυσίδες. Οι πρωτεΐνες αποτελούνται από μία ή περισσότερες τέτοιες αλυσίδες και είναι τα πλέον διαδεδομένα και πολυδιάστατα μόρια. Ακόμη και σε ένα απλό κύτταρο όπως αυτό των βακτηρίων υπάρχουν εκατοντάδες διαφορετικές πρωτεΐνες, καθεμία από τις οποίες έχει έναν ιδιαίτερο ρόλο στη ζωή του κυττάρου, ενώ στο ανθρώπινο σώμα υπάρχουν περισσότερες από τριάντα χιλιάδες διαφορετικές πρωτεΐνες. Ένα μεγάλο μέρος των πρωτεϊνών είναι ένζυμα, δηλαδή βιολογικοί καταλύτες που ρυθμίζουν τις ταχύτητες, με τις οποίες συμβαίνουν οι χημικές διαδικασίες του κυττάρου.

Μία τυπική πρωτεΐνη, πολύ μεγάλης σημασίας για το ανθρώπινο σώμα, είναι η αιμοσφαιρίνη, η ερυθρά πρωτεΐνη που ευθύνεται για το χρώμα του αίματος. Για την πολυπλοκότητα του μορίου της αιμοσφαιρίνης μπορούμε να πάρουμε μία ιδέα από το χημικό της τύπο, ο οποίος είναι: $C_{3032}H_{4816}O_{872}N_{780}S_8Fe_4$, όπου C, H, O, N, S, Fe τα άτομα του Άνθρακα, Υδρογόνου, Οξυγόνου, Αζώτου, Θείου και Σιδήρου αντιστοίχως. Από τον χημικό τύπο της αιμοσφαιρίνης κάποιος θα μπορούσε να υποθέσει πως αυτή είναι μία μεγάλη πρωτεΐνη, ενώ στην πραγματικότητα είναι μετρίου μεγέθους.

Εφόσον λοιπόν τα νουκλεοτίδια είναι τέσσερα και τα αμινοξέα είκοσι, πρέπει συνδυασμοί των βάσεων ανά τρεις να κωδικοποιούν τα αμινοξέα. Με τον τρόπο αυτόν δημιουργούνται εξήντα τέσσερις διαφορετικοί συνδυασμοί ($4^3 = 64$) οι οποίοι επαρκούν και περισσεύουν για τα είκοσι αμινοξέα. Αποτέλεσμα αυτού, είναι ορισμένες τριάδες νουκλεοτιδίων να αντιστοιχούν στο ίδιο αμινοξύ. Η αντιστοιχία αυτή κωδικονίων και αμινοξέων συνθέτουν το γενετικό κώδικα, ο οποίος απεικονίζεται στον Πίνακα 2.2.

		Second Position of Codon					
		T	C	A	G		
First Position	T	TTT Phe [F] TTC Phe [F] TTA Leu [L] TTG Leu [L]	TCT Ser [S] TCC Ser [S] TCA Ser [S] TCG Ser [S]	TAT Tyr [Y] TAC Tyr [Y] TAA *Ter* [end] TAG *Ter* [end]	TGT Cys [C] TGC Cys [C] TGA *Ter* [end] TGG Trp [W]	T C A G	Third Position
	C	CTT Leu [L] CTC Leu [L] CTA Leu [L] CTG Leu [L]	CCT Pro [P] CCC Pro [P] CCA Pro [P] CCG Pro [P]	CAT His [H] CAC His [H] CAA Gln [Q] CAG Gln [Q]	CGT Arg [R] CGC Arg [R] CGA Arg [R] CGG Arg [R]	T C A G	
	A	ATT Ile [I] ATC Ile [I] ATA Ile [I] ATG Met [M]	ACT Thr [T] ACC Thr [T] ACA Thr [T] ACG Thr [T]	AAT Asn [N] AAC Asn [N] AAA Lys [K] AAG Lys [K]	AGT Ser [S] AGC Ser [S] AGA Arg [R] AGG Arg [R]	T C A G	
	G	GTT Val [V] GTC Val [V] GTA Val [V] GTG Val [V]	GCT Ala [A] GCC Ala [A] GCA Ala [A] GCG Ala [A]	GAT Asp [D] GAC Asp [D] GAA Glu [E] GAG Glu [E]	GGT Gly [G] GGC Gly [G] GGA Gly [G] GGG Gly [G]	T C A G	

Πίνακας 2.2: Γενετικός Κώδικας

Ο Γενετικός Κώδικας χαρακτηρίζεται από τις παρακάτω ιδιότητες:

- Είναι τριαδικός, δηλαδή τρεις βάσεις κωδικοποιούν ένα αμινοξύ.
- Είναι συνεχής, δηλαδή το mRNA διαβάζεται συνεχώς ανά τρία νουκλεοτίδια, χωρίς να παραλείπεται κάποιο νουκλεοτίδιο.
- Είναι μη επικαλυπτόμενος, δηλαδή κάθε βάση αποτελεί μέρος ενός μόνο κωδικονίου.
- Είναι σχεδόν καθολικός. Σχεδόν όλοι οι οργανισμοί έχουν τον ίδιο γενετικό κώδικα, κι έτσι το mRNA από οποιονδήποτε οργανισμό θα δώσει τα ίδια αποτελέσματα πρωτεϊνοσύνθεσης.

- Είναι εκφυλισμένος. Με δύο μόνο εξαιρέσεις αμινοξέων που κωδικοποιούνται από ένα κωδικόνιο το καθένα, κάθε άλλο αμινοξύ κωδικοποιείται από περισσότερα του ενός κωδικόνια και τότε λέμε πως υπάρχει εκφυλισμός του κώδικα.
- Έχει σινιάλα έναρξης και λήξης. Το AUG (ή ATG) που κωδικοποιεί τη μεθειονίνη, είναι το κωδικόνιο έναρξης για όλες τις πρωτεΐνες. Τα τρία κωδικόνια λήξης UAA, UAG και UGA σημαίνουν τη λήξη της πρωτεϊνοσύνθεσης και δεν αντιστοιχούν σε κάποιο αμινοξύ (αποκαλούνται και κωδικόνια χωρίς νόημα).

2.5 *Βιοεπιστήμες*

Τα βιολογικά αυτά δεδομένα είναι από τα κύρια αντικείμενα μελέτης των Βιοεπιστημών. Με τον όρο Βιοεπιστήμες εννοούμε τη μελέτη της δομής της βιολογικής πληροφορίας και των βιολογικών συτημάτων, που συνδυάζει την έρευνα στη Βιολογία (κυρίως στη Μοριακή Βιολογία) με τις επιστήμες των υπολογιστών. Ο επιστημονικός κλάδος αυτός προήλθε από την ανάγκη για χρησιμοποίηση, αποθήκευση και καλύτερη κατανόηση τεράστιων ποσοτήτων δεδομένων που συνεχώς ανακαλύπτονται από τις έρευνες των βιολόγων.

Το ζήτημα που τίθεται στις Βιοεπιστήμες είναι το πώς τα βιολογικά αυτά δεδομένα θα μεταφερθούν από το χώρο της βιολογίας στο χώρο των υπολογιστών, με όσον το δυνατό πιο αξιόπιστο τρόπο. Ενδεικτικά αναφέρουμε δύο εργαστηριακές μεθόδους που χρησιμοποιούνται για το σκοπό αυτόν:

- Οι ακολουθοποιητές (sequencers) είναι μηχανές ικανές να διαβάσουν ακολουθίες νουκλεοτιδίων από τα μόρια του DNA σε βιολογικά δείγματα. Οι μηχανές αυτές είναι συνδεδεμένες με υπολογιστές που απεικονίζουν το DNA που αναλύεται. Η απεικόνιση αυτή δείχνει επίσης το βαθμό αξιοπιστίας της ταυτοποίησης του κάθε νουκλεοτιδίου. Οι τωρινοί ακολουθιοποιητές μπορούν να παράγουν περισσότερα των τριακοσίων χιλιάδων ζευγών βάσεων, με σχετικώς λογικά κόστη.
- Μία εκ των σημαντικοτέρων δυσκολιών στην καταγραφή του γονιδιώματος ενός οργανισμού, είναι το γεγονός ότι οι ακολουθίες που συλλέγονται στο βιολογικό εργαστήριο αποτελούνται από μικρού μήκους και τυχαίως διασκορπισμένα τμήματα-θραύσματα, τα οποία πρέπει να επανασυγκεντρωθούν με τη χρήση προγραμμάτων υπολογιστών. Η μέθοδος αυτή ακολουθιοποίησης ονομάζεται μέθοδος του πυροβόλου όπλου (shotgun method of sequencing) και συναντά δυσκολίες καθώς το DNA περιλαμβάνει επαναλαμβανόμενες υπακολουθίες και συνεπώς η τοποθέτηση των τμημάτων δεν μπορεί να είναι μονοσήμαντη (ένα τμήμα μπορεί να τοποθετηθεί σε δύο ή περισσότερες θέσεις του γονιδιώματος του οργανισμού).

2.6 *Πρότυπα αναπαράστασης δεδομένων Βιοεπιστημών*

Τα δεδομένα των Βιοεπιστημών είναι αρκετά δύσκολο να μοντελοποιηθούν, και εξαιτίας αυτής της δυσκολίας τα συστήματα μοντελοποίησης πρέπει να είναι εύκολα διαμορφώσιμα. Η δυσκολία αυτή οφείλεται εν μέρει στη μεγάλη ποικιλία διαφόρων τύπων δεδομένων καθώς και στις αλληλεξαρτήσεις των τύπων αυτών. Επιπροσθέτως, νέοι τύποι δεδομένων εμφανίζονται ανά τακτά χρονικά διαστήματα, τροποποιώντας τις γνώσεις και αντιλήψεις μας για τους παλαιούς τύπους. Συνεπώς είναι απαραίτητη καθιέρωση ενός κοινού συστήματος καταγραφής δεδομένων, με ιεραρχική δομή, όπως η γλώσσα XML.

Λόγω της ιεραρχικής φύσεως της XML είναι δυνατή, μέσω αυτής, η ανταλλαγή πληροφοριών μεταξύ ερευνητών, διαφορετικών επιστημονικών υποβάθρων, με στόχο την ανάπτυξη της γνώσης. Η XML είναι ιδανική για συστήματα διαχείρισης γνώσης (knowledge management systems), καθώς με την ιεραρχική δομή της με ετικέτες (hierarchical tag structure), συνδυάζει τα δεδομένα με επιπλέον πληροφορίες για αυτά (μεταδεδομένα – metadata).

Είναι το πρώτο ευρέως υιοθετημένο πρότυπο (standard) έκφρασης και μοντελοποίησης πληροφοριών και γίνεται σταδιακά αποδεκτή στο χώρο των Βιοεπιστημών. Διάφορες επιστημονικές ομάδες του χώρου αυτού έχουν δημιουργήσει γλώσσες τύπου XML, σημαντικότερες των οποίων είναι οι:

- BIOML (BIOpolymer Markup Language)

Η BIOML αναπτύχθηκε από τις επιστημονικές ομάδες των Proteometrics LLC και Proteometrics Canada Ltd και χρησιμοποιείται για την περιγραφή πειραματικών πληροφοριών που αφορούν πρωτεΐνες, γονίδια και άλλα βιοπολυμερή (μόρια που σχηματίζονται από την ένωση μικροτέρων μορίων). Ένα αρχείο της BIOML περιγράφει ένα φυσικό αντικείμενο (για παράδειγμα μία πρωτεΐνη) με τρόπο τέτοιον, ώστε όλες οι γνωστές πληροφορίες πειραματικού επιπέδου να συσχετίζονται λογικά και σημασιολογικά με το αντικείμενο. Οι πληροφορίες είναι φωλιασμένες σε διαφορετικά επίπεδα σε μορφή δένδρου-φύλλων (όπως και στην XML). [Bml07]

- BSML (Bioinformatic Sequence Markup Language)

Το Εθνικό Ινστιτούτο Έρευνας Ανθρώπινου Γονιδιώματος (National Human Genome Research Institute-NHGRI) χρηματοδότησε το 1997 την ανάπτυξη της BSML ως ένα πρότυπο κοινού τομέα (public domain standard) για την κοινότητα των Βιοεπιστημών. Μεταξύ των πρώτων στόχων της BSML ήταν η δημιουργία ενός μοντέλου αναπαράστασης δεδομένων ακολουθιών. Το μοντέλο που δημιουργήθηκε τελικά, περιέχει, εκτός από τις ίδιες τις ακολουθίες, σχόλια, περιγραφές καθώς και συνδέσμους (links) με άλλες ακολουθίες και διαδικτυακούς τόπους. Έτσι με την BSML μπορούν να περιγραφούν φαινόμενα που έχουν

σχέση με ακολουθίες, από το βιομοριακό επίπεδο έως το επίπεδο του πλήρους γονιδιώματος. Η εταιρεία LabBook έχει τα δικαιώματα συγγραφής και ιδιοκτησίας της BSML. [Bsm07]

- PSDML (Protein Sequence Database Markup Language)

Η βάση δεδομένων πρωτεϊνικών πληροφοριών και πόρων PIR (Protein Information Resource database) προέκυψε από τη συνεργασία μεταξύ του Εθνικού Ιδρύματος Βιοϊατρικής Έρευνας στο Ιατρικό κέντρο του Πανεπιστήμιου του Georgetown, του κέντρου πληροφοριών πρωτεϊνικών ακολουθιών στο Μόναχο και της διεθνούς βάσης δεδομένων πρωτεϊνικών πληροφοριών της Ιαπωνίας. Η PIR είναι εμπλουτισμένη με σχόλια και επιτρέπει αναζήτηση πληροφοριών με κριτήρια όπως η ομοιότητα ακολουθιών ή το ταίριασμα κειμένου. Η PSDML λοιπόν, είναι η γλώσσα XML που χρησιμοποιείται για την αποθήκευση πρωτεϊνικών πληροφοριών στη βάση δεδομένων PIR. [Pir06]

- GAME (Genome Annotation Markup Language)

Η GAME μπορεί να χρησιμοποιηθεί για την αναπαράσταση περιγραφών ή σχολίων που αφορούν συγκεκριμένες περιοχές πρωτεϊνών, αλλά και γονιδίων. Αυτά τα σχόλια και οι περιγραφές προκύπτουν είτε από προγράμματα ανάλυσης ακολουθιών είτε από εργαστηριακά αποτελέσματα και μπορούν να διαφοροποιηθούν μέσω της GAME. Η διευκόλυνση της ανταλλαγής σχολίων και περιγραφών μεταξύ ερευνητών, ερευνητικών κέντρων και βάσεων δεδομένων γίνεται με την καταγραφή αυτών των δεδομένων σε μορφή XML και έχει ως στόχο την καλύτερη εξαγωγή επιστημονικών συμπερασμάτων. Η πρώτη ευρέως χρησιμοποιούμενη έκδοση της GAME δημιουργήθηκε στο BDGP (Berkeley Drosophila Genome Project). [Gam07]

- SBML (Systems Biology Markup Language)

Το σχέδιο έργου βιολογικών συστημάτων SBW (Systems Biology Workbench) στο Ίδρυμα Τεχνολογίας της Καλιφόρνιας (California Institute of Technology) έχει ως στόχο τη δημιουργία μοντέλων αναπαράστασης εργαλείων ανάλυσης και προσομοίωσης βιολογικών συστημάτων. Μία εκ των δύο προσεγγίσεων που επιδιώκουν την επίτευξη αυτού του στόχου ήταν η σταδιακή ανάπτυξη της SBML (Systems Biology Markup Language), η οποία είναι μία τύπου XML αναπαράσταση διαδικτυακών βιοχημικών μοντέλων. [Sbm07]

- CellML

Η CellML είναι μία γλώσσα XML, σκοπός της οποίας είναι η αποθήκευση και η ανταλλαγή υπολογιστικών βιολογικών μοντέλων. Η CellML αναπτύχθηκε πρωταρχικά από την Physiome Sciences του Princeton, στο New Jersey και το ίδρυμα Βιο-μηχανικής (Bioengineering) του Πανεπιστημίου του Auckland. Η CellML επιτρέπει τον συμμερισμό μοντέλων ή συστατικών αυτών, ακόμα και αν είναι κατασκευασμένα με διαφορετικό λογισμικό. Αυτή η δυνατότητα επιταχύνει τη διαδικασία κατασκευής μοντέλων. [Cml07]

- MAGE-ML (Microarray Gene Expression Markup Language)

Η MAGE-ML προήλθε αυτομάτως από το MAGE-OM (Microarray Gene Expression Object Model) και είναι μία γλώσσα τύπου XML, σχεδιασμένη για την περιγραφή και μετάδοση πληροφοριών πειραμάτων σε μικροπίνακες (microarrays-πειράματα που περιγράφουν την έκφραση των γονιδίων στα κύτταρα). Οι πληροφορίες αυτές μπορεί να αφορούν τον σχεδιασμό μικροπινάκων, την προετοιμασία και εκτέλεση πειραμάτων μικροπινάκων, δεδομένα γονιδιακής έκφρασης και αποτελέσματα ανάλυσης δεδομένων. Η MAGE-ML αντικατέστησε την MAML (Microarray Markup Language) από τον Φεβρουάριο του 2002. [Mag07]

2.7 *Παραδείγματα Αναπαράστασης Βιολογικής Πληροφορίας*

Στη συνέχεια παραθέτουμε ως παράδειγμα ένα απόσπασμα της CellML περιγραφής από την Catherine Lloyd για το βιολογικό μοντέλο των Novak και Tyson που αναπαριστά τον έλεγχο της αντιγραφής του DNA της μαγιάς (yeast).

```
<?xml version="1.0"?>
<!-- FILE : novak_model_1997.xml
CREATED : 22nd April 2003
LAST MODIFIED : 22nd April 2003
AUTHOR : Catherine Lloyd
Bioengineering Institute
The University of Auckland
MODEL STATUS : This model conforms to the CellML 1.0 Specification
released on
10th August 2001, and the 16/1/02 CellML Metadata 1.0 Specification.
DESCRIPTION : This file contains a CellML description of Novak and
Tyson's
1997 model of the control of DNA replication in fission yeast.
CHANGES:
-->
<model name="novak_model_1997" cmeta:id="novak_model_1997"
xmlns="http://www.cellml.org/cellml/
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:bqs="http://www.cellml.
     <!--
     The following RDF block contains metadata that applies to this
      document as a whole, as indicated by the empty about attribute
      on the <rdf:Description> element.
      -->
    <rdf:Description rdf:about="">
      <!--
       The Model Builder Metadata. The Dublin Core "creator" element
       is used to indicate the person who translated the model into
       CellML.
      -->
      <dc:creator rdf:parseType="Resource">
        <vCard:N rdf:parseType="Resource">
          <vCard:Family>Lloyd</vCard:Family>
          <vCard:Given>Catherine</vCard:Given>
          <vCard:Other>May</vCard:Other>
        </vCard:N>
        <vCard:EMAIL rdf:parseType="Resource">
```

```
          <rdf:value>c.lloyd@auckland.ac.nz</rdf:value>
          <rdf:type
           rdf:resource="http://imc.org/vCard/3.0#internet"/>
        </vCard:EMAIL>
        <vCard:ORG rdf:parseType="Resource">
          <vCard:Orgname>The University of Auckland</vCard:Orgname>
          <vCard:Orgunit>The Bioengineering Institute</vCard:Orgunit>
        </vCard:ORG>
      </dc:creator>
```

… Το περιεχόμενο του αρχείου έχει παραλειφθεί σε αυτό το σημείο

```
<connection>
  <map_components component_1="G1K" component_2="G1R" />
  <map_variables variable_1="G1R" variable_2="G1R" />
  <map_variables variable_1="G1K" variable_2="G1K" />
</connection>

<connection>
  <map_components component_1="PG2" component_2="PG2R" />
  <map_variables variable_1="PG2R" variable_2="PG2R" />
  <map_variables variable_1="PG2" variable_2="PG2" />
</connection>

<connection>
  <map_components component_1="UbE" component_2="IE" />
  <map_variables variable_1="IE" variable_2="IE" />
</connection>

</model>
```

Στο αρχείο αυτό εκτός από την ίδια την αναπαράσταση του μοντέλου των Novak και Tyson παρατίθενται και πληροφορίες όπως η ημερομηνία συγγραφής και τελευταίας ενημέρωσης, τα ονόματα των συγγραφέων και οι τρόποι επικοινωνίας με αυτούς, η σχετική βιβλιογραφία και δημοσιεύσεις κ.α.

Επίσης ένα παράδειγμα της χρήσης XML για αναπαράσταση μιας νουκλεοτιδικής ακολουθίας που έχει ληφθεί από μία καταχώρηση DNA ακολουθίας του NCBI, απεικονίζεται παρακάτω.

```
<dna_sequence_entry>
  <header>
     <definition>
        Human Cu/Zn superoxide dismutase (SOD1) gene
     </definition>
     <accession_no>
          <base_no>L44135<</base_no>
          <version>L44135.1</version>
          <GI>1237400</GI>
     </accesion_no>
     <keyword>Cu/Zn superoxide dismutase</keyword>
     <keyword>Human SOD1 gene</keyword>
  </header>
  <source>
       <name>Human Cu/Zn superoxide dismutase (SOD1) gene,
exon1.</name>
       <organism>Homo sapiens</organism>
     <taxonomy>
         <cell_type>Eukaryota<cell_type>
         <kingdom>Metazoa</kingdom>
         <phylum>Chordata</phylum>
```

```
          <subphylum>Vertebrata</subphylum>
          <class>Mammalia</class>
          <infraclass>Eutheria</infraclass>
          <order>Primates</order>
          <family>Homonidae</family>
          <genus>Homo</genus>
          <species>sapiens</species>
      </taxonomy>
  </source>
   <reference>
        <author> Levanon,D. </author>
        <author> Lieman-Hurwitz,J </author>
        <title> Architecture and anatomy of the chromosomal locus in
human chromosome 21
         encoding the Cu/Zn superoxide dismutase </title>
        <journal> EMBO J. 4 (1), 77-84 (1985) </journal>
   </reference>
  <dna_sequence>
gtaccctgtttacatcattttgccattttcgcgtactgcaaccggcgggccacgccgtgaaaagaaggt
tgttttctccacagtttcggggttctggacgtttcccggctgcggggcgggggggagtctccggcgcacg
cggcccctggcccgccccagtcattcccggccactcgcgacccgaggctgccgcagggggcgggctga
gcgcgtgcgaggccattggtttggggcc…To αρχείο έχει παραλειφθεί σε αυτό το σημείο
  </dna_sequence>
   <features>
      <protein>
          <type>CDS</type>
          <location>1..799</location>
          <gene>SOD1</gene>
          <codon_start>1</codon_start>
          <chromosome>21</chromosome>
          <map>21q22.1</map>
          <product>Cu/Zn-superoxide dismutase</product>
          <protein_id>AAB05661.1</protein_id>
          <db_xref>GI:1237407</db_xref>
          <amino_acid_sequence>
MATKAVCVLKGDGPVQGIINFEQKESNGPVKVWGSIKGLTEGLHFHVHEFGDNTAGCTSAGPHFNPLSR
KHGGPKDEERHVGDL… To αρχείο έχει παραλειφθεί σε αυτό το σημείο
          </amino_acid_sequence>
      </protein>
   <features>
</dna_sequence_entry>
```

Παρατηρούμε πως στο αρχείο αυτό εμφανίζονται πληροφορίες όπως το όνομα του γονιδίου, ο αριθμός πρόσβασής του στη βάση δεδομένων, λέξεις κλειδιά που διευκολύνουν την αναζήτηση (<keyword>), η ταξονομία (ιεραρχική ταξινόμηση) των ειδών, αλλά και πληροφορίες για, σχετικά με την ακολουθία, άρθρα σε επιστημονικά περιοδικά. Επίσης παρατίθεται η πλήρης νουκλεοτιδική ακολουθία καθώς και πληροφορίες για την πρωτεΐνη που αυτή κωδικοποιεί. Παρόλα αυτά όσο περισσότερα μαθαίνει κάποιος για τις βιολογικές διαδικασίες, τόσο περισσότερες πληροφορίες θα ήθελε να προσθέσει σε ένα τέτοιο αρχείο. Παραδείγματα των πιθανών επιπρόσθετων πληροφοριών ακολουθούν:

- Τα γονίδια έχουν τμήματα που κωδικοποιούν πρωτεΐνες (τα εξώνια) και τμήματα που δεν κωδικοποιούν πρωτεΐνες (τα εσώνια). Μία πληροφορία λοιπόν που θα θέλαμε στην XML

αναπαράσταση θα μπορούσε να είναι ο διαχωρισμός και η ταυτοποίηση αυτών των περιοχών.

- Ένα γονίδιο μπορεί να κωδικοποιεί περισσότερες της μίας πρωτεΐνες οι οποίες θα έπρεπε να αναφερθούν, μαζί με τις βιολογικές διαδικασίες που αυτές οι πρωτεΐνες επηρεάζουν.
- Η εξωτερική έκφραση ενός γονιδίου (φαινότυπος) σε δεδομένο οργανισμό είναι εξαιρετικά χρήσιμη πληροφορία. Ο φαινότυπος αυτός μπορεί για παράδειγμα να είναι το χρώμα ματιών, το χρώμα μαλλιών, ο αριθμός των δακτύλων κλπ.
- Ένα γονίδιο μπορεί επίσης να συσχετίζεται με αριθμό ασθενειών, όπως η κυστική ίνωση, η τύφλωση, είδη καρκίνου και άλλες. Οι ασθένειες αυτές μπορεί να οφείλονται σε μία απλή αλλαγή αζωτούχου βάσεως σε κάποιες περιοχές του γονιδίου (μετάλλαξη), οι οποίες και πρέπει να καταγράφονται.

Με βάση τα παραπάνω και δεδομένου ότι η έρευνα στο μικροβιολογικό επίπεδο είναι εν εξελίξει, είναι δύσκολο να προβλεφθούν οι πληροφορίες που ένας ερευνητής θα μπορούσε να επισυνάψει για μία ακολουθία. Γι' αυτό το λόγο πρέπει το μοντέλο πληροφορίας που χρησιμοποιούμε να είναι ιεραρχικής δομής και εύκολα διαμορφώσιμο. Η ιεραρχική δομή χρησιμοποιείται για να δειχθούν οι σχέσεις εντός του μοντέλου. Για παράδειγμα μία πρωτεΐνη περιέχεται μέσα στο στοιχείο (element) ενός γονιδίου. Αν το ίδιο γονίδιο κωδικοποιεί πολλές πρωτεΐνες, τότε αυτές θα είναι μέσα στο στοιχείο του γονιδίου, ενώ εάν υπάρχουν πολλά γονίδια, τότε θα υπάρχουν πολλά στοιχεία γονιδίων με το καθένα να περιέχει ως στοιχεία τις πρωτεΐνες που κωδικοποιεί.

Επίσης πρέπει να τονιστεί πως οι παραδοσιακές βάσεις δεδομένων είναι σχεδιασμένες ώστε να διαχειρίζονται δεδομένα, δημιουργώντας ένα στατικό πλαίσιο που θα περιέχει δυναμικά δεδομένα. Με αυτήν την πρόταση εννοούμε πως τα στοιχεία δεδομένων μπορούν να διαχειριστούν, εφόσον όλα τα μεταδεδομένα (δεδομένα για τα δεδομένα) έχουν οριστεί εκ των προτέρων. Αυτή η μεθοδολογία όμως αδυνατεί να χειριστεί ολόκληρη την πληροφορία. Η λύση σε αυτό το πρόβλημα είναι η διαχείριση των μεταδεδομένων με τον ίδιο δυναμικό τρόπο που τα ίδια τα δεδομένα διαχειρίζονται. Η λύση αυτή παρέχει δύο σημαντικά πλεονεκτήματα:

- Όλοι οι περιορισμοί στη χρήση δυναμικών τύπων δεδομένων αφαιρούνται. Νέοι τύποι δεδομένων μπορούν απλά να οριστούν και να προστεθούν στη βάση δεδομένων.
- Οι διαδικασίες του σχεδιασμού και διαμόρφωσης της βάσης δεδομένων γίνονται ταχύτερες και λιγότερο κοπιαστικές.

Τα πλεονεκτήματα αυτά αποτελούν χαρακτηριστικά του συστήματος διαχείρισης XML Neocore XMS, το οποίο χειρίζεται δεδομένα και μεταδεδομένα με τον ίδιο δυναμικό τρόπο. Το Neocore XMS σχεδιάστηκε με στόχο να επιτευχθούν οι ακόλουθοι στόχοι:

- Δυναμική διαχείριση των μεταδεδομένων

Όλη η πληροφορία αναπαρίσταται με μία εσωτερική δομή που ονομάζεται δίστιχα πληροφορίας (information couplets), τα οποία είναι ζεύγη δεδομένων και ολοκληρωμένων μεταδεδομένων. Τα δίστιχα πληροφορίας αντιμετωπίζονται ως μοτίβα, με αποτέλεσμα να μην υπάρχουν γραμμές ή στήλες και να μην είναι απαραίτητος ο προκαθορισμός δεικτών. Μία πληροφορία μπορεί να συσχετιστεί με ένα κομμάτι πληροφορίας, χωρίς να χρειάζεται να έχει οριστεί, τοποθετηθεί ή προστεθεί σε κάποιο άλλο κομμάτι πληροφορίας. Αυτό σημαίνει πως ένας εντελώς νέος τύπος δεδομένων μπορεί να προστεθεί οποιαδήποτε στιγμή σε μία εφαρμογή, χωρίς να χρειαστεί να κάνουμε κάτι στη βάση δεδομένων.

- Άμεση διαθεσιμότητα της πληροφορίας

Η πληροφορία δεικτοδοτείται αμέσως μετά την καταχώρησή της. Στην πραγματικότητα δεν υπάρχει ξεχωριστή διαδικασία δεικτοδότησης καθώς δεν χρειάζεται να καθοριστεί το τι πρέπει να δεικτοδοτηθεί. Η Neocore XMS αυτομάτως δημιουργεί μοτίβα πληροφορίας (data patterns) που επιτρέπουν την ανάκτηση πληροφορίας βασισμένοι σε ερωτήσεις. Πολύπλοκες ερωτήσεις (queries) τοποθετούνται σύμφωνα με έναν αλγόριθμο σύγκλισης ιεραρχικών διανυσμάτων (hierachical vector convergence algorithm), ο οποίος συγκλίνει σύνολα ταιριάσματος μοναδικών μοτίβων (sets of individual pattern matches) για να εντοπίσει τμήματα πληροφορίας βασιζόμενος σε πολλαπλά κριτήρια. Η διαδικασία σύγκλισης διανύσματος (vector convergence) εφαρμόζεται σε κάθε πληροφορία που καταχωρείται, χωρίς να απαιτείται κάποιος προκαθορισμός. Η δομή των δεικτών που δημιουργούνται με το XMS κάνει πολύ μικρό το χρόνο πρόσβασης σε όλους τους κόμβους του συστήματος. Επίσης δεν υπάρχουν χρονικές ποινές πρόσβασης ανάλογα με τη δομή των δεδομένων που είναι αποθηκευμένα στο σύστημα.

- Ανεξαρτησία Σχήματος (Schema Independence)

Το Neocore XMS σχεδιάστηκε έτσι ώστε να αγνοεί σχήματα ή DTD. Τα περισσότερα συστήματα διαχείρισης XML απαιτούν όλη η πληροφορία να έχει περιγραφεί από ένα σχήμα ή DTD για λόγους αντιστοίχισης δεδομένων (data mapping). Το πρόβλημα που η εξάρτηση από το σχήμα θέτει, είναι ότι αφαιρεί τη δυνατότητα ύπαρξης ετερογενών δεδομένων μέσα σε παρόμοιους τύπους αρχείων. Επίσης η εξάρτηση αυτή κάνει αδύνατη τη χρήση του κυριοτέρου πλεονεκτήματος της XML, που είναι η επεκτασιμότητα. Η ανεξαρτησία σχήματος συνεπάγεται ότι δεν χρειάζεται να σχεδιαστεί η βάση δεδομένων και ότι δεν υπάρχει καμία ποινή για αλλαγή.

- Δυνατότητα κλιμάκωσης (Scalability)

Το Neocore XMS είναι σχεδιασμένο για να διαχειρίζεται τεράστιες αποθήκες αρχείων XML όλων των τύπων. Αυτό επιτεύχθηκε μέσω του χειρισμού των αρχείων XML, ως συλλογών

πληροφοριών (aggregations of information). Το Neocore XMS γνωρίζει, αλλά αγνοεί λειτουργικά, την κειμενο-κεντρική (document-centric) δομή της XML πληροφορίας. Τα συστήματα διαχείρισης δεδομένων XML είναι γνωστά για την κακή κλιμάκωση όταν το μέγεθος του εγγράφου ή το πλήθος των εγγράφων αυξάνεται. Το Neocore XMS σχεδιάστηκε ώστε να ανταποκρίνεται σε κλιμάκωση μεγάλων τμημάτων πληροφορίας, επιδεικνύοντας σταθερή (flat) επίδοση όσο το μέγεθος του συστήματος αυξάνεται. Επίσης η αύξηση μεγέθους ενός εγγράφου δεν έχει καμία επίδραση στην επίδοση του συστήματος.

- Αποτελεσματική χρήση αποθήκευσης (Effective use of storage)

Τα δίστιχα πληροφορίας αποτελούν ένα θεμελιώδες μέσο αποθήκευσης και διαχείρισης πληροφορίας. Ο διαχωρισμός της πληροφορίας σε δίστιχα μαζί με την αποτελεσματική δεικτοδότηση που χρησιμοποιεί η Neocore DPP (Digital Pattern Processing) τεχνολογία, δημιουργεί ένα πολύ αποτελεσματικότερο σχήμα αποθήκευσης (storage format) σε σχέση με οποιαδήποτε άλλη μεθοδολογία διαχείρισης πληροφορίας.

Επιτυγχάνοντας τους παραπάνω στόχους το Neocore XMS αποτελεί ένα ιδανικό εργαλείο για τη διαχείριση πολύπλοκης βιολογικής πληροφορίας. Μία από τις σημαντικότερες χρήσεις του συστήματος αυτού είναι σε προγράμματα ευθυγράμμισης βιολογικών ακολουθιών, όπως το BLAST.

2.8 *Σύνοψη και Συμπεράσματα*

Συνοψίζοντας την αναφορά μας στα πρότυπα XML που χρησιμοποιούνται στις Βιοεπιστήμες, πρέπει να τονίσουμε πως αυτά δεν είναι το ίδιο έυχρηστα, φιλικά, ή συμβατά με άλλα πρότυπα και γενικά έχουν διαφορετικά χαρακτηριστικά. Αυτά μπορεί να οφείλονται, για παράδειγμα, στο σχεδιασμό του προτύπου, στο είδος της πληροφορίας που αναπαριστά, ή στο πόσο πρόσφατα αναπτύχθηκε το πρότυπο και αναπαρίστανται στους Πίνακες 2.3 και 2.4.

Χαρακτηριστικά	CellML	SBML	BSML	BIOML
Αναπαράσταση	Βιολογικών Μοντέλων	Βιολογικών Μοντέλων	Γονιδιακών - Πρωτεϊνικών Πληροφοριών	Πειραματικών Γονιδιακών - Πρωτεϊνικών Πληροφοριών
Φιλικότητα	√√	√	√√√	√√√
Σαφήνεια	√√√	√√	√√√	√√√
Ποσότητα Πληροφορίας	√√√	√	√√√	√√√
Φορητότητα	√√	√√√	√√√	√√√
Διαθέσιμα Εργαλεία	√√	√√√	√√√	√√√
Δημοφιλία	√√√	√√	√√√	√√√

Πίνακας 2.3: Χαρακτηριστικά Προτύπων XML Μέρος 1

Χαρακτηριστικά	PSDML	GAME	MAGE-ML
Αναπαράσταση	Πρωτεϊνικών Πληροφοριών	Γονιδιακών - Πρωτεϊνικών Πληροφοριών	Πληροφοριών Πειραμάτων Μικροπινάκων
Φιλικότητα	√√√	√	√
Σαφήνεια	√√√	√√	√√
Ποσότητα Πληροφορίας	√√√	√√	√√
Φορητότητα	√√√	√√	√√√
Διαθέσιμα Εργαλεία	√√√	√	√√√
Δημοφιλία	√√√	√	√√√

Πίνακας 2.4: Χαρακτηριστικά Προτύπων XML Μέρος 2

3

Σύγκριση Ακολουθιών

3.1 *Περιγραφή Ακολουθιών*

Αναφερόμενοι σε ακολουθίες, εννοούμε ακολουθίες νουκλεοτιδίων (που περιέχονται στο DNA ή RNA), ή ακολουθίες αμινοξέων (τα οποία συνθέτουν τις πρωτεΐνες).

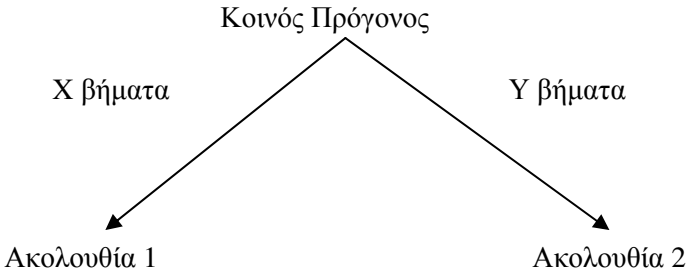
Ένα χαρακτηριστικό παράδειγμα μιας ακολουθίας αμινοξέων που συνθέτουν μια πρωτεΐνη με αριθμό πρόσβασης (Accession Number) IPI00000001 στην πρωτεΐνική βάση δεδομένων του Ευρωπαϊκού Εργαστηρίου Μοριακής Βιολογίας (EMBL) απεικονίζεται παρακάτω. Η πρωτεϊνική αυτή ακολουθία έχει όνομα SPLICE ISOFORM LONG OF DOUBLE-STRANDED RNA-BINDING PROTEIN STAUFEN HOMOLOG 1, μήκος 577 συμβόλων-αμινοξέων, ευρίσκεται στον ανθρώπινο οργανισμό και σε πλήρη ανάπτυξη έχει ως εξής:

```
MSQVQVQVQN PSAALSGSQI LNKNQSLLSQ PLMSIPSTTS SLPSENAGRP IQNSALPSAS        60
ITSTSAAAES ITPTVELNAL CMKLGKKPMY KPVDPYSRMQ STYNYNMRGG AYPPRYFYPF       120
PVPPLLYQVE LSVGGQQFNG KGKTRQAAKH DAAAKALRIL QNEPLPERLE VNGRESEEEN       180
LNKSEISQVF EIALKRNLPV NFEVARESGP PHMKNFVTKV SVGEFVGEGE GKSKKISKKN       240
AAIAVLEELK KLPPLPAVER VKPRIKKKTK PIVKPQTSPE YGQGINPISR LAQIQQAKKE       300
KEPEYTLLTE RGLPRRREFV MQVKVGNHTA EGTGTNKKVA KRNAAENMLE ILGFKVPQAQ       360
PTKPALKSEE KTPIKKPGDG RKVTFFEPGS GDENGTSNKE DEFRMPYLSH QQLPAGILPM       420
VPEVAQAVGV SQGHHTKDFT RAAPNPAKAT VTAMIARELL YGGTSPTAET ILKNNISSGH       480
VPHGPLTRPS EQLDYLSRVQ GFQVEYKDFP KNNKNEFVSL INCSSQPPLI SHGIGKDVES       540
CHDMAALNIL KLLSELDQQS TEMPRTGNGP MSVCGRC                                577
```

Στην παράσταση αυτή μετά από κάθε 60 αμινοξέα σημειώνεται το ως εκεί τοπικό μήκος της ακολουθίας. Τελικώς λοιπόν ο αριθμός 577 δείχνει το ολικό μήκος της πρωτεΐνης. Για παράδειγμα τα τελευταία επτά σύμβολα (έξι διαφορετικά αμινοξέα) MSVCGRC είναι τα

αμινοξέα μεθειονίνη (M), σερίνη (S), βαλίνη (V), κυστεΐνη (C), γλυκίνη (G), και αργινίνη (R). Στη συνέχεια ορίζουμε την έννοια των ομόλογων και ορθόλογων ακολουθιών:

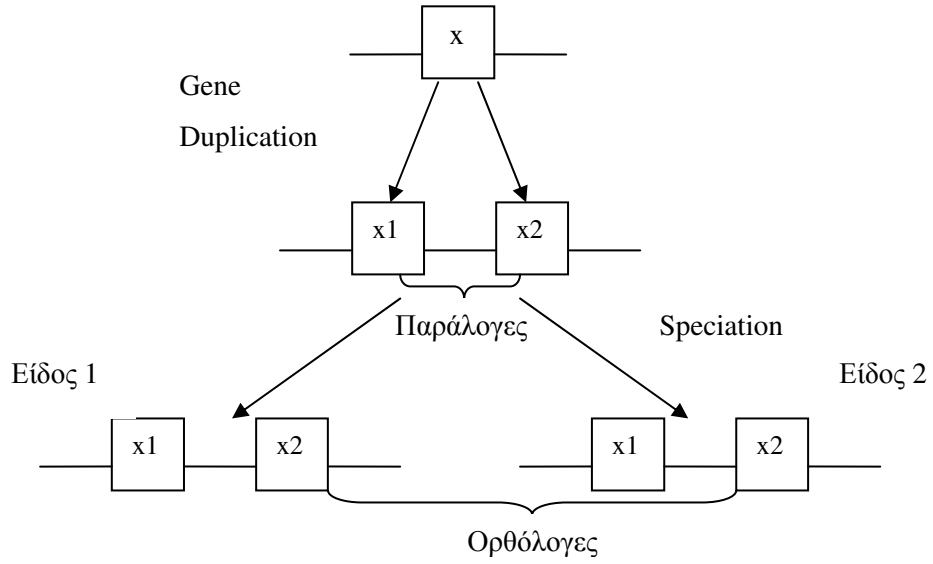
Ορισμός 1. *Ομόλογες (homologous) ονομάζονται δύο ακολουθίες αν έχουν τον ίδιον πρόγονο, κάτι που δεν είναι πάντα εύκολο να διαπιστωθεί. Πιο συγκεκριμένα οι ομόλογες διακρίνονται σε ορθόλογες (orthologs) ή παράλογες (paralogs).* (Σχήμα 3.1)



Σχήμα 3.1: Ομόλογες ακολουθίες

Ορισμός 2. *Ορθόλογες είναι οι ομόλογες ακολουθίες που ανήκουν σε διαφορετικά είδη, αλλά προήλθαν από κοινό γονιδιακό πρόγονο κατά τη διάρκεια της δημιουργίας των ειδών (speciation), ενώ παράλογες ονομάζονται οι ομόλογες ακολουθίες που ανήκουν στο ίδιο είδος και προήλθαν από γονιδιακή αντιγραφή (gene duplication).*

Για παράδειγμα αν ένα γονίδιο υπάρχει σε ένα είδος και στη συνέχεια το είδος αυτό μέσω εξελικτικών διαδικασιών διαχωριστεί σε δύο είδη, τότε τα διαχωρισμένα αντίγραφα του συγκεκριμένου αυτού γονιδίου στα διαφορετικά είδη είναι ορθόλογες μεταξύ τους ακολουθίες. Από την άλλη αν ένα γονίδιο σε έναν οργανισμό διπλασιάζεται για να καταλάβει δύο διαφορετικές θέσεις σε ένα γονιδίωμα, τότε τα δύο αυτά αντίγραφα είναι παράλογες ακολουθίες μεταξύ τους (Σχήμα 3.2).



Σχήμα 3.2: Παράλογες και Ορθόλογες ακολουθίες

Όλες οι γνωστές στους ερευνητές βιολογικές ακολουθίες είναι αποθηκευμένες σε διάφορες βάσεις δεδομένων οι σημαντικότερες των οποίων είναι οι εξής:

- Η GenBank είναι μία βάση δεδομένων ακολουθιών νουκλεοτιδίων και είναι ελεύθερα διαθέσιμη στην επιστημονική κοινότητα. Βρίσκεται υπό την αιγίδα του Εθνικού Ινστιτούτου Υγείας των Η.Π.Α και η κύρια πηγή της πληροφορίας που περιέχει προέρχεται από απ'ευθείας υποβολές δεδομένων που προκύπτουν ως αποτελέσματα εργαστηριακών πειραμάτων διαφόρων ερευνητικών ομάδων. Τα νέα δεδομένα που υποβάλλονται στην τράπεζα υφίστανται επεξεργασία και προστίθενται σχόλια για τη διευκόλυνση των ερευνητών. Ανά τακτά χρονικά διαστήματα τα ήδη κατατεθειμένα δεδομένα επανεξετάζονται και είναι πιθανόν να διορθώνονται, εάν εν τω μεταξύ έχουν προκύψει κάποια νέα δεδομένα. Η διαδικασία κατάθεσης των δεδομένων μπορεί να πραγματοποιηθεί εύκολα μέσω του διαδικτύου και στη συνέχεια οι υπεύθυνοι της βάσης αναλαμβάνουν την επεξεργασία και τη δημοσιοποίηση των δεδομένων αυτών. [Gen06]
- Η EMBL-Bank αποτελεί τη μεγαλύτερη βάση νουκλεοτιδικών ακολουθιών στην Ευρώπη και υπεύθυνο για αυτήν είναι το Ευρωπαϊκό Εργαστήριο Μοριακής Βιολογίας (EMBL). Τα δεδομένα προέρχονται από ανεξάρτητα ερευνητικά εργαστήρια καθώς και από ομάδες που ασχολούνται με τον προσδιορισμό των γονιδιωμάτων διαφόρων οργανισμών. Η κατάθεση δεδομένων στην EMBL-Bank είναι επίσης απλή διαδικασία και γίνεται μέσω του διαδικτύου κατά αντίστοιχο τρόπο με αυτόν της κατάθεσης στην GenBank ενώ και στη συνέχεια οι ακολουθίες χειρίζονται ομοίως με πριν (επεξεργασία από τους υπευθύνους και προσθήκη σχολίων). [Emb06]
- Η SWISS-PROT είναι μία βάση δεδομένων πρωτεϊνικών ακολουθιών που ιδρύθηκε το 1986 και συντηρείται από το Ελβετικό Ινστιτούτο Βιοπληροφορικής (Swiss Institute of Bioinformatics) σε συνεργασία με το Ευρωπαϊκό Ινστιτούτο Βιοπληροφορικής (European Bioinformatics Institute). Εκτός από τις ακολουθίες υπάρχουν και σχόλια, βιβλιογραφικές αναφορές, σύνδεσμοι με άλλες βάσεις δεδομένων σχετικές με κάθε εγγραφή, καθώς και άλλες πληροφορίες, όπως η βιολογική λειτουργία της κάθε ακολουθίας (όπου αυτή είναι γνωστή). [Spr06]
- Η PIR εδράζεται στο πανεπιστήμιο του Georgetown και αποτελεί τμήμα του Εθνικού Ιδρύματος Βιοϊατρικής Έρευνας (NBRF) των Η.Π.Α. Περιλαμβάνει μια σειρά από βάσεις δεδομένων που σχετίζονται με τη μελέτη πρωτεϊνών, κυριότερη των οποίων είναι η PIR-PSD (International Protein Sequence Database-Διεθνής Βάση Δεδομένων πρωτεϊνικών ακλουθιών). Η PSD είναι επίσης μία βάση δεδομένων πρωτεϊνικών ακολουθιών, που συνοδεύεται από σχόλια και πρόσθετες πληροφορίες. [Pir06]

3.2 *Σύγκριση Ακολουθιών (Comparing Sequences)*

Μία βασική τεχνική που χρησιμοποιείται στις Βιοεπιστήμες είναι η τεχνική της σύγκρισης ακολουθιών. Το κύριο πρόβλημα που τίθεται και στο οποίο προσπαθούμε να δώσουμε λύση με την τεχνική αυτή, είναι η αναζήτηση δομικών ή λειτουργικών ομοιοτήτων ανάμεσα σε γονιδιώματα και πρωτεΐνες που έχουμε αποθηκεύσει σε βάσεις δεδομένων (*Ως γονιδίωμα ορίζεται το σύνολο του γενετικού υλικού ενός κυττάρου*). Από την οπτική γωνία της βιολογίας, το κίνητρο της σύγκρισης ακολουθιών είναι ότι εφόσον όλοι οι ζωντανοί οργανισμοί συσχετίζονται εξελικτικά μεταξύ τους, θα πρέπει τα γονίδια και οι πρωτεΐνες των ειδών που είναι εξελικτικά κοντά, να παρουσιάζουν μεγάλες ομοιότητες.

Λόγω λοιπόν του τεραστίου όγκου των διαθεσίμων ακολουθιών υπάρχει άμεση ανάγκη να αναπτυχθούν αλγόριθμοι, ικανοί να συγκρίνουν πολυάριθμες ακολουθίες μεγάλου μήκους. Αυτοί οι αλγόριθμοι πρέπει να επιτρέπουν τη διαγραφή, εισαγωγή ή αντικατάσταση συμβόλων (π.χ. αμινοξέων), προσομοιώνοντας ουσιαστικά τέτοιες μεταλλαγές που γίνονται στη φύση. Όσον αφορά τη σύγκριση νουκλεοτιδίων τα προς σύγκριση σύμβολα είναι συνήθως ταυτόσημα (λόγω του μικρού πλήθους των βάσεων των νουκλεοτιδίων), ενώ όσον αφορά τα αμινοξέα, τα προς σύγκριση σύμβολα είτε είναι ταυτόσημα, είτε το ένα μπορεί να προέρχεται από το άλλο μετά από αντικαταστάσεις που συμβαίνουν στη φύση. Το λεξιλόγιο των ακολουθιών νουκλεοτιδίων αποτελείται από τέσσερα σύμβολα, ενώ το αντίστοιχο των ακολουθιών αμινοξέων περικλείει είκοσι, κάτι που υπαγορεύει πως οι τελευταίες εσωκλείουν περισσότερες πληροφορίες από τις πρώτες. Οι ακολουθίες νουκλεοτιδίων μπορούν επίσης να κωδικοποιούν τη δημιουργία έξι ακολουθιών αμινοξέων (οι οποίες μπορεί να εμπεριέχουν πρωτεΐνες) και στη συνέχεια να χρησιμοποιηθούν για σύγκριση κατά ζεύγη.

Παρακάτω ορίζεται η τεχνική της ευθυγράμμισης ακολουθιών:

Ορισμός 3. *Ευθυγράμμιση (alignment) ονομάζεται η διαδικασία της κατακόρυφης στοίχισης ακολουθιών, με στόχο την επίτευξη μεγίστου επιπέδου ταυτοσημίας. Το επίπεδο αυτό εκφράζει και το βαθμό ομοιότητας των ακολουθιών και κατ'επέκταση την πιθανότητα δύο ακολουθίες να είναι ομόλογες.*

Η ευθυγράμμιση ακολουθιών μπορεί να είναι είτε τοπική (local), είτε καθολική (global):

Ορισμός 4. *Η τοπική ευθυγράμμιση (local alignment) βρίσκει μικρού μήκους περιοχές ομοιότητας μεταξύ δύο ακολουθιών. Οι τελευταίες δεν χρειάζεται να έχουν υψηλά επίπεδα ομοιότητας σε όλο το μήκος τους, προκειμένου η ευθυγράμμιση να είναι αποτελεσματική. Το χαρακτηριστικό (και συνάμα πλεονέκτημα) αυτό της τοπικής ευθυγράμμισης, την κάνει εξαιρετικά χρήσιμη σε περιπτώσεις που επιθυμούμε την αναζήτηση περιοχών-τομέων ιδιαίτερου ενδιαφέροντος σε πρωτεΐνες, ή περιοχών εσωνίων στα γονίδια.*

Ορισμός 5. *Η τεχνική της καθολικής ευθυγράμμισης (global alignment) βρίσκει τη βέλτιστης ομοιότητας ευθυγράμμιση δύο ακολουθιών καθ'όλο το μήκος τους. Εξ'αιτίας αυτού του γεγονότος η τεχνική αυτή δίνει καλά αποτελέσματα κυρίως σε περιπτώσεις ομόλογων και σχετικά συγκρίσιμου μήκους νουκλεοτιδικών ακολουθιών και είναι συνεπώς λιγότερο ευαίσθητη σε περιπτώσεις μεγάλης απόκλισης δύο ακολουθιών.*

Συνήθως χρησιμοποιούμε κενά (gaps), αναπαριστώντας τα με το σύμβολο '-', για να δειχθεί ότι είναι προτιμότερο να μην ευθυγραμμισθούν δύο σύμβολα, έτσι ώστε πολλά άλλα ζεύγη να μπορούν να ευθυγραμμισθούν. Σε τοπικές ευθυγραμμίσεις υπάρχουν μεγάλες περιοχές κενών, ενώ στις καθολικές τα κενά είναι περισσότερο διασκορπισμένα σε όλο το μήκος της ακολουθίας. Ένα μέτρο ομοιότητας είναι η επί τοις εκατό ταυτοσημία (percent identity), που ορίζεται ως ο επί τοις εκατό λόγος του αριθμού των στηλών με όμοια σύμβολα, προς τον αριθμό συμβόλων της μακρύτερης ακολουθίας. Για παράδειγμα στο Σχήμα 3.3 ευθυγραμμίζονται δύο ακολουθίες, οι AGTCAACTACCC και GCTGAACGTCGC και οι περιοχές ομοίων συμβόλων έχουν υπογραμμιστεί.

A G T C A A C T A C C C
G C T G A A C G T C G C

Σχήμα 3.3: Ευθυγράμμιση ακολουθιών

Σε αυτό το παράδειγμα συγκρίνονται δύο ακολουθίες μήκους δώδεκα συμβόλων, ενώ τα κοινά μεταξύ τους σύμοβολα είναι έξι. Συνεπώς η επί τοις εκατό ταυτοσημία τους είναι 6/12 = 50%.

Ένα πιθανό μέτρο ή αποτέλεσμα-σκορ (score) μίας ευθυγράμμισης υπολογίζεται αθροίζοντας τα ταιριάσματα των ίδιων (ή παρομοίων) συμβόλων και μετρώντας τα κενά ως αρνητικά. Όσο μεγαλύτερο είναι το σκορ, τόσο μεγαλύτερος είναι και ο βαθμός ομοιότητας των ευθυγραμμισμένων ακολουθιών.

3.3 *Αλγόριθμοι Σύγκρισης Ακολουθιών*

Οι αλγόριθμοι που χρησιμοποιούνται για τη σύγκριση ακολουθιών λειτουργούν είτε ευθυγραμμίζοντας ακολουθίες κατά ζεύγη, είτε ευθυγραμμίζοντας πολλές ακολουθίες μαζί.

Στην ευθυγράμμιση κατά ζεύγη (pairwise alignment) τα κενά προστίθενται εκεί που πρέπει, ώστε να μεγιστοποιηθεί το πλήθος των ταιριασμάτων. Π.χ αν θέλουμε να συγκρίνουμε τις ακολουθίες ABCD και BD τότε θα γίνει η ευθυγράμμιση

A B C D
- B - D

ενώ για τη σύγκριση της ABCD με την BUC θα έχουμε την ευθυγράμμιση

A B - C D
- B U C -

Η λειτουργία αυτών των αλγορίθμων ομοιάζει με αυτήν αλγορίθμων αυτόματης ορθογραφικής διόρθωσης όπου γίνεται αναζήτηση σε λεξικό για τα καλύτερα ταιριάσματα (με υποδείξεις για πιθανές διορθώσεις). Βάσει όλων αυτών συμπεραίνουμε ότι η σύγκριση ακολουθιών είναι ένα πρόβλημα βελτιστοποίησης. Ένας διαισθητικός τρόπος για ευθυγράμμιση δύο ακολουθιών είναι η κατασκευή ενός πίνακα αληθείας (Boolean) που αναπαριστά τις πιθανές ευθυγραμμίσεις που μπορούν να παρατηρηθούν εξ'όψεως. Αν έχοντας αρχικώς μηδενικές τιμές στα στοιχεία του πίνακα, αντιστοιχήσουμε τα σύμβολα της πρώτης ακολουθίας στις γραμμές του πίνακα, ενώ τα της δεύτερης στις στήλες, τότε οι τιμές του πίνακα θα γίνουν μονάδες (1) όταν κάποιο σύμβολο της γραμμής αντιστοιχεί σε ίδιο σύμβολο στη στήλη. Ένα παράδειγμα είναι ο Πίνακας 3.1.

	A	B	C	D	F	G	E
C	0	0	1	0	0	0	0
D	0	0	0	1	0	0	0
E	0	0	0	0	0	0	1
D	0	0	0	1	0	0	0
F	0	0	0	0	1	0	0
A	1	0	0	0	0	0	0
E	0	0	0	0	0	0	1

Πίνακας 3.1: Πίνακας αληθείας ακολουθιών

Η χρήση των πινάκων αυτών απαιτεί οπτικό έλεγχο για ανίχνευση μεγάλων τμημάτων διαγωνίων (με μονάδες) που υποδεικνύουν πιθανές κοινές περιοχές ομοίων συμβόλων. Στην περίπτωση που δύο ακολουθίες είναι μεγάλες και το λεξιλόγιό τους είναι μικρό (περίπτωση των τεσσάρων βάσεων νουκλεοτιδίων στο DNA), διασκορπίζονται οι άσοι στον πίνακα και πρέπει να ελεγχθούν πολλές διαγώνιοι ώσπου να βρεθεί η βέλτιστη που μεγιστοποιεί τον αριθμό των ταιριασμάτων (περίπτωση θορύβου, Πίνακας 3.2). Για να μειωθεί ο θόρυβος συνήθως συγκρίνονται πολλαπλά σύμβολα ταυτόχρονα, αντί για μόνο ένα.

	A	T	C	G	T	T	G	C	A
C	0	0	1	0	0	0	0	1	0
G	0	0	0	1	0	0	1	0	0
T	0	1	0	0	1	1	0	0	0
A	1	0	0	0	0	0	0	0	1
T	0	1	0	0	1	1	0	0	0
C	0	0	1	0	0	0	0	1	0
C	0	0	1	0	0	0	0	1	0
A	1	0	0	0	0	0	0	0	1
T	0	1	0	0	1	1	0	0	0

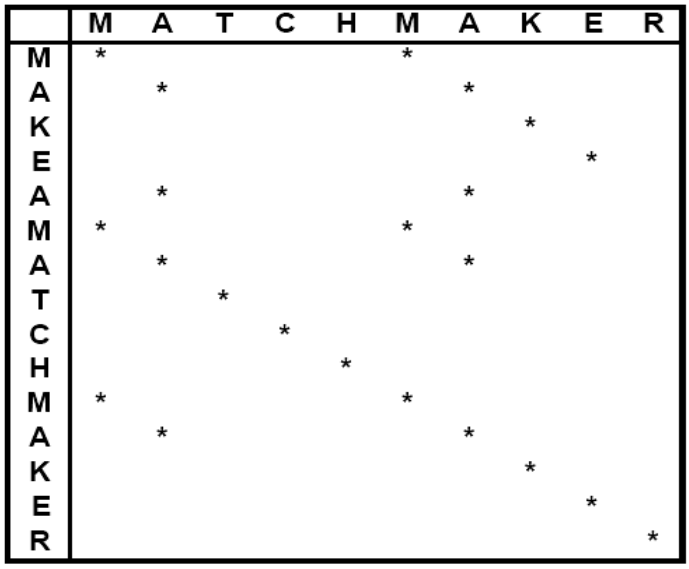
Πίνακας 3.2: Παράδειγμα περίπτωσης θορύβου

Μία ενδιαφέρουσα περίπτωση που συχνά συναντούμε στη βιολογία είναι όταν μία ακολουθία περιλαμβάνει επαναλαμβανόμενα αντίγραφα υπακολουθιών της. Τότε και πάλι έχουμε πολλαπλές διαγωνίους και χρειάζεται οπτική θεώρηση για την ανίχνευση της βέλτιστης (Πίνακας 3.3).

	A	C	G	A	C	G	A	C	G
A	1	0	0	1	0	0	1	0	0
C	0	1	0	0	1	0	0	1	0
T	0	0	0	0	0	0	0	0	0
G	0	0	1	0	0	1	0	0	1
C	0	1	0	0	1	0	0	1	0
T	0	0	0	0	0	0	0	0	0
C	0	1	0	0	1	0	0	1	0
A	1	0	0	1	0	0	1	0	0
C	0	1	0	0	1	0	0	1	0

Πίνακας 3.3: Παράδειγμα ακολουθίας με επαναλαμβανόμενα αντίγραφα υπακολουθιών της

Εναλλακτικά των πινάκων αληθείας, μπορούν να χρησιμοποιηθούν οι πίνακες με τελείες (dot matrices). Οι πίνακες με τελείες μπορούν να προκύψουν από τους πίνακες αληθείας, αν τα μηδενικά σβηστούν και οι μονάδες αντικατασταθούν με τελείες (Σχήμα 3.4).

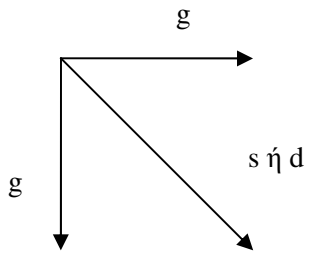


Σχήμα 3.4: Πίνακας με τελείες

Οι δύο αυτές μέθοδοι είναι απλά τρόποι για να οδηγηθούμε, οπτικά, σε ποιοτική εκτίμηση μίας ευθυγράμμισης, χωρίς να παρέχονται πληροφορίες για τη στατιστική σημασία της ευθυγράμμισης αυτής. Η πολυπλοκότητα χώρου και χρόνου των μεθόδων αυτών μπορεί να γίνει ανυπόφορη για μεγάλα μήκη ακολουθιών. Παρ'όλα αυτά για ακολουθίες με μεγάλη ομοιότητα δε χρειάζεται να κατασκευαστεί ολόκληρος ο πίνακας παρά μόνο τα στοιχεία γύρω από τη διαγώνιο και τότε η πολυπλοκότητα είναι γραμμική.

Όταν λοιπόν ευθυγραμμίζουμε δύο ακολουθίες μήκους n, τότε οι πιθανές καθολικές ευθυγραμμίσεις είναι $\binom{2n}{n} = \frac{(2n!)}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{2\pi n}}$,δηλαδή πάρα πολλές. Για να βρούμε λοιπόν τη βέλτιστη ευθυγράμμιση αποτελεσματικά, χρησιμοποιούμε το δυναμικό προγραμματισμό (dynamic programming).

Ο δυναμικός προγραμματισμός ουσιαστικά επιλύει ένα στιγμιότυπο προβλήματος εκμεταλλευόμενος λύσεις που έχουν ήδη υπολογιστεί για μικρότερα υποπροβλήματα του προβλήματος αυτού. Στην περίπτωσή μας ο δυναμικός προγραμματισμός εντοπίζει τη βέλτιστη ευθυγράμμιση δύο ακολουθιών, βάσει των βελτίστων ευθυγραμμίσεων των προθεμάτων των ακολουθιών. Με τη μέθοδο αυτή λοιπόν κατασκευάζεται ένας κατευθυνόμενος ακυκλικός γράφος βάσει του οποίου αναζητείται η βέλτιστη ευθυγράμμιση (το βέλτιστο μονοπάτι) με κριτήριο το κόστος. Οι πίνακες αληθείας, που δείξαμε παραπάνω, διασχίζονται ξεκινώντας από το στοιχείο (1,1) με την κατεύθυνση που φαίνεται στο παρακάτω σχήμα, αθροίζοντας τα κόστη για κάθε κίνηση.



Σχήμα 3.5: Κατεθύνσεις διάσχισης

Ορίζοντας ως g το κόστος εισαγωγής ενός κενού, s το κόστος του ταιριάσματος δύο συμβόλων και d το κόστος ανοχής ευθυγράμμισης δύο ανόμοιων συμβόλων αναζητούμε τη βέλτιστη ευθυγράμμιση, δηλαδή το αλγεβρικά μέγιστο συνολικό κόστος-αποτέλεσμα. Με ενδεικτικές τιμές g = -2, s = 1, d = -1, συνεπάγεται πως για ταυτόσημες ακολουθίες μήκους n θα είχαμε μέγιστο κόστος ίσο με n (διάσχιση της κύριας διαγωνίου), ενώ για σύγκριση ακολουθίας μήκους n με την κενή ακολουθία θα είχαμε το ελάχιστο κόστος -2n (εισαγωγή n κενών). Για παράδειγμα για την σύγκριση των ακολουθιών ATGTCGCCTGTACGTACCT και AGGGCAGCATACT μπορούμε να έχουμε την εξής ευθυγράμμιση:

A	T	G	T	C	G	C	C	T	G	T	A	C	G	T	A	C	C	T
A	G	G	G	C	A	G	C	A	-	T	A	C	-	T	-	-	-	-

όπου υπάρχουν οχτώ όμοια σύμβολα, πέντε ανόμοια και έξι στοιχίσεις με κενά, όπως φαίνεται και στο Σχήμα 3.6, όπου τα γράμματα Ο, Α και Κ αντιπροσωπεύουν τις λέξεις όμοιος, ανόμοιος και κενό.

A	T	G	T	C	G	C	C	T	G	T	A	C	G	T	A	C	C	T
A	G	G	G	C	A	G	C	A	-	T	A	C	-	T	-	-	-	-
O	A	O	A	O	A	A	O	A	K	O	O	O	K	O	K	K	K	K

Σχήμα 3.6: Παράδειγμα ευθυγράμμισης

και συνεπώς το αποτέλεσμα είναι ίσο με $1\times 8+(-1)\times 5+(-2)\times 6=-9$. Πρέπει να τονιστεί βέβαια, πως οι ποινές της χρήσης των κενών δεν είναι πάντα σταθερές, όπως τις επιλέξαμε στο προηγούμενο παράδειγμα. Συνήθως η εισαγωγή κενού σε μία περιοχή όπου δεν υπήρχε κενό (άνοιγμα κενού), επιφέρει μεγάλη ποινή, ενώ η επέκταση ενός κενού επιφέρει πολύ μικρότερη (περίπου 10 φορές μικρότερη ποινή). Η ποινή χρήσης κενού G μπορεί να δίνεται από τον τύπο: $G = g + x \times n$, όπου g η ποινή ανοίγματος κενού, x η ποινή επέκτασης κενού και n το μήκος των επεκτάσεων. Για παράδειγμα, με $g=-12$ και $x=-1$ στην περίπτωση του Σχήματος 4.6 (διατηρώντας ίδια τα κόστη για όμοια και ανόμοια σύμβολα), θα είχαμε αποτέλεσμα ίσο με $1\times 8+(-1)\times 5+3\times(-12)+3\times(-1)=-36$.

Ο δυναμικός προγραμματισμός εφαρμόζεται για τη σύγκριση δύο οποιωνδήποτε ακολουθιών, αρκεί η επιλογή των βαρών–κοστών g, s, d να είναι η κατάλληλη. Στην ευθυγράμμιση ακολουθιών νουκλεοτιδίων η μέθοδος αυτή έχει πολύ καλά αποτελέσματα, ενώ για την ευθυγράμμιση ακολουθιών αμινοξέων είμαστε περισσότερο προσεκτικοί γιατί πρέπει να ληφθεί υπόψιν η εξέλιξη των ειδών. Για τη σύγκριση λοιπόν των ακολουθιών αμινοξέων, οι βιολόγοι έχουν αναπτύξει τριγωνικούς πίνακες διαστάσεων 20×20 (επειδή υπάρχουν 20 διαφορετικά αμινοξέα στις πρωτεΐνες). Οι πίνακες αυτοί, οι γνωστότεροι των οποίων είναι οι PAM (Percent Accepted Mutation-Επί τοις εκατό Αποδεκτή Μετάλλαξη) και BLOSUM (Blocks Substitution Matrix-Πίνακας Υποκατάστασης Στοιχείων), περιέχουν τα κόστη για σύγκριση ομοίων και διαφορετικών αμινοξέων. Οι τιμές στον πίνακα είναι ανάλογες της πιθανότητας μετάλλαξης ενός αμινοξέως σε άλλο. Συνήθως χρησιμοποιείται ένας αριθμός μετά το όνομα του πίνακα (π.χ PAM 2), που υποδεικνύει την επιείκεια στην εκτίμηση των διαφορών. Όσο μεγαλύτερος είναι ο αριθμός αυτός, τόσο μεγαλύτερη και η επιείκεια του πίνακα στις διαφορές, δηλαδή η αναμενόμενη εξελικτική απόσταση (για παράδειγμα εάν στην προηγούμενη περίπτωση το κόστος ανοχής ανόμοιων συμβόλων d ήταν 0 αντί για -1, τότε ο πίνακας θα ήταν πιο επιεικής).

Οι πίνακες PAM έχουν κατασκευαστεί με βάση καθολικές ευθυγραμμίσεις στενά συνδεδεμένων πρωτεϊνών, ενώ οι πίνακες BLOSUM κατασκευάστηκαν με βάση τοπικές ευθυγραμμίσεις, όχι απαραιτήτως συσχετισμένων ακολουθιών.

Για παράδειγμα ο πίνακας BLOSUM62 είναι ο Πίνακας 3.4,

BLOSUM 62

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	4	0	-2	-1	-2	0	-2	-1	-1	-1	-1	-2	-1	-1	-1	1	0	0	-3	-2
C		9	-3	-4	-2	-3	-3	-1	-3	-1	-1	-3	-3	-3	-3	-1	-1	-1	-2	-2
D			6	2	-3	-1	-1	-3	-1	-4	-3	1	-1	0	-2	0	-1	-3	-4	-3
E				5	-3	-2	0	-3	1	-3	-2	0	-1	2	0	0	-1	-2	-3	-2
F					6	-3	-1	0	-3	0	0	-3	-4	-3	-3	-2	-2	-1	1	3
G						6	-2	-4	-2	-4	-3	0	-2	-2	-2	0	-2	-3	-2	-3
H							8	-3	-1	-3	-2	1	-2	0	0	-1	-2	-3	-2	2
I								4	-3	2	1	-3	-3	-3	-3	-2	-1	3	-3	-1
K									5	-2	-1	0	-1	1	2	0	-1	-2	-3	-2
L										4	2	-3	-3	-2	-2	-2	-1	1	-2	-1
M											5	-2	-2	0	-1	-1	-1	1	-1	-1
N												6	-2	0	0	1	0	-3	-4	-2
P													7	-1	-2	-1	-1	-2	-4	-3
Q														5	1	0	-1	-2	-2	-1
R															5	-1	-1	-3	-3	-2
S																4	1	-2	-3	-2
T																	5	0	-2	-2
V																		4	-3	-1
W																			11	2
Y																				7

Πίνακας 3.4: BLOSUM 62

ενώ ο PAM250 είναι ο Πίνακας 3.5.

PAM250

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	2	-2	0	0	-2	0	0	1	-1	-1	-2	-1	-1	-3	1	1	1	-6	-3	0	0	0	0	-8
R	-2	6	0	-1	-4	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	2	-4	-2	-1	0	-1	-8
N	0	0	2	2	-4	1	1	0	2	-2	-3	1	-2	-3	0	1	0	-4	-2	-2	2	1	0	-8
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-6	-1	0	0	-7	-4	-2	3	3	-1	-8
C	-2	-4	-4	-5	12	-5	-5	-3	-3	-2	-6	-5	-5	-4	-3	0	-2	-8	0	-2	-4	-5	-3	-8
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2	1	3	-1	-8
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2	3	3	-1	-8
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	0	1	0	-7	-5	-1	0	0	-1	-8
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2	1	2	-1	-8
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4	-2	-2	-1	-8
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2	-3	-3	-1	-8
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2	1	0	-1	-8
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2	-2	-2	-1	-8
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1	-4	-5	-2	-8
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1	-1	0	-1	-8
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1	0	0	0	-8
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0	0	-1	0	-8
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6	-5	-6	-4	-8
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2	-3	-4	-2	-8
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4	-2	-2	-1	-8
B	0	-1	2	3	-4	1	3	0	1	-2	-3	1	-2	-4	-1	0	0	-5	-3	-2	3	2	-1	-8
Z	0	0	1	3	-5	3	3	0	2	-2	-3	0	-2	-5	0	0	-1	-6	-4	-2	2	3	-1	-8
X	0	-1	0	-1	-3	-1	-1	-1	-1	-1	-1	-1	-1	-2	-1	0	0	-4	-2	-1	-1	-1	-1	-8
*	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	1

Πίνακας 3.5: PAM250

Το μεγάλο πλεονέκτημα της χρήσης του δυναμικού προγραμματισμού είναι ότι εγγυάται της εύρεσης της βέλτιστης ευθυγράμμισης δύο ακολουθιών, ενώ τα κύρια μειονεκτήματά του είναι η βραδύτητα και η υπολογιστική πολυπλοκότητα. Οι δύο συχνότερα χρησιμοποιούμενοι

αλγόριθμοι δυναμικού προγραμματισμού είναι αυτοί που αναπτύχθηκαν από τα ζευγάρια συγγραφέων Needleman-Wunsch και Smith-Waterman και αναλύονται στη συνέχεια:

- Αλγόριθμος Needleman-Wunch (1970)

Ο αλγόριθμος αυτός βρίσκει τη βέλτιστη καθολική ευθυγράμμιση δύο ακολουθιών μήκους m και n, χρησιμοποιώντας ένα συγκεκριμένο σύστημα αποτελεσμάτων και έναν δοσμένο πίνακα υποκατάστασης στοιχείων (substitution matrix). Ο αλγόριθμος αυτός σχηματίζει έναν δισδιάστατο πίνακα με τις δύο ακολουθίες να αντιστοιχούν σε στήλη και γραμμή αντίστοιχα (όπως στους πίνακες αληθείας) και στη συνέχεια συμπληρώνει όλα τα στοιχεία του πίνακα, βασιζόμενος σε συγκεκριμένο σύστημα αποτελεσμάτων. Το στοιχείο M στη θέση (i,j) του πίνακα δίνεται από τη σχέση $M(i,j) = MAX(M_{i-1,j-1} + S(A_i B_j), M_{i-1,j} + g, M_{i,j-1} + g)$, όπου g το κόστος χρήσης κενού, και $S(A_i, B_j)$ το κόστος σύγκρισης δύο συμβόλων (όμοιων ή ανόμοιων, που μπορεί να λαμβάνεται από πίνακες υποκατάστασης στοιχείων). Εφόσον λοιπόν συμπληρωθούν όλα τα στοιχεία του πίνακα, ξεκινούμε από το τελευταίο στοιχείο του πίνακα (στοιχείο (m,n)) και σχηματίζουμε προς τα πίσω, το βέλτιστο (μεγίστου αποτελέσματος) μονοπάτι έως το πρώτο στοιχείο του πίνακα. Το μονοπάτι αποτελεί και τη βέλτιστη ευθυγράμμιση, το αποτέλεσμα-σκορ της οποίας, ισούται με την τιμή του στοιχείου (m,n) του πίνακα, από το οποίο και ξεκίνησε το βέλτιστο μονοπάτι.

Επίσης πρέπει να αναφερθεί πως προκειμένου να συμπληρωθούν τα στοιχεία της πρώτης στήλης και της πρώτης γραμμής ($i = 1$ και $j = 1$) πρέπει να αρχικοποιηθεί η "(-1) κατάσταση", κάτι που μπορεί να γίνει είτε με μηδενικά, είτε θεωρώντας την ευθυγράμμιση κάθε ακολουθίας με την κενή ακολουθία). Για παράδειγμα εάν θέλουμε να ευθυγραμμίσουμε τις ακολουθίες ACTGATTCA και ACGCATCA, με δοσμένο σύστημα αποτελεσμάτων (κόστος $g = -2$, κόστος ανοχής ανόμοιων συμβόλων $d = -3$ και κόστος ταιριάσματος όμοιων συμβόλων $s = 2$) και αρχικοποίηση με την κενή ακολουθία θα έχουμε τους πίνακες:

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

Συνεπώς η βέλτιστη ευθυγράμμιση είναι η

A	C	T	G	–	A	T	T	C	A
A	C	–	G	C	A	T	–	C	A

όπου παρατηρούμε πως η κίνηση του μονοπατιού οριζοντίως, συνεπάγεται χρήση κενού στην κατακόρυφη ακολουθία (ακολουθία που τα στοιχεία της αντιστοιχούν στις γραμμές του πίνακα) και το αντίστροφο. Προκειμένου να επαληθεύσουμε το αποτέλεσμα της ευθυγράμμισης, παρατηρούμε πως η τιμή του στοιχείου έναρξης του μονοπατιού είναι ίση με 8. Πράγματι, εφόσον στην ευθυγράμμιση έχουμε επτά όμοια στοιχεία και τρείς χρήσεις κενών το αποτέλεσμα είναι: $7\times 2+3\times(-2)=14-6=8$. Τέλος πρέπει να αναφέρουμε πως τόσο η χρονική, όσο και η χωρική πολυπλοκότητα του αλγόρίθμου Needleman-Wunch για ακολουθίες μήκους m και n, είναι $O(m\times n)$ ή γενικότερα $O(N^2)$.

- Αλγόριθμος Smith-Waterman (1981)

Ο αλγόριθμος Smith-Waterman εντοπίζει τη βέλτιστη τοπική ευθυγράμμιση δύο ακολουθιών χρησιμοποιώντας διαφορετικό σύστημα αποτελεσμάτων απ'ότι ο Needleman-Wunch και κάποιν πίνακα υποκατάστασης στοιχείων. Ομοίως με πριν κατασκευάζεται ο δισδιάστατος πίνακας των δύο ακολουθιών προκειμένου να συμπληρωθεί. Εδώ, η τιμή του στοιχείου M(i,j) δίνεται από τον τύπο: $M(i,j)=MAX(M_{i-1,j-1}+S(A_i,B_j),M_{i-1,j}+g,M_{i,j-1}+g,0)$, όπου με τη χρήση του 0, αποκλείονται οι αρνητικές τιμές. Με μηδενική αρχικοποίηση της "(-1) κατάστασης" συμπληρώνονται τα στοιχεία του πίνακα και στη συνέχεια εντοπίζεται μέσα στον πίνακα το στοιχείο με την υψηλότερη τιμή. Από το στοιχείο αυτό ξεκινά το βέλτιστο (μεγίστου αποτελέσματος) τοπικό μονοπάτι που τερματίζεται σε μηδενικό στοιχείο. Το σκορ της ευθυγράμμισης ισούται και πάλι με την τιμή του στοιχείου έναρξης του μονοπατιού. Για παράδειγμα, εάν θέλουμε να ευθυγραμμίσουμε τις ακολουθίες ATGCATCCCATGAC και TCTATATCCGT με τον αλγόριθμο Smith-Waterman με τιμές $g=-2$, $d=-3$ και $s=2$, θα έχουμε τον πίνακα

		A	T	G	C	A	T	C	C	C	A	T	G	A	C
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	2	0	0	0	2	0	0	0	0	2	0	0	0
C	0	0	0	0	2	0	0	4	2	2	0	0	0	0	2
T	0	0	2	0	0	0	0	2	1	0	0	2	0	0	0
A	0	2	0	0	0	2	0	0	0	0	2	0	0	2	0
T	0	0	4	2	0	0	2	0	0	0	0	4	2	0	0
A	0	2	0	0	0	2	0	0	0	0	2	0	0	2	0
T	0	0	4	2	0	0	4	2	0	0	0	4	0	0	0
C	0	0	2	0	4	0	0	6	4	2	0	0	0	0	2
C	0	0	0	0	2	0	0	4	8	6	4	2	0	0	2
G	0	0	0	2	0	0	0	2	6	5	3	1	4	2	0
T	0	0	2	0	0	0	2	0	4	3	2	5	3	1	0

όπου η ευθυγράμμιση είναι η

A	T	C	C
A	T	C	C

με αποτέλεσμα ίσο με $4 \times 2 = 8$, που ισούται με την τιμή του στοιχείου έναρξης του μονοπατιού. Η χρονική και χωρική πολυπλοκότητα και αυτού του αλγορίθμου για ακολουθίες μήκους m και n, είναι $O(m \times n)$ ή γενικότερα $O(N^2)$.Ο αλγόριθμος αυτός χρησιμοποιείται από το πρόγραμμα BLAST που είναι και το δημοφιλέστερο εργαλείο τοπικής ευθυγράμμισης ακολουθιών στο χώρο των Βιοεπιστημών.

Οι δύο αυτοί αλγόριθμοι είναι οι πιο αξιόπιστοι και εγγυώνται της εύρεσης βέλτιστων τοπικών και καθολικών ευθυγραμμίσεων, αλλά η εκτέλεση τους απαιτεί αξιοσημείωτο χρόνο και χώρο, εκτός αν χρησιμοποιούμε εξειδικευμένο υλικό ή ερευνούμε σε μικρή βάση δεδομένων.

3.4 *Συστήματα Σύγκρισης*

Προκειμένου λοιπόν να αποφύγουμε την χρονική και χωρική πολυπλοκότητα του δυναμικού προγραμματισμού καταφεύγουμε σε ευριστικούς (heuristic) αλγορίθμους, οι οποίοι αν και δεν εγγυώνται πως θα βρουν τη βέλτιστη λύση-ευθυγράμμιση, εγγυώνται ότι θα βρουν έυκολα και σε πολύ μικρό χρόνο, μία λύση κοντά στη βέλτιστη. Έτσι, τα πλέον δημοφιλή πραγματικά συστήματα που πραγματοποιούν συγκρίσεις βιολογικών ακολουθιών και τα οποία θα εξετάσουμε, είναι ευριστικές μέθοδοι. Προφανώς οι μέθοδοι αυτές είναι λιγότερο ευαίσθητες από τον δυναμικό προγραμματισμό, αλλά είναι ταυτόχρονα και σημαντικά ταχύτερες. Συγκεκριμένα θα ασχοληθούμε με τα συστήματα BLAST, FASTA και CLUSTAL, ξεκινώντας με τη μελέτη του BLAST.

3.4.1 *BLAST*

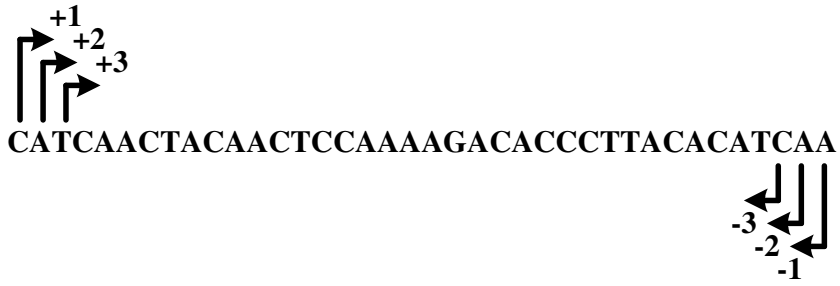
Ένα από τα πλέον εύχρηστα συστήματα που δίνει λύση στο πρόβλημα πολυπλοκότητας του δυναμικού προγραμματισμού είναι το BLAST (Basic Local Alignment Search Tool-Βασικό εργαλείο αναζήτησης τοπικής ευθυγράμμισης). Το BLAST είναι μία ευριστική (heuristic) μέθοδος που πρωτοδημοσιεύθηκε το 1990 από το NCBI και σήμερα είναι το δημοφιλέστερο εργαλείο τοπικών ευθυγραμμίσεων βιολογικών ακολουθιών, με χιλιάδες καθημερινές διαδικτυακές επισκέψεις και έρευνες-εκτελέσεις. Μπορεί να εκτελεστεί μέσω διαδικτύου είτε χωρίς αυτό, εάν βέβαια ο χρήστης κατέχει τα απαραίτητα προγράμματα, καθώς και τις απαιτούμενες βάσεις δεδομένων, οι οποίες συνήθως είναι μεγάλου μεγέθους.

Το BLAST μπορεί να χρησιμοποιηθεί για τη σύγκριση διαφόρων τύπων ακολουθιών. Ο Πίνακας 3.6 απεικονίζει τις διάφορες εκδοχές του (Με τον όρο "ακολουθία ερώτηση"-query sequence, ονομάζουμε την βιολογική ακολουθία που εισάγουμε ως είσοδο στο πρόγραμμα BLAST).

blastp	Συγκρίνει την ακολουθία-ερώτηση αμινοξέων έναντι βάσης δεδομένων ακολουθιών πρωτεϊνών.
blastn	Συγκρίνει την ακολουθία-ερώτηση νουκλεοτιδίων έναντι βάσης δεδομένων ακολουθιών νουκλεοτιδίων .
blastx	Συγκρίνει την ακολουθία-ερώτηση νουκλεοτιδίων μεταφρασμένη σε όλα τα πλαίσια ανάγνωσης, έναντι βάσης δεδομένων ακολουθιών πρωτεϊνών. Αυτή η εκδοχή μπορεί να χρησιμοποιηθεί για την εύρεση πιθανών προϊόντων μετάφρασης μιας άγνωστης ακολουθίας νουκλεοτιδίων.
tblastn	Συγκρίνει την ακολουθία-ερώτηση πρωτεΐνης έναντι βάσης δεδομένων ακολουθιών νουκλεοτιδίων, δυναμικά μεταφραζόμενων σε όλα τα πλαίσια αναγνώσεώς τους.
tblastx	Συγκρίνει τις έξι-πλαισίων μεταφράσεις της ακολουθίας-ερώτησης νουκλεοτιδίων έναντι των έξι-πλαισίων μεταφράσεων της βάσης δεδομένων ακολουθιών νουκλεοτιδίων.

Πίνακας 3.6: Εκδοχές του BLAST ανάλογα με τις χειριζόμενες ακολουθίες

Πρέπει να σημειωθεί πως όταν μιλούμε για έξι πλαισίων μεταφράσεις στο BLAST (αλλά και στις άλλες μεθόδους ευθυγράμμισης ακολουθιών), δεν εννοούμε τα πλαίσια ανάγνωσης που είδαμε στο Σχήμα 2.6. Σε αυτήν την περίπτωση, εννοούμε τις τρεις μεταφράσεις μιας νουκλεοτιδικής ακολουθίας ξεκινώντας τη μετάφραση από τα αριστερά προς τα δεξιά και τρεις ακόμα ξεκινώντας αντίστροφα. Ανάλογα με το νουκλεοτίδιο εκκίνησης της μετάφρασης, το πλαίσιο ανάγνωσης αριθμείται και ανάλογα με την κατεύθυνση μετάφρασης παίρνει πρόσημο (Σχήμα 3.7).



Σχήμα 3.7: Πλαίσια ανάγνωσης στο BLAST

Ο καλύτερος τρόπος για να εξηγήσουμε τη λειτουργία του BLAST είναι καταφεύγοντας και πάλι στους πίνακες αληθείας. Το BLAST σπάει τις ακολουθίες (νουκλεοτιδίων ή αμινοξέων) σε μικρότερες υπακολουθίες (μέσω κατακερματισμού-hashing και δεικτοδότησης-indexing) και προσπαθεί να τις επεκτείνει διαγωνίως, προς αριστερά και δεξιά. Ουσιαστικά δηλαδή προσπαθεί να εντοπίσει τμήματα διαγωνίων στους πίνακες, να τα επεκτείνει όσο μπορεί και τελικώς να επιλέξει το μεγίστου μήκους ζεύγος (Maximal Segment Pair-MSP) ισομηκών τμημάτων ακολουθιών που έχει το υψηλότερο άθροισμα βαρών - υψηλότερο αποτέλεσμα-σκορ (Highest Scoring Pair-HSP).

Τα όρια του MSP επιλέγονται έτσι ώστε να μεγιστοποιούν το συνολικό αποτέλεσμα και συνεπώς μπορεί να είναι οποιουδήποτε μήκους. Το αποτέλεσμα αυτό μας παρέχει ένα μέτρο ομοιότητας δύο ακολουθιών, αλλά μόνο σε τοπικό επίπεδο. Ωστόσο, στη μοριακή βιολογία κάποιος μπορεί να ενδιαφέρεται για όλες τις κοινές περιοχές σε δύο ακολουθίες και όχι μόνο για αυτήν με το MSP. Επομένως καθορίζουμε κάποιο μήκος ζευγών ως τοπικά μέγιστο αν το αποτέλεσμα του δε μπορεί να βελτιωθεί, αυξάνοντας ή μειώνοντας το μήκος. Το BLAST λοιπόν αναζητά όλα τα τοπικά μέγιστα με αποτέλεσμα πάνω από κάποιο όριο S.

Προκειμένου να επιταχυνθεί η διαδικασία αναζήτησης στη βάση δεδομένων ακολουθιών, το BLAST ελαχιστοποιεί το χρόνο απασχόλησής του σε περιοχές όπου η ομοιότητα με την υπό έρευνα ακολουθία φανερώνει μικρή πιθανότητα να ξεπεραστεί το όριο S. Αρχικώς, το BLAST αναζητά ποιες ακολουθίες της βάσης δεδομένων έχουν κάποιο κοινό ή σχεδόν κοινό (για παράδειγμα οι λέξεις MAT και MAN) τμήμα με την υπό έρευνα ακολουθία, που να ξεπερνά τουλάχιστον ένα προκαθορισμένο μήκος λέξης w και στη συνέχεια ψάχνει ποιο από αυτά τα ζεύγη τμημάτων έχει αποτέλεσμα ίσο ή μεγαλύτερο από ένα κατώφλι T. Κάθε τέτοια επιτυχία-χτύπημα (hit) επεκτείνεται για να καθοριστεί αν περιέχεται σε κάποιο ζεύγος τμημάτων με αποτέλεσμα ίσο ή μεγαλύτερο από το S. Το προκαθορισμένο (default) μήκος λέξης w για τις ακολουθίες αμινοξέων είναι ίσο με 3, ενώ για τις ακολουθίες νουκλεοτιδίων είναι ίσο με 11.

Ένα αριθμητικό παράδειγμα που θα βοηθούσε στην καλύτερη κατανόηση είναι το εξής: Έστω ότι έχουμε μία ακολουθία την οποία επιθυμούμε να συγκρίνουμε με μία βάση δεδομένων που περιέχει 100 ακολουθίες. Αυτές ελέγχονται από το πρόγραμμα και διαπιστώ-

νεται πως 40 ακολουθίες έχουν κοινό με την υπό έρευνα ακολουθία τμήμα μήκους μικρότερου του w, ενώ οι 60 υπόλοιπες έχουν κάποιο κοινό τμήμα με την υπό έρευνα μήκους μεγαλύτερου ή ίσου του w. Έστω στις 60 αυτές ακολουθίες ανιχνεύονται 150 τέτοια κοινά με την υπό έρευνα ακολουθία τμήματα.

Αυτά τα κοινά τμήματα ελέγχονται για να διαπιστωθεί ποια έχουν αποτέλεσμα (σκορ) μεγαλύτερο του ορισμένου κατωφλίου T. Έστω ότι 120 είναι αυτά. Τα 120 αυτά τμήματα ονομάζονται επιτυχίες-χτυπήματα (hits) και επεκτείνονται για να ευρεθούν όσα ανήκουν σε τμήματα που όταν συγκριθούν με την υπό έρευνα έχουν αποτέλεσμα-σκορ μεγαλύτερο από το όριο S. Έστω ότι αυτά είναι 20. Τότε η έξοδος του συστήματος BLAST θα είναι οι 20 αυτές επιτυχίες-χτυπήματα που θα απεικονιστούν και γραφικώς.

Όσο μεγαλύτερο είναι το κατώφλι T, τόσο αυξάνεται η πιθανότητα ένα ζεύγος τμημάτων με αποτέλεσμα τουλάχιστον S να περιέχει ένα ζεύγος λέξης με αποτέλεσμα τουλάχιστον T. Αντίθετα μια μικρή τιμή του T αυξάνει τον αριθμό των χτυπημάτων (hits) και συνεπώς μεγεθύνει το χρόνο εκτέλεσης του αλγορίθμου. Η παράμετρος T δηλαδή, υπαγορεύει την ταχύτητα αλλά και την ευαισθησία της έρευνάς μας, καθώς όσο αυξάνεται το T, αυξάνεται η ταχύτητα εκτέλεσης και μειώνεται η ευαισθησία.

Συνοψίζοντας λοιπόν, το BLAST εκτελείται σε τρία βήματα:

- Συντάσσεται η λίστα των λέξεων μήκους w που όταν συγκριθούν με τμήματα της εν λόγω ακολουθίας έχουν αποτέλεσμα τουλάχιστον T.
- Σαρώνεται η βάση δεδομένων για να ευρεθούν επιτυχίες-χτυπήματα (hits).
- Επεκτείνονται τα χτυπήματα για να αναζητηθούν ζεύγη τμημάτων με αποτελέσματα που υπερβαίνουν το όριο S.

Πρέπει να σημειωθεί βέβαια πως όσες ακολουθίες περάσουν επιτυχώς το τρίτο βήμα του BLAST, υπόκεινται σε στατιστική ανάλυση προκειμένου να υπολογιστεί η στατιστική σημασία τους (statistical significance), δηλαδή να υπολογιστεί το πόσο ουσιαστικό είναι ένα αποτέλεσμα.

Δύο πολύ σημαντικές παράμετροι που χρησιμοποιούνται για αυτόν το σκοπό στο πρόγραμμά αυτό, είναι οι τιμές E και p. Η τιμή προσδοκίας E (Expectation value) ισούται με τον αριθμό των ευθυγραμμίσεων με αποτελέσματα ίσα ή μεγαλύτερα από το S, που είναι αναμενόμενο να συμβούν με τυχαίο τρόπο σε μία αναζήτηση σε βάση δεδομένων. Ουσιαστικά λοιπόν η τιμή του E περιγράφει τον τυχαίο θόρυβο που υπάρχει σε ταιριάσματα δύο ακολουθιών. Όσο μικρότερο είναι το E ή όσο πιο κοντά στο 0, τόσο σημαντικότερη είναι η ευθυγράμμισή μας. Αναλυτικότερα:

- Αν η τιμή του E είναι μικρότερη από 1×10^{-50} τότε το χτύπημα είναι πολύ όμοιο με την ακολουθία-ερώτηση και είναι πολύ πιθανό να είναι εξελικτικά συσχετισμένο με αυτήν.

- Αν η τιμή του E είναι μεταξύ των τιμών 1×10^{-50} και 1×10^{-2} τότε το χτύπημα έχει κάποια σχέση με την ακολουθία και ίσως να είναι συσχετισμένο με αυτήν. Είναι πιθανότερο πάντως να ανήκουν στην ίδια οικογένεια ή να έχουν στενά συνδεδεμένους λειτουργικούς τομείς.
- Αν η τιμή είναι μεταξύ του 1×10^{-2} και του 1, τότε το χτύπημα έχει μικρή πιθανότητα να είναι συσχετισμένο με την ακολουθία μας και μπορεί να συνεπάγεται μια μακρινή εξελικτική σχέση.
- Αν η τιμή είναι πάνω από 1, τότε το χτύπημα δεν είναι στενά συσχετισμένο με την ακολουθία μας.

Εξαίρεση στα παραπάνω εμφανίζεται συχνά, σε συγκρίσεις μικρού μήκους ακολουθιών. Στη σύγκριση λοιπόν μικρού μήκους ακολουθιών, ακόμα και αν αναφερόμαστε σε σχεδόν ταυτόσημες ακολουθίες, η τιμή του E μπορεί να είναι σχετικά υψηλή και άρα να μας οδηγεί ψευδώς στο συμπέρασμα πως οι ακολουθίες είναι μεταξύ τους ασυσχέτιστες. Αυτό συμβαίνει επειδή στον υπολογισμό του E λαμβάνεται υπόψιν το μήκος της υπό έρευνας ακολουθίας. Η τιμή του E υπολογίζεται από τον τύπο: $E = K \times m \times n \times e^{-\lambda S}$, όπου K και λ στατιστικές παράμετροι, m και n τα μήκη των ακολουθιών (ακολουθίας-ερώτησης και της εκάστοτε συγκρινόμενης με αυτήν) και S το σκορ της κάθε επεξεργασίας. Όπως φαίνεται και από τον τύπο αυτόν, η τιμή του E μειώνεται εκθετικά ως προς το σκορ S.

Η τιμή p εκφράζει την πιθανότητα να συμβεί μία ευθυγράμμιση, με αποτέλεσμα ίσο ή μεγαλύτερο από κάποιο ζητούμενο. Η τιμή αυτή υπολογίζεται συσχετίζοντας το αποτέλεσμα της παρατηρούμενης στοίχισης με την προσδοκώμενη κατανομή των HSPs αποτελεσμάτων συγκρίσεων τυχαίων ακολουθιών (με ίδιο μήκος και σύνθεση με την υπό έρευνα ακολουθία) με τη βάση δεδομένων. Οι πιο σημαντικές τιμές p θα είναι αυτές κοντά στο 0. Οι τιμές E και p είναι ουσιαστικά δύο διαφορετικές μέθοδοι αναπαράστασης της σημασίας μας στοίχισης. Η σχέση μεταξύ p και E δίνεται από τον τύπο $p = 1 - e^{-E}$. Μάλιστα οι πολύ μικρές (και συνεπώς σημαντικές) τιμές του E προσεγγίζουν τις τιμές του p (Πίνακας 3.7).

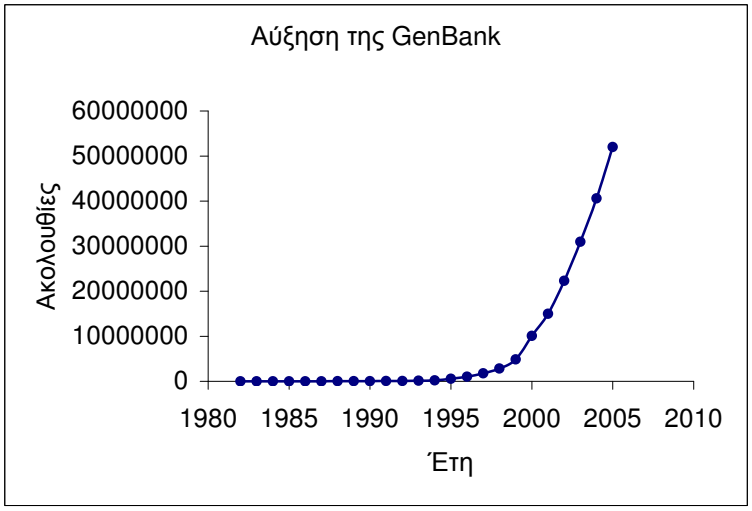
E	p
10	0.99995460
5	0.99326205
2	0.86466472
1	0.63212056
0.1	0.09516258 $\simeq$ 0.1
0.05	0.4877058 $\simeq$ 0.05
0.001	0.00099950 $\simeq$ 0.001
0.0001	0.000100

Πίνακας 3.7: E και p στο BLAST

Πρέπει να σημειωθεί πως, αν και οι παράμετροι E και p είναι αμφότερες μέτρα απεικόνισης της σημασίας μίας στοίχισης, η παράμετρος E είναι πολύ πιο ευανάγνωστη για τιμές μεγαλύτερες του 1, σε αντίθεση με την παράμετρο p (όπως φαίνεται και στον Πίνακα 3.7).

Επίσης χρησιμοποιούνται οι στατιστικές παράμετροι λ και K. Η παράμετρος λ θεωρείται ως φυσική κλίμακα για το σύστημα των αποτελεσμάτων (scoring system), ενώ η παράμετρος K ως φυσική κλίμακα για το μέγεθος του χώρου έρευνας (search space size). Οι τιμές αμφοτέρων των παραμέτρων αυτών χρησιμοποιούνται στη μετατροπή των ακατέργαστων αποτελεσμάτων (raw score) σε αποτελέσματα σε bits (bit scores). Η μετατροπή αυτή γίνεται με τον τύπο: $S(bits) = \frac{\lambda \times S - \ln K}{\ln 2}$. Σε αυτήν την περίπτωση η τιμή E υπολογίζεται από τον τύπο $E = m \times n \times 2^{-S(bits)}$, όπου S είναι το εκάστοτε σκορ και m, n είναι τα μήκη των αλληλουχιών. Η μετατροπή σε bits είναι επιθυμητή επειδή τα αποτελέσματα σε bits κάνουν ευκολότερη τη σύγκριση αποτελεσμάτων μεταξύ διαφορετικών ερευνών. Αυτό συμβαίνει γιατί τα αποτελέσματα σε bits είναι κανονικοποιημένα (normalized), όσον αφορά τη χρήση διαφορετικών πινάκων συγκρίσεως και διαφορετικών μεγεθών βάσεων δεδομένων. Εάν δεν κανονικοποιηθούν τα αποτελέσματα και παραμείνουν ακατέργαστα, τότε είναι μικρής σημασίας εάν δεν συνοδεύονται από τις στατιστικές παραμέτρους λ και K.

Λόγω του γεγονότος ότι ο ρυθμός αλλαγής του μεγέθους των βάσεων δεδομένων μας, υπερβαίνει τις ταχύτητες των επεξεργαστών, οι υπολογιστές που εκτελούν το BLAST υπόκεινται σε αυξανόμενο φορτίο και χρόνο εκτέλεσης. Για παράδειγμα ο ρυθμός αύξησης της τράπεζας ακολουθιών νουκλεοτιδίων GenBank απεικονίζεται στο Σχήμα 3.8.



Σχήμα 3.8: Αύξηση της GenBank

Ωστόσο ο συνδυασμός διαφόρων αλγοριθμικών ιδεών επέτρεψε τη δημιουργία νεότερων εκδόσεων του BLAST που χρησιμοποιούν τεχνικές που στοχεύουν στη βελτίωση της

ευαισθησίας και την επαυξημένη ταχύτητα του αλγορίθμου. Παρακάτω αναφέρονται οι σημαντικότερες τρεις βελτιώσεις που εισήχθησαν.

3.4.1.1 Βελτιώσεις του BLAST

- **BLAST με χρήση κενών**

Αρχικώς λοιπόν, προστέθηκε η ικανότητα του προγράμματος να γεννά ευθυγραμμίσεις με κενά. Η αρχική έκδοση του BLAST (original BLAST) συχνά βρίσκει μερικές ευθυγραμμίσεις με έρευνα σε μία μόνο βάση δεδομένων, οι οποίες συνυπολογιζόμενες παράγουν στατιστικά σημαντικό αποτέλεσμα. Ωστόσο η παράβλεψη έστω και μίας εξ' αυτών των στοιχίσεων θέτει σε κίνδυνο το συνδυαζόμενο αποτέλεσμα. Εισάγοντας λοιπόν έναν αλγόριθμο που γεννά ακολουθίες με κενά (gapped alignments), γίνεται απαραίτητη η εύρεση μιας μονάχα στοίχισης, παρά η εύρεση όλων των χωρίς κενά-στοιχίσεων (ungapped) και η τελική υπαγωγή τους σε ένα σημαντικό αποτέλεσμα. Αυτό επιτρέπει την αύξηση της παραμέτρου T, και κατά συνέπεια την αύξηση της ταχύτητας του αρχικού σαρώματος της βάσης δεδομένων. Ο νέος αλγόριθμος με εισαγωγή κενών (Gapped BLAST), χρησιμοποιεί το δυναμικό προγραμματισμό, όπως είδαμε προηγουμένως, με σκοπό να επεκτείνει ένα κεντρικό ζεύγος στοιχισμένων τμημάτων και προς τις δύο κατευθύνσεις.

Ο αλγόριθμος με χρήση κενών (Gapped BLAST) τρέχει κατά προσέγγιση τρεις φορές γρηγορότερα από τον πρωτότυπο (original BLAST) και στη συντριπτική πλειοψηφία των περιπτώσεων ανιχνεύει μεγαλύτερο αριθμό στατιστικά σημαντικών παρατάξεων.

- **BLAST και μέθοδος δύο χτυπημάτων-επιτυχιών**

Η δεύτερη βελτίωση του βασικού αλγορίθμου ήταν η προσθήκη της μεθόδου των δύο χτυπημάτων-επιτυχιών (two-hit method). Η κεντρική ιδέα του αλγορίθμου BLAST, όπως προείπαμε, είναι ότι μια στατιστικά σημαντική ευθυγράμμιση ακολουθιών είναι πιθανόν να περιέχει ένα HSP. Το BLAST αρχικά σαρώνει τη βάση δεδομένων για λέξεις μήκους w, που έχουν αποτέλεσμα τουλάχιστον T όταν συγκριθούν με κάποια λέξη της υπό έρευνα ακολουθίας. Κάθε στοιχισμένο ζεύγος λέξεων που ικανοποιεί αυτή τη συνθήκη καλείται επιτυχία-χτύπημα. Στη συνέχεια, κάθε τέτοιο χτύπημα επεκτείνεται για να ελεγχθεί εάν εμπεριέχεται σε ευθυγράμμιση με αποτέλεσμα αρκετά μεγάλο, ώστε να αναφερθεί. Το βήμα αυτό επέκτασης είναι αρκετά δαπανηρό υπολογιστικά και μπορεί να αντιστοιχεί σε ποσοστό πάνω από 90% του συνολικού χρόνου εκτέλεσης του BLAST. Είναι λοιπόν επιθυμητή η μείωση του αριθμού των εκτελούμενων επεκτάσεων.

Ο βελτιωμένος λοιπόν αλγόριθμος που χρησιμοποιείται σήμερα, βασίζεται στην παρατήρηση πως ένα ενδιαφέρον HSP είναι πολύ μακρύτερο από ένα απλό ζεύγος λέξεων και άρα μπορεί να συνεπάγεται πολλαπλά χτυπήματα στην ίδια διαγώνιο και σε σχετικά μικρή απόσταση μεταξύ τους. Τότε επιλέγουμε ένα μήκος παραθύρου A και προκαλούμε μία επέκταση μόνον

όταν δύο μη επικαλυπτόμενα χτυπήματα στην ίδια διαγώνιο βρίσκονται σε απόσταση μικρότερη ή ίση του A Κάθε χτύπημα που επικαλύπτει το αμέσως προηγούμενό του, αγνοείται. Για αποδοτική εκτέλεση αποθηκεύουμε σε έναν πίνακα την ταυτότητα του εκάστοτε πιο πρόσφατου χτυπήματος (συντεταγμένες) για κάθε διαγώνιο.

Επειδή χρειαζόμαστε δύο χτυπήματα αντί για ένα για να προκαλέσουμε την επέκταση, η παράμετρος κατωφλίου T πρέπει να μικρύνει για να διατηρηθεί η απαιτούμενη ευαισθησία. Η συνέπεια της μείωσης του T θα είναι η εύρεση πολύ περισσοτέρων χτυπημάτων, εκ των οποίων όμως, μόνο ένα μικρό κλάσμα θα έχει ένα συσχετισμένο δεύτερο χτύπημα και θα προκαλέσει μία επέκταση. Η συντριπτική πλειοψηφία των χτυπημάτων θα απορριφθεί μετά από μία σειρά μικρών ελέγχων για την κατάλληλη διαγώνιο, για την επικάλυψη ή όχι των χτυπημάτων και για την τήρηση της απόστασης του παραθύρου. Εμπειρικά έχει παρατηρηθεί πως ο υπολογισμός που γλιτώνουμε με το μικρότερο αριθμό επεκτάσεων υπερβαίνει κατά πολύ τον επιπλέον υπολογισμό της επεξεργασίας των πολυάριθμων χτυπημάτων.

Προκειμένου να συγκρίνουμε τις σχετικές ταχύτητες των μεθόδων του ενός και των δύο χτυπημάτων, σημειώνουμε πως η μέθοδος των δύο χτυπημάτων γεννά περίπου 3.2 φορές περισσότερα χτυπήματα, αλλά 0.14 φορές λιγότερες επεκτάσεις. Επειδή η απόφαση για το αν ένα χτύπημα πρέπει να επεκταθεί ή όχι, καταλαμβάνει το $\frac{1}{9}$ του χρόνου σε σχέση με το χρόνο που χρειάζεται μία επέκταση, η διαδικασία επεξεργασίας των χτυπημάτων είναι δύο φορές γρηγορότερη στη μέθοδο των δύο χτυπημάτων, από την αντίστοιχη στη μέθοδο του ενός χτυπήματος.

- **PSI-BLAST**

Οι έρευνες του BLAST μπορούν να είναι επαναλαμβανόμενες, με έναν πίνακα αποτελεσμάτων συγκεκριμένης θέσης PSSM (position-specific score matrix), που δημιουργείται από σημαντικές ευθυγραμμίσεις που γίνονται στο i-οστό βήμα και χρησιμοποιείται στο (i+1)-οστό βήμα. Μέθοδοι αναζήτησης μοτίβων ή προφίλ είναι συχνά πιο ευαίσθητες από τις μεθόδους σύγκρισης ακολουθιών κατά ζεύγη, όσον αφορά σε ακολουθίες με μακρινή σχέση μεταξύ τους. Ωστόσο η δημιουργία ενός συνόλου από μοτίβα ή προφίλ που περιγράφουν μία οικογένεια πρωτεϊνών και η αναζήτηση σε μια βάση δεδομένων με αυτά, τυπικά περιλαμβάνει τη χρήση πολλών διαφορετικών προγραμμάτων και σημαντική παρέμβαση του χρήστη σε διάφορα στάδια της διαδικασίας. Ο αλγόριθμος του BLAST εύκολα γενικεύεται, για να χρησιμοποιήσει έναν πίνακα αποτελεσμάτων συγκεκριμένης θέσης, αντί μιας ακολουθίας-ερώτησης και ενός σχετικού πίνακα υποκατάστασης στοιχείων. Συνεπώς αυτοματοποιήθηκε η διαδικασία γέννησης ενός τέτοιου πίνακα ως εξόδου μιας έρευνας του BLAST και προσαρμόστηκε ο αλγόριθμος προκειμένου να δέχεται αυτόν τον πίνακα σαν είσοδο. Το συνεπαγόμενο επαναλαμβανόμενο BLAST συγκεκριμένης θέσης (PSI-BLAST,

Position Specific Iterated BLAST) μπορεί να μην είναι τόσο ευαίσθητο όσο τα καλύτερα προγράμματα αναζήτησης μοτίβων, αλλά η ταχύτητα και η ευκολία χρήσης του κάνουν δυνατή την ευρύτερη χρήση του.

Το PSI-BLAST συνήθως καταναλώνει αξιοσημείωτα σημαντικότερο χρόνο από τον Gapped BLAST, πρωτίστως λόγω του χρόνου που απαιτείται για την κατασκευή του πίνακα αποτελεσμάτων συγκεκριμένης θέσης από το μεγάλο αριθμό των σημαντικών ευθυγραμμίσεων που έχει ήδη ανιχνεύσει το BLAST. Αντιθέτως σε περιπτώσεις ερευνών όπου οι σημαντικές ευθυγραμμίσεις είναι λίγες, κάθε επανάληψη του PSI-BLAST εκτελείται σε χρόνο περίπου ίσο με αυτόν του BLAST.

Στη συνέχεια θα παραθέσουμε αρκετά παραδείγματα εκτέλεσης του αλγορίθμου BLAST στις διάφορες εκδόσεις του, Gapped BLAST, Original (ungapped) BLAST και PSI-BLAST.

3.4.1.2 *Εκτελέσεις του BLAST*

Η εκτέλεση του αλγορίθμου έγινε μέσω της διαδικτυακής υπηρεσίας BLAST του ελβετικού ινστιτούτου Βιοπληροφορικής (Swiss Institute of Bioinformatics-[Che06]), αλλά και μέσω της ιστοσελίδας του εθνικού κέντρου πληροφοριών βιοτεχνολογίας (National Center for Biotechnology Information-[Ncb06]).

Στους διαδικτυακούς τόπους που χρησιμοποιούνται, παρέχονται πολλές επιλογές και δυνατότητες, σημαντικότερες των οποίων είναι οι παρακάτω:

- Υποβολή της ακολουθίας-ερώτησης είτε με τη μορφή απλού κειμένου (text format), είτε με τον αριθμό ταυτότητας (ID number) με τον οποίον μια ακολουθία είναι αποθηκευμένη στη βάση δεδομένων (αν αυτός είναι γνωστός), ή με τη μορφή Fasta (Fasta Format-η οποία ξεκινά με μία γραμμή περιγραφής με πρώτο σύμβολο το σύμβολο του μεγαλύτερου (>) και στη συνέχεια ακολουθείται από γραμμές δεδομένων ακολουθίας).
- Περιορισμός της έρευνας μας σε συγκεκριμένο χώρο. Αυτό γίνεται είτε επιλέγοντας τη βάση δεδομένων των ακολουθιών με τις οποίες θα συγκριθεί η υποβαλλόμενη σε έρευνα, είτε περιοριζόμενοι σε ομάδες βάσεων δεδομένων, που αφορούν μόνο συγκεκριμένα είδη, όπως π.χ. τον άνθρωπο.
- Ρύθμιση παραμέτρων της έρευνας που έχουν προαναφερθεί, όπως το μήκος λέξης w καθώς και το κατώφλι T.
- Εμφάνιση των απαραιτήτων πληροφοριών για τις αλληλουχίες που παράγουν σημαντικά αποτελέσματα, όπως τα αποτελέσματα (σκορ), το είδος στο οποίο αυτές ανήκουν, αν είναι ομόλογες κάποιας άλλης, καθώς και το μήκος τους.
- Γραφική απεικόνιση της ευθυγράμμισης, κάτι που βοηθά στην καλύτερη κατανόηση της μεθόδου.

- Δυνατότητα επιλογής της μορφής παρουσίασης των αποτελεσμάτων (output format), που μπορεί να είναι HTML, XML και άλλες.

Πρέπει να σημειωθεί πως είναι αρκετά πιθανόν πολλές εκ των εκτελέσεων ενός χρήστη να μην έχουν αποτέλεσμα και να εμφανίζεται το μήνυμα “No Significant similarity found”. Αυτό μπορεί να οφείλεται σε λόγους όπως:

- Η ακολουθία που θέτουμε προς έρευνα είναι μικρού μήκους.
- Η ακολουθία προς έρευνα είναι χαμηλής πολυπλοκότητας. (Ως χαμηλής πολυπλοκότητας ακολουθία-low complexity sequence, ορίζεται μία ακολουθία με περιοχές συνεχόμενων ομοίων συμβόλων. Παραδείγματα ακολουθιών χαμηλής πολυπλοκότητας είναι οι “PPCDPP PPPKDKKKKDDGPP” (ακολουθία αμινοξέων) και η “AAATAAAAAAAATAAAAAAT” (ακολουθία νουκλεοτιδίων).
- Ο χρήστης εισάγει λάθος τύπου ακολουθία ως είσοδο του προγράμματος. Για παράδειγμα εάν ο χρήστης εισάγει μία νουκλεοτιδική ακολουθία και προσπαθήσει να χρησιμοποιήσει το πρόγραμμα blastp (το οποίο χειρίζεται ακολουθίες αμινοξέων, βλέπε Πίνακα 4.6) τότε θα έχουμε εμφάνιση του μηνύματος σφάλματος.
- Ο χρήστης κάνει άλλα σφάλματα, όπως η εισαγωγή λάθος συμβόλων, ή η λάθος επιλεγμένη μορφή (format) της ακολουθίας εισόδου.

Στη συνέχεια ακολουθούν τα παραδείγματα εκτέλεσης του BLAST:

- $1^{ο}$ παράδειγμα

Στο πρώτο παράδειγμα η εκδοχή που χρησιμοποιούμε είναι η blastp με χρήση κενών, που συγκρίνει την υπό έρευνα ακολουθία αμινοξέων έναντι βάσης δεδομένων ακολουθιών πρωτεϊνών. Η πρώτη υπό έρευνα ακολουθία (query) αμινοξέων που χρησιμοποιήσαμε είναι η MSQVQVQVQNPSAALSGSQILNKNQSLLSQPLMSIPSTTSSLPSENAGRPIQNSALPSAS και την οποία εισαγάγαμε σε μορφή κειμένου (text). Χρησιμοποιήσαμε τη πρωτεϊνική βάση δεδομένων swissprot και σε περίπου 10 δευτερόλεπτα λάβαμε την απάντησή μας.

BLAST2.0 query receipt | Program: blastp | Database: swiss | Format: plain_text

Sequence:

MSQVQVQVQNPSAALSGSQILNKNQSLLSQPLMSIPSTTSSLPSENAGRPIQNSALPSAS

Processing, please wait...

blastall.remote -s sib-blast2.unil.ch -p blastp -d swiss wwwtmp/sq.21466.txt -M blosum62 -K 0 -e 10.0 -F T -G def -E def -X def -v 50 -b 50 -g T |

Here are your search results:

SIB BLAST network server version 1.7 of June 24, 2005

compiled by GNU C version 3.4.4 20050721 (Red Hat 3.4.4-2)

compiled on Dec 19 2005.

Welcome to the SIB BLAST Network Service (blast01.vital-it.ch)

Swiss Institute of Bioinformatics (SIB)

Ludwig Institute for Cancer Research (LICR)

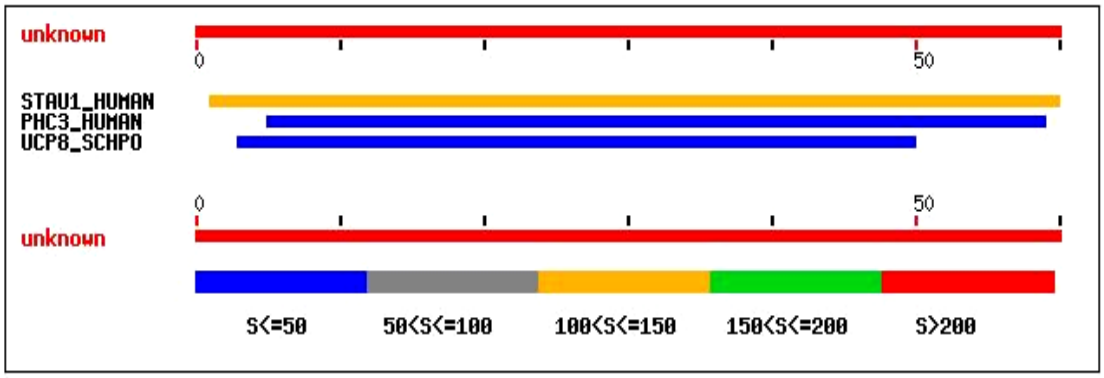
Swiss Institute for Experimental Cancer Research (ISREC)

Query= unknown (60 letters)

Database: swiss

217,551 sequences; 79,825,379 total letters

Searching.....done



Sequences producing significant alignments:	(bits)Score	E
sp|O95793|STAU1_HUMAN (STAU1)Double-stranded RNA-binding protein...	115	4e-26
sp|Q8NDX5|PHC3_HUMAN (PHC3)Polyhomeotic-like protein 3 (hPH3) (H...	30	2.3
sp|O94685|UCP8_SCHPO (ucp8)UBA domain-containing protein 8.[Schi...	28	8.9

- sp|O95793|STAU1_HUMAN (STAU1)Double-stranded RNA-binding protein

Staufen homolog 1.[Homo sapiens]

Length = 577

Score = 115 bits (287), Expect = 4e-26

Identities = 60/60 (100%), Positives = 60/60 (100%)

Query: 1

MSQVQVQVQNPSAALSGSQILNKNQSLLSQPLMSIPSTTSSLPSENAGRPIQNSALPSAS 60

MSQVQVQVQNPSAALSGSQILNKNQSLLSQPLMSIPSTTSSLPSENAGRPIQNSALPSAS

Sbjct: 1

MSQVQVQVQNPSAALSGSQILNKNQSLLSQPLMSIPSTTSSLPSENAGRPIQNSALPSAS 60

- sp|Q8NDX5|PHC3_HUMAN (PHC3)Polyhomeotic-like protein 3 (hPH3)

(Homolog of polyhomeotic 3) (Early development regulatory protein 3).[Homo sapiens]

Length = 983

Score = 29.6 bits (65), Expect = 2.3

Identities = 17/55 (30%), Positives = 29/55 (52%), Gaps = 1/55 (1%)

Query: 5 QVQVQNPSAALSGSQILNKNQSLLSQPLMSIPSTTSSLPSENAGRPIQNSALPSA 59

Q+ + +P + +S Q++ + Q QP+ ST PS++ P+QN LP A

Sbjct: 310 QIPLHSPPSKVSHHQLILQQQQQQIQPITLQNSTQDPPPSQHC-IPLQNHGLPPA 363

- sp|O94685|UCP8_SCHPO (ucp8)UBA domain-containing protein 8.[Schizosaccharomyces pombe]

Length = 884

Score = 27.7 bits (60), Expect = 8.9

Identities = 18/48 (37%), Positives = 22/48 (45%), Gaps = 1/48 (2%)

Query: 3 QVQVQVQNPSAALSGSQILNKNQSLLSQPLMSIPSTTSSLPSENAGRP 50

Q QV QN + SQ S Q + IP TT S+P + AG P

Sbjct: 468 QDQVSTQNQANTADISQNTESGSSTQGQ-MFGIPPTTQSIPLKMAGFP 514

Database: swiss

Posted date: May 10, 2006 5:01 AM

Number of letters in database: 79,825,379

Number of sequences in database: 217,551

Lambda K H

0.300 0.113 0.286

Gapped

Lambda K H

0.267 0.0410 0.140

Matrix: BLOSUM62

Gap Penalties: Existence: 11, Extension: 1

Number of Hits to DB: 7,792,908

Number of Sequences: 217551

Number of extensions: 212294

Number of successful extensions: 493

Number of sequences better than 10.0: 3

Number of HSP's better than 10.0 without gapping: 1

Number of HSP's successfully gapped in prelim test: 2

Number of HSP's that attempted gapping in prelim test: 492

Number of HSP's gapped (non-prelim): 3

Length of query: 60

Length of database: 79,825,379

Effective HSP length: 33

Effective length of query: 27

Effective length of database: 72,646,196

Effective search space: 1961447292

Effective search space used: 1961447292

T: 11, A: 40

Στα αποτελέσματα της έρευνας παρατηρούμε αρχικώς διάφορες πληροφορίες για τις παραμέτρους που θέσαμε, όπως π.χ. για τη βάση δεδομένων που επιλέξαμε. Η βάση δεδομένων swissprot λοιπόν, περιέχει 217,551 ακολουθίες και συνολικά 79,825,379 γράμματα-σύμβολα (αμινοξέα), ενώ ο πίνακας που χρησιμοποιήθηκε ήταν ο BLOSUM62. Οι επιτυχίες-χτυπήματα στη βάση δεδομένων μας ήταν 7,792,908 , οι επιτυχημένες επεκτάσεις ήταν 493 και τελικώς τα ζεύγη τμημάτων με αποτέλεσμα-σκορ μεγαλύτερο του 10 ήταν 3. Στη γραφική απεικόνιση των στοιχίσεων ανάλογα με το σκορ S χρωματίζεται και η ευθυγράμμιση. Για παράδειγμα η ακολουθία STAU1_HUMAN έχει σκορ 115 και χρωματίζεται με το πορτοκαλί χρώμα, με το οποίο θα χρωματιζόταν και οποιαδήποτε άλλη ακολουθία με σκορ S στην περιοχή μεταξύ 100 και 150 bits.

Για κάθε μία από τις τρεις ακολουθίες σημειώνονται ονομαστικές και δομικές πληροφορίες, το μήκος της, το πλήθος των ομοίων συμβόλων σε σχέση με την υπό ακολουθία ερώτηση (Identities), το πλήθος των αμινοξέων που σε σύγκριση με τα αμινοξέα της ακολουθίας μας έχουν θετικά κόστη (Positives, που φυσικά περιλαμβάνουν και τα όμοια σύμβολα εφόσον η κύρια διαγώνιος στον πίνακα Blosum62 έχει θετικά κόστη, βλέπε πίνακα 4.4) καθώς και το πλήθος των κενών που χρησιμοποιήθηκαν. Σημειώνεται επίσης το αποτέλεσμα-σκορ, καθώς και η τιμή προσδοκίας E, αλλά και οι τιμές των στατιστικών παραμέτρων λ και K. Παραδείγματος χάριν, η δεύτερη ακολουθία έχει μήκος 983 αμινοξέων, το σκορ της είναι 65 (ή 29,6 bits), η τιμή προσδοκίας είναι 2.3 , τα όμοια αμινοξέα είναι 17 , ενώ εκείνα που στη σύγκριση έχουν θετικά κόστη είναι 29 , και έγινε χρήση ενός μόνο κενού. Στην εκτέλεση αυτή η μόνη επιτυχία-χτύπημα που είναι στενά συσχετισμένη με την υπό έρευνα ακολουθία είναι η STAU1_HUMAN καθώς για αυτήν η τιμή προσδοκίας είναι 4×10^{-26}. Οι άλλες δύο επιτυχίες με τιμές E 2.3 και 8.9 δεν είναι στενά συσχετισμένες με την ακολουθία μας.

- $2^{ο}$ παράδειγμα

Η επόμενη ακολουθία που χρησιμοποιήσαμε ήταν η MASSCSVQELGHRRKKPTVTHDW VRGPEHSHFVEHLHERPKCKDYKVEESGYAGFIEVYFKNKEEPRFYDLFLHLEGHPH LRCE με χρήση του προγράμματος blastp με χρήση κενών, η οποία και προκάλεσε την ανίχνευση πολυάριθμων ακολουθιών με σημαντικά αποτελέσματα, οπότε και θα

παραθέσουμε μόνο ορισμένες εξ'αυτών. Ο χρόνος επεξεργασίας ήταν περί τα 15 δευτερόλεπτα και η πρωτεϊνική βάση δεδομένων είναι η Non Redundant.

BLAST2.0 query receipt

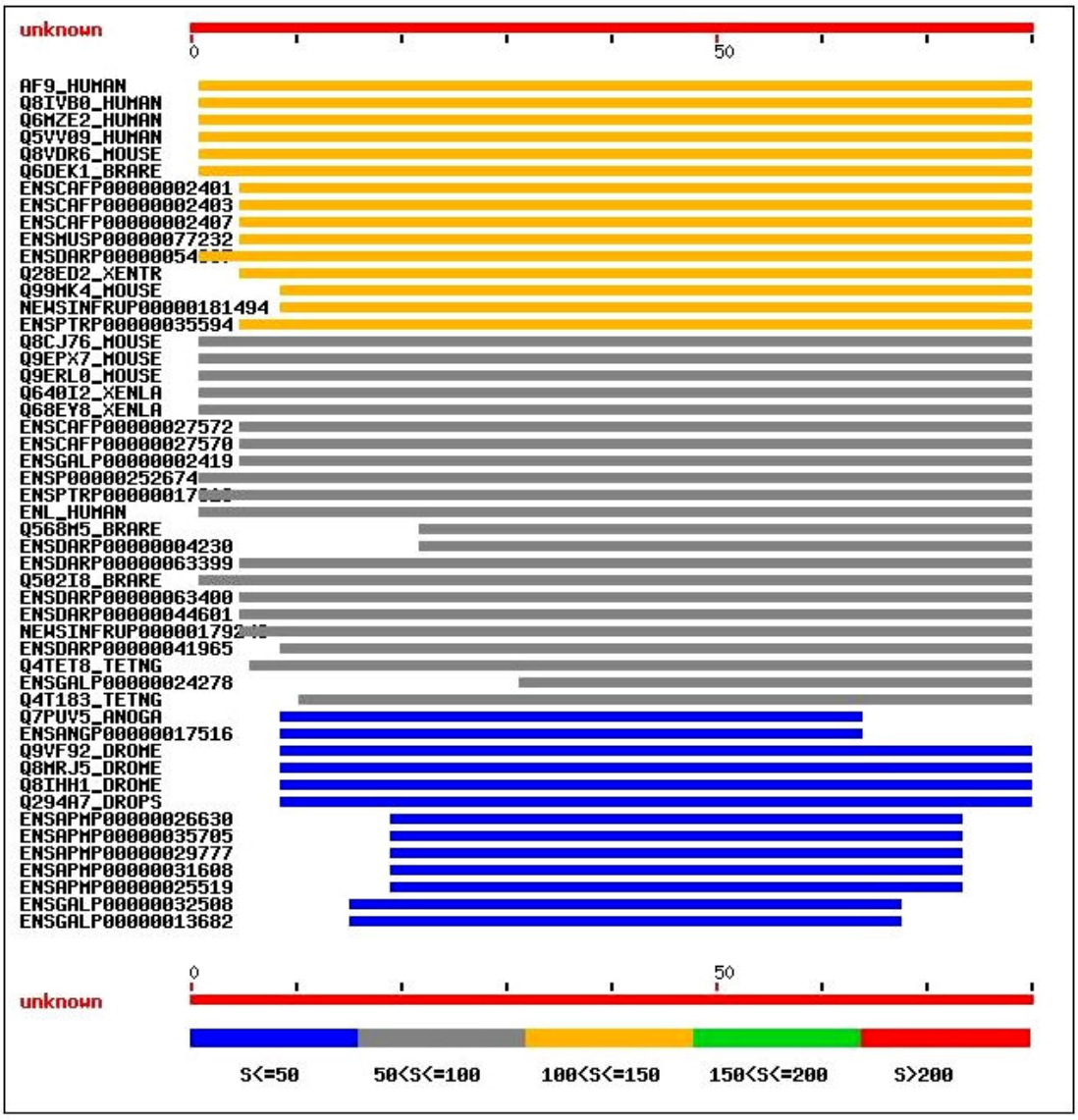
Program: blastp | Database: nrcrc | Format: plain_text

Sequence:

MASSCSVQELGHRRKKPTVTHDWVRGPEHSHFVEHLHERPKCKDYKVEESGYAGFIEVYFKN
KEEPRFYDLFLHLEGHPHLRCE

Here are your search results: Query= unknown(84 letters)

Searching.....done



Sequences producing significant alignments: Score(bits) E

sp|P42568|AF9_HUMAN (MLLT3)Protein AF-9 (ALL1 fused gene from … 120 2e-26

tr|Q8IVB0|Q8IVB0_HUMAN (MLLT3)Myeloid/lymphoid or mixed-lineage ...119 4e-26

ens|ENSCAFP00000002401 pep:novel chromosome:BROADD1:11:412770… 111 7e-24

tr|Q640I2|Q640I2_XENLA (LOC494666)LOC494666 protein.[Xenopus lae... 96 4e-19

ens|ENSANGP00000017516 pep:novel chromosome:AgamP3:2R:1610037... 49 7e-05

tr|Q294A7|Q294A7_DROPS (Dpse\GA18521)GA18521-PA (Fragment).[Dro... 46 5e-04

ens|ENSAPMP00000026630 pep:novel group:AMEL2.0:Group13:2215530:2... 39 0.055

- sp|P42568|AF9_HUMAN (MLLT3)Protein AF-9 (ALL1 fused gene from chromoso-

-me 9 protein) (Myeloid/lymphoid or mixed-lineage leukemia translocated to chromosome 3 protein). [Homo sapiens]

Length = 568

Score = 120 bits (300), Expect = 2e-26

Identities = 83/120 (69%), Positives = 83/120 (69%), Gaps = 36/120 (30%)

Query: 1 MASSCSVQ---ELGHR---RKKPTV---THDW---VRGPEHS---HFVE----HLHE--- 38

MASSCSVQ ELGHR RKKPTV THDW VRGPEHS HFVE HLHE

Sbjct: 1

MASSCSVQVKLELGHRAQVRKKPTVEGFTHDWMVFVRGPEHSNIQHFVEKVVFHLHESFP 60

Query: 39 RPK--CKD--YKVEESGYAGFI---EVYFKNKEEPR----FYDLFLHLEGHP---HLRCE 84

RPK CKD YKVEESGYAGFI EVYFKNKEEPR YDLFLHLEGHP HLRCE

Sbjct: 61

RPKRVCKDPPYKVEESGYAGFILPIEVYFKNKEEPRKVRFDYDLFLHLEGHPPVNHLRCE 120

- ens|ENSANGP00000017516 pep:novel chromosome:AgamP3:2R:16100379:161036

02:-1 gene:ENSANGG00000015027 transcript:ENSANGP00000017516

Length = 920

Score = 48.5 bits (114), Expect = 7e-05

Identities = 35/85 (41%), Positives = 49/85 (57%), Gaps = 26/85 (30%)

Query: 9 ELGHR---RKKPTV---THDW------VRGPEHSHFVE----HLHER-PKCK------DY 45

E+GH + +PT THDW + G + SHFV+ +LHE PK K Y

Sbjct: 8

EIGHVASVKSRPTAEGYTHDWELFVRGLDGTDISHFVDKVVFNLHESFPKPKRVFKEPPY 67

Query: 46 KVEESGYAGFI---EVYFKNKEEPR 67

V+E+GYAGFI E+YFKN+++P+

Sbjct: 68 LVKEAGYAGFILPVEIYFKNRDDPK 92

Database: swiss_nr

Posted date: May 10, 2006 5:04 AM

Number of letters in database: 79,815,504

Number of sequences in database: 217,519

. . .

Lambda K H

0.321 0.139 0.457

Gapped

Lambda K H

0.267 0.0410 0.140

Matrix: BLOSUM62

Gap Penalties: Existence: 11, Extension: 1

Number of Hits to DB: 221,535,737

Number of Sequences: 3249763

Number of extensions: 8051184

Number of successful extensions: 14084

Number of sequences better than 10.0: 50

Number of HSP's better than 10.0 without gapping: 6

Number of HSP's successfully gapped in prelim test: 46

Number of HSP's that attempted gapping in prelim test: 13982

Number of HSP's gapped (non-prelim): 52

length of query: 84

length of database: 1,115,806,840

effective HSP length: 54

effective length of query: 30

effective length of database: 940,319,638

effective search space: 28209589140

effective search space used: 28209589140

T: 11 A: 40

Στην εκτέλεση αυτή 50 ακολουθίες παρήγαγαν σημαντικά αποτελέσματα, ορισμένες εκ των οποίων αναλύονται. Για παράδειγμα η ακολουθία ens|ENSANGP00000017516 της βάσης δεδομένων είχε σκορ 48.5 bits (ή 114) και τιμή E ίση με 7e-05. Άρα από την τιμή του E συμπεραίνουμε πως η ακολουθία αυτή έχει κάποια σχέση με την υπό έρευνα ακολουθία και ίσως να είναι συσχετισμένη με την τελευταία.

3ο παράδειγμα

Στο επόμενο παράδειγμα χρησιμοποιούμε το πρόγραμμα blastn, που συγκρίνει την ακολουθία-ερώτηση νουκλεοτιδίων έναντι βάσης δεδομένων ακολουθιών νουκλεοτιδίων. Με χρήση κενών και με αναζήτηση στη βάση δεδομένων ακολουθιών νουκλεοτιδίων EMBL, για την ακολουθία ATGCGTAGTCAGATAATACAGAAAACACTGATATGTCGATGCA έχουμε τα παρακάτω αποτελέσματα μετά από 12 δευτερόλεπτα:

Program: blastn | Database: embl-sp | Format: plain_text

Sequence: ATGCGTAGTCAGATAATACAGAAAACACTGATATGTCGATGCA

Processing, please wait...

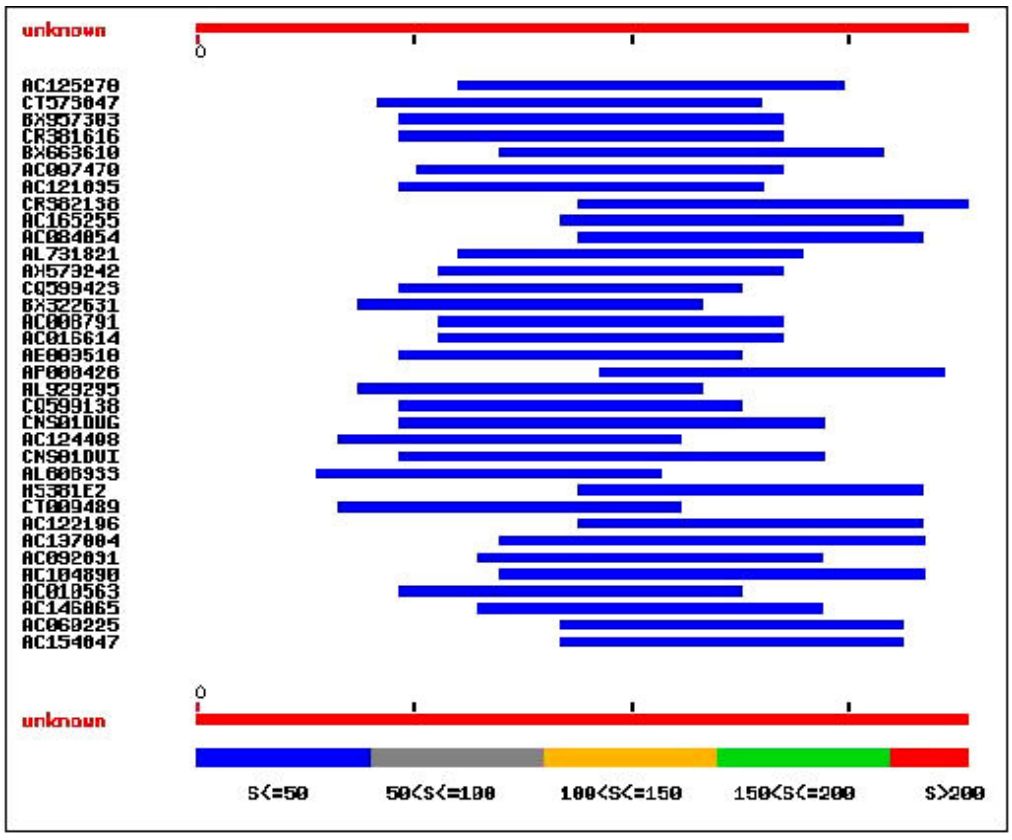
blastall.remote -s sib-gm1.unil.ch -p blastn -d embl-sp wwwtmp/sq.9192.txt -M blosum62 -K 0 -e 10.0 -F T -G def -E def -X def -v 50 -b 50 -g T |

Welcome to the SIB BLAST Network Service (sib-gm1)

Here are your search results:

Query= unknown(43 letters)

Searching.....done



Sequences producing significant alignments:

	Score(bits)	E(value)
emb|CT573047|CT573047 [Mus musculus]Mouse DNA sequence from clon...	40	0.33
emb|AC125278|AC125278 [Mus musculus]Mus musculus BAC clone RP24-...	40	0.33

emb|AC097470|AC097470 [Homo sapiens]Homo sapiens BAC clone RP11-... 38 1.3

emb|CR382138|CR382138 [Debaryomyces hansenii CBS767]Debaryomyces... 38 1.3

emb|Z98748|HS381E2 [Homo sapiens]Human DNA sequence from clone R... 36 5.2

emb|AP000426|AP000426 [Homo sapiens]Homo sapiens genomic DNA, ch... 36 5.2

Λόγω του μεγάλου πλήθους των ακολουθιών που παρήγαγαν σημαντικά αποτελέσματα παραθέτουμε τις λεπτομέρειες και τον τρόπο ευθυγράμμισης μόνο μερικών εξ' αυτών:

- emb|CT573047|CT573047 [Mus musculus]Mouse DNA sequence from clone RP23-73L5 on chromosome 14

Length = 211944

Score = 40.1 bits (20), Expect = 0.33

Identities = 20/20 (100%)

Strand = Plus / Plus

Query: 9 tcagataatacagaaaacac 28

| | | | | | | | | | | | | | | | | | | |

Sbjct: 138673 tcagataatacagaaaacac 138692

- emb|AC125278|AC125278 [Mus musculus]Mus musculus BAC clone RP24-481N2 from 12, complete sequence.

Length = 132398

Score = 40.1 bits (20), Expect = 0.33

Identities = 20/20 (100%)

Strand = Plus / Plus

Query: 13 ataatacagaaaacactgat 32

| | | | | | | | | | | | | | | | | | | |

Sbjct: 58319 ataatacagaaaacactgat 58338

και ούτω καθεξής …

Lambda K H

1.37 0.711 1.31

Gapped

Lambda K H

1.37 0.711 1.31

Matrix: blastn matrix:1 -3

Gap Penalties: Existence: 5, Extension: 2

length of query: 43

length of database: 16,798,337,069

effective HSP length: 19

effective length of query: 24

effective length of database: 16,695,274,305

effective search space: 400686583320

effective search space used: 400686583320

Σε αυτήν την εκτέλεση παρατηρούμε πως 34 νουκλεοτιδικές ακολουθίες παρήγαγαν σημαντικά αποτελέσματα στη έρευνά μας, εκ των οποίων 2 αναλύονται περαιτέρω. Ομοίως για κάθε ακολουθία παρατίθενται πληροφορίες για το είδος ή περιοχή (χρωμόσωμα) στα οποία ανήκει, το μήκος, το αποτέλεσμα-σκορ, η τιμή προσδοκίας E καθώς και το πλήθος των ομοίων συμβόλων. Για παράδειγμα στην στοίχιση με την ακολουθία CT573047 μήκους 211944 συμβόλων η οποία ανήκει στο DNA του ποντικιού, το σκορ είναι 20 (ή 40.1 bits), ενώ η τιμή προσδοκίας Expect είναι 0.33 και οι ταυτοσημίες (Identities) είναι 20(ποσοστό 100%). Συμπεραίνουμε πως η ακολουθία αυτή έχει μικρή πιθανότητα να είναι συσχετισμένη με την υπό έρευνα ακολουθία.

- $4^{ο}$ παράδειγμα

Στο παράδειγμά μας αυτό επιχειρούμε μία πιο εξειδικευμένη, από ιατρικής άποψης, έρευνα. Από την ιστοσελίδα της PIR ([Pir06]) και από την πρωτεϊνική βάση δεδομένων UniRef100 εντοπίσαμε μία πρωτεΐνη, η οποία λειτουργεί ως αντιγόνο ενάντια στον καρκίνο του προστάτη και βρίσκεται σε ένα είδος βατράχου με όνομα Xenopus Tropicalis. Εφόσον λοιπόν αυτή η πρωτεΐνη λειτουργεί θεραπευτικά για τον άνθρωπο, θα θέλαμε να δούμε ποιες νουκλεοτιδικές ακολουθίες στον ανθρώπινο οργανισμό μεταφράζονται σε αυτήν, ή σε ακολουθίες, ομόλογες αυτής. Έτσι θα επιλέξουμε την tblastn εκδοχή του BLAST, η οποία συγκρίνει την ακολουθία-ερώτηση πρωτεΐνης, έναντι βάσης δεδομένων ακολουθιών νουκλεοτιδίων, δυναμικά μεταφραζόμενων σε όλα τα πλαίσια αναγνώσεώς τους.

Η ακολουθία αμινοξέων της πρωτεΐνης αυτής είναι η MDNKRQRARVQGGWAGGARPGSRPA VPPAQRPPAWGKEQPTQEKHFVYSEPAEAVRRVPEPQVIDAAGTHEISFSPSGVSRKLLPSFPPL HRAIWGKDNSQAFETRQGQAITISVRGGAVINILPLGICPYSDCHPHFCLPQWHPLLTMLKDRIE EVTGYGFNSLLCNLYRHDKDSIDWHSDDEPALGTSPIIASLSFGETRNFQMRKKPPPEERGDYT YVERVHVPLDHGTLLLMEGATQQDWQHRVPKEYHDRRPRINLTFRTMYPEP και είναι μη-κους 270 αμινοξέων. Θέτοντάς την ως είσοδο στο πρόγραμμα blastn χωρίς τη χρήση κενών και περιορίζοντας την έρευνά μας στη βάση δεδομένων νουκλεοτιδικών ακολουθιών Unigene Human σε περίπου 19 δευτερόλεπτα είχαμε τα παρακάτω αποτελέσματα. Εξ'αιτίας του πλήθους των επιτυχιών παραθέτουμε μόνο ορισμένα εξ'αυτών.

Program: tblastn
Database: Hs_UG
Number: 24134
Format: plain_text
Sequence:
MDNKRQRARVQGGWAGGARPGSRPAVPPAQRPPAWGKEQPTQEKHFVYSEPAEAVRRVPE
PQVIDAAGTHEISFSPSGVSRKLLPSFPPLHRAIWGKDNSQAFETRQGQAITISVRGGAVINILPL
GICPYSDCHPHFCLPQWHPLLTMLKDRIEEVTGYGFNSLLCNLYRHDKDSIDWHSDDEPALGT
SPIIASLSFGETRNFQMRKKPPPEERGDYTYVERVHVPLDHGTLLLMEGATQQDWQHRVPKEY
HDRRPRINLTFRTMYPEP

Processing, please wait...
blastall.remote -s sib-gm1.unil.ch -p tblastn -d Hs_UG wwwtmp/sq.24134.txt -M blosum62 -K 0 -e 10.0 -F T -G def -E def -X def -v 50 -b 50 -g F |

Here are your search results:
SIB BLAST network server version 1.6 of November 26, 2002
compiled by GNU C version 2.96 20000731 (Red Hat Linux 7.3 2.96-113)
compiled on Jun 10 2003.
Welcome to the SIB BLAST Network Service (sib-gm1)
…
Starting job.
Job was split into 22 pieces: 1 queued, 21 running, 0 done (0.0%)
Job was split into 22 pieces: 1 queued, 20 running, 1 done (4.5%)
TBLASTN 1.5.4-Paracel [2003-06-05]
Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs", Nucleic Acids Res. 25:3389-3402.

Query= Neo (270 letters)
Database: Hs_UG 6,731,038 sequences; 4,024,909,140 total letters

Searching.....done

```
Sequences producing significant alignments:    Score(bits)  E

ug|BC103813.368920|Hs.368920 Homo sapiens alkB...  276    8e-88
g|CD364334.368920|Hs.368920 UI-H-FT2-bjf-f-03...   276    3e-72
ug|BM824362.368920|Hs.368920 K-EST009 Homo sa...   276    3e-72
ug|BF062547.368920|Hs.368920 7h61c05.x1 NCI_CG...  202    4e-50
ug|AI636076.368920|Hs.368920 tz92e06.x1 Homo ...   179    5e-43
ug|BE293567.368920|Hs.368920 601186716F1 NIH_M...  165    4e-51
```

- ug|BC103813.368920|Hs.368920 Homo sapiens alkB, alkylation repair homolog 3 (E. coli), mRNA (cDNA clone MGC:118792 IMAGE:40000375), complete cds

```
Length = 1158
Score =  276 bits (594), Expect(3) = 8e-88
Identities = 103/130 (79%), Positives = 116/130 (89%)
Frame = +2

Query: 141 PQWHPLLTMLKDRIEEVTGYGFNSLLCNLYRHDKDSIDWHSDDEPALGTSPIIASLSFGE 200
           P WHP+L  LK+RIEE TG+ FNSLLCNLYR++KDS+DWHSDDEP+LG  PIIASLSFG
Sbjct: 635 PHWHPVLRTLKNRIEENTGHTFNSLLCNLYRNEKDSVDWHSDDEPSLGRCPIIASLSFGA 814

Query: 201 TRNFQMRKKPPPEERGDYTYVERVHVPLDHGTLLLMEGATQQDWQHRVPKEYHDRRPRIN 260
           TR F+MRKKPPPEE GDYTYVERV +PLDHGTLL+MEGATQ DWQHRVPKEYH R PR+N
Sbjct: 815 TRTFEMRKKPPPEENGDYTYVERVKIPLDHGTLLIMEGATQADWQHRVPKEYHSREPRVN 994

Query: 261 LTFRTMYPEP 270
```

```
           LTFRT+YP+P
Sbjct: 995 LTFRTVYPDP 1024

Score = 65.4 bits (136), Expect(3) = 8e-88
Identities = 26/48 (54%), Positives = 33/48 (68%)
Frame = +2

Query: 37  KEQPTQEKHFVYSEPAEAVRRVPEPQVIDAAGTHEISFSPSGVSRKLL 84
           KE    ++ FV+ EP + VRR PEP+VID  G +EIS SP+GVSR  L
Sbjct: 314 KEHHLSDREFVFKEPQQVVRRAPEPRVIDREGVYEISLSPTGVSRVCL 457

Score = 31.4 bits (62), Expect(3) = 1e-87
Identities = 11/15 (73%), Positives = 13/15 (86%)
Frame = +3

Query: 1   MDNKRQRARVQGGWA 15
           M+ KR+RARVQG WA
Sbjct: 182 MEEKRRRARVQGAWA 226


Database: Hs_UG
Posted date:  Jun 18, 2007 11:45 AM
Number of letters in database: 4,024,909,140
Number of sequences in database:  6,731,038
  Lambda     K      H
  0.319    0.138    0.443
Matrix: BLOSUM62
length of query: 270
length of database: 1,341,636,380
effective HSP length: 54
effective length of query: 216
effective length of database: 978,160,328
effective search space: 211282630848
effective search space used: 211282630848
```

Σε αυτό το παράδειγμα λοιπόν σημειώνεται για κάθε ακολουθία της βάσης δεδομένων οι σημαντικές τοπικές ευθυγραμμίσεις που εντοπίστηκαν, το σκορ S και η τιμή E. Επίσης, εφόσον το tblastn μεταφράζει τις ακολουθίες της βάσης δεδομένων, σημειώνεται και το εκάστοτε πλαίσιο μετάφρασης, με το πρόσημο να δείχνει τη φορά μετάφρασης και με τον αριθμο (από 1 έως 3) να δείχνει από ποιο νουκλεοτίδιο ξεκινά η μετάφραση. Εάν σε κάποια ακολουθία υπάρχουν περισσότερες της μίας αξιοσημείωτες ευθυγραμμίσεις, τότε αναφέρονται όλες, αλλά το αποτέλεσμα-σκορ (που συγκρίνεται με αποτελέσματα άλλων ακολουθιών) ισούται με το μέγιστο των αποτελεσμάτων των ευθυγραμμίσεων.

- $5^{ο}$ παράδειγμα

Το $5^{ο}$ παράδειγμα αφορά την PSI έκδοση του BLAST και σύγκριση τυχαίας πρωτεϊνικής ακολουθίας μετά από 10 δευτερόλεπτα με χρήση του πίνακα Blosum62. Οι παράμετροι της έρευνας ήταν οι παρακάτω:

Βάση δεδομένων: (Database of proteins) Swiss-Prot, TrEMBL, Swiss-Prot splice variants, trEST, trGEN, trome, Current ENSEMBL peptides for all species, Microbial complete proteomes, RefSeq Release, RefSeq weekly updates.

Taxonomic range none.

Method BLOSUM62 - 11/1

E-value inclusion 1e-3

E-value report 1

Cluster matches to approx. level of identity 70 %

Στη γραφική απεικόνιση του αποτελέσματος της εκτέλεσης του προγράμματος, με κόκκινο χρώμα απεικονίζονται οι ακολουθίες που σε σύγκριση με την υπό έρευνα ακολουθία, παρήγαγαν μη σημαντικά αποτελέσματα, ενώ με πράσινο απεικονίζονται όσες παρήγαγαν σημαντικά. Παρατηρούμε λοιπόν πως αν και το πλήθος των ακολουθιών που παρήγαγαν σημαντικά αποτελέσματα ήταν μεγάλο, μόνο ορισμένες εξ' αυτών απεικονίζονται γραφικά, ενώ οι υπόλοιπες είναι αρκετά όμοιες με τις απεικονιζόμενες. Μάλιστα ο βαθμός ομοιότητάς τους καταγράφεται στα αποτελέσματα.

Στη συνέχεια παρατίθενται λεπτομέρειες της έρευνας μόνο για ορισμένες από τις ακολουθίες που βρέθηκαν (λόγω του μεγάλου πλήθους αυτών).

te:STR_7051_1/265-327 Bit-score: **109**, E-value: **9e-25**,.

PROTSIM=ref:NP_199373.1; Build: 29 [entry , synonyms , graphics]

```
Q:     1 KLELGHRAQVRKKPTVEGFTHDWMVFVRGPE-----------VFHLHESFPRPKRVCKDPPYK
         KLELGHRAQVRKKPT+EGFTHDWMVFVRGPE           VFHLHESFPRPKRVCKDPPYK
S:   265 KLELGHRAQVRKKPTLEGFTHDWMVFVRGPEHSNIQHFVEKVVFHLHESFPRPKRVCKDPPYK
```

ens:ENSCAFP00000002401/7-69 Bit-score: **107**, E-value: **3e-24,**, The matched region is approx. **98 %** identical to the one of te:STR_7051_1/265-327. *>ENSCAFP00000002401 pep:novel chromosome:BROADD1:11:41277048:41551589:-1 gene:ENSCAFG00000001635 transcript:ENSCAFT00000002585* [entry , synonyms , graphics]

tr:Q9ERL0_MOUSE/10-72 Bit-score: **94.3**, E-value: **3e-20,**, The matched region is approx. **84 %** identical to the one of te:STR_7051_1/265-327. *Btk-PH-domain binding protein (Activated spleen cDNA, RIKEN full-length enriched library, clone:F830102A04 product:myeloid/lymphoid ormixed lineage-leukemia translocation to 1 homolog (Drosophila), fullinsert sequence) (Lung RCB-0558 LLC cDNA, RIKEN full-length enrichedlibrary, clone:G730008H16 product:myeloid/lymphoid or mixed lineage-leukemia translocation to 1 homolog (Drosophila), full insertsequence) (Myeloid/lymphoid or mixed lineage-leukemia translocation to1 homolog) (Drosophila)* [entry , synonyms , graphics]

tr:Q4TET8_TETNG/9-67 Bit-score: **65.4**, E-value: **2e-11,**. *Chromosome undetermined SCAF5131, whole genome shotgun sequence.(Fragment)* [entry , synonyms , graphics]

```
Q:     5 KLELGHRAQVRKKPTVEGFTHDWMVFVRGPE-----------VFHLHESFPRPKRVCKDPPYK
           G ++    ++  V  FTHDWMVFVRGPE           VF LHESFP+PKRVCK+PPYK
S:     9     GPQSSAEEEGDVRSFTHDWMVFVRGPETGDIQHFVDKVVFRLHESFPKPKRVCKEPPYK
```

ens:GSTENP00002085001/9-67 Bit-score: **65.4**, E-value: **2e-11,**, The matched region is approx. **100 %** identical to the one of tr:Q4TET8_TETNG/9-67. *>GSTENP00002085001 pep:known chromosome:TETRAODON7:Un_random:75178682:75180508:1 gene:GSTENG00002085001 transcript:GSTENT00002085001* [entry , synonyms , graphics]

rs:XP_655411/19-71 Bit-score: **40.0**, E-value: **7e-04,**. *conserved hypothetical protein [Entamoeba histolytica HM-1:IMSS]* [entry , synonyms , graphics]

```
Q:    13 KLELGHRAQVRKKPTVEGFTHDWMVFVR-------------GPEVFHLHESFPRPKRVCKDPPYK
                   K TV   TH+W +F+R                  FHLHESF  P R    PPY+
S:    19           KKTVSNNTHNWTLFIRPFNEEDIELFNVIDSVTFHLHESFQNPHRRVSQPPYE
```

tr:Q51B02_ENTHI/19-71 Bit-score: **40.0**, E-value: **7e-04,**, The matched region is approx. **100 %** identical to the one of rs:XP_655411/19-71. *Hypothetical protein* [entry , synonyms , graphics]

Παρατηρούμε λοιπόν πως και σε αυτήν την εκδοχή του BLAST απεικονίζονται γραφικώς οι στοιχίσεις των ακολουθιών της βάσης δεδομένων με την υπό έρευνα ακολουθία, και στη συνέχεια για κάθε μία παρατίθενται πληροφορίες όπως το σκορ και η τιμή προσδοκίας (που για παράδειγμα στην rs:XP_655411/19-71 είναι 40 bits, και 7e-04), αλλά και πληροφορίες όπως η ομοιότητα με άλλες, καθώς και δυνατότητες γραφικής αναπαράστασης (graphics).

3.4.2 *FASTA*

Ένα άλλο πρόγραμμα παρόμοιας στρατηγικής με το BLAST, είναι το FASTA. Το FASTA βασίζεται σε μία μέθοδο που αναπτύχθηκε από τους Pearson και Lipman το 1985 και λειτουργεί πάνω σε διαφορετικό σύνολο υποθέσεων απ'ότι το BLAST και συνεπώς παράγει διαφορετικά αποτελέσματα. Η λέξη FASTA προέρχεται από το FAST-All, που σημαίνει πως

μπορεί να χρησιμοποιηθεί για γρήγορη πρωτεϊνική ή νουκλεοτιδική σύγκριση. Το πρόγραμμα αυτό επιτυγχάνει υψηλό επίπεδο σε αναζήτηση ομοιοτήτων και με υψηλή ταχύτητα. Αυτό γίνεται πραγματοποιώντας βέλτιστες αναζητήσεις για τοπικές ευθυγραμμίσεις χρησιμοποιώντας έναν πίνακα υποκατάστασης στοιχείων. Η υψηλή ταχύτητα του προγράμματος οφείλεται στη χρήση του παρατηρούμενου μοτίβου χτυπημάτων λέξεων (pattern of word hits) για την ανίχνευση πιθανών ταιριασμάτων προτού ακόμα επιχειρηθεί η χρονοβόρα βέλτιστη αναζήτηση. Το πάρε-δώσε (trade off) μεταξύ ταχύτητας και ευαισθησίας ελέγχεται από την παράμετρο ktup που καθορίζει το μήκος λέξης που θέλουμε. Αντί να ερευνώνται όλα τα χτυπήματα, αναζητούνται μόνο τα τμήματα εκείνα που περιέχουν γειτονικά χτυπήματα-επιτυχίες.

Το πρόγραμμα αυτό είναι πιο ευαίσθητο από το BLAST κάτι που αντανακλάται και στο χρόνο που απαιτείται για την παραγωγή αποτελεσμάτων, ο οποίος είναι μεγαλύτερος. Το FASTA παράγει βέλτιστα αποτελέσματα (scores) τοπικών ευθυγραμμίσεων για τη σύγκριση της υπό έρευνα ακολουθίας με κάθε ακολουθία της βάσης δεδομένων μας. Η πλειοψηφία αυτών των αποτελεσμάτων περιέχει ασυσχέτιστες ακολουθίες και μπορεί να χρησιμοποιηθεί για την εκτίμηση των παραμέτρων λ και K.

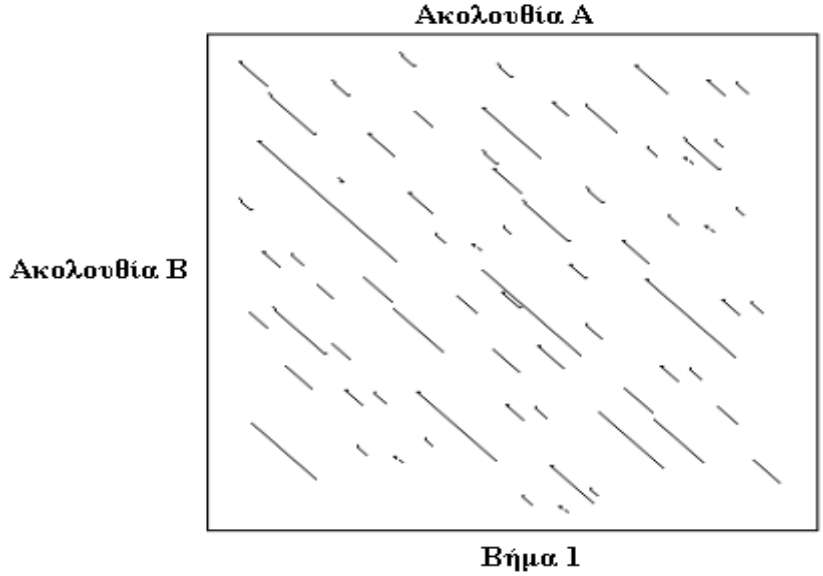
Το FASTA χρησιμοποιεί τέσσερα βήματα για να υπολογίσει τρία αποτελέσματα που χαρακτηρίζουν την ομοιότητα δύο ακολουθιών:

- Βήμα 1

Ταυτοποιεί περιοχές κοινές στις δύο ακολουθίες με τη μέγιστη πυκνότητα μονών ταυτοσημιών (ktup=1) ή ζευγών ταυτοσημιών (ktup=2) ή περισσοτέρων συνεχόμενων ταυτοσημιών, κάτι που επιτυγχάνεται με τη χρήση μιας γρήγορης τεχνικής αναζήτησης ταυτοσημιών ανάμεσα σε δύο ακολουθίες. Το FASTA επιτυγχάνει μεγάλο μέρος της ταχύτητας και επιλεκτικότητας αυτού του βήματος χρησιμοποιώντας έναν πίνακα αναζήτησης (lookup table) για να εντοπίσει όλες τις ταυτοσημίες ή τις ομάδες ταυτοσημιών ανάμεσα σε δύο ακολουθίες νουκλεοτιδίων ή αμινοξέων. Η παράμετρος ktup καθορίζει πόσες διαδοχικές ταυτοσημίες απαιτούνται σε ένα ταίριασμα. Η τιμή ktup=2 συχνά χρησιμοποιείται για τη σύγκριση ακολουθιών πρωτεϊνών, και σημαίνει πως το πρόγραμμα εξετάζει μόνο τα τμήματα εκείνα των συγκρινόμενων ακολουθιών που έχουν τουλάχιστον δύο γειτονικά ταυτόσημα στοιχεία. Περισσότερο ευαίσθητες έρευνες μπορούν να γίνουν όταν ktup=1. Όσον αφορά τη σύγκριση ακολουθιών DNA η παράμετρος ktup κυμαίνεται μεταξύ 1 και 6, ενώ οι συνιστώμενες τιμές είναι μεταξύ 4 και 6.

Σε συνδυασμό με τον πίνακα αναζήτησης χρησιμοποιείται η μέθοδος της διαγωνίου για την εύρεση όλων των περιοχών ομοιότητας ανάμεσα σε δύο ακολουθίες, μετρώντας ταυτόχρονα τα ktup ταιριάσματα και συνυπολογίζοντας τις ποινές για τα παρεμβαλλόμενα ανόμοια σύμβολα (όπως αναλύσαμε παραπάνω για τους πίνακες αληθείας-boolean matrices). Έτσι η

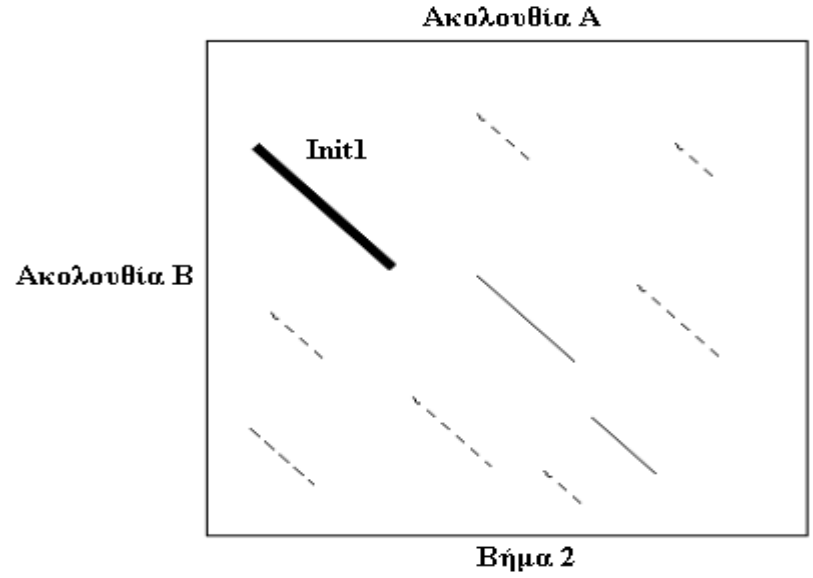
μέθοδος αυτή αναγνωρίζει περιοχές μιας διαγωνίου με τη μέγιστη πυκνότητα ktup ταιριασμάτων και στη συνέχεια το FASTA αποθηκεύει τις δέκα καλύτερες τοπικές περιοχές ανεξαρτήτως αν βρίσκονται στην ίδια ή σε διαφορετικές διαγωνίους (Σχήμα 3.9).



Σχήμα 3.9: FASTA Βήμα 1

- Βήμα 2

Σαρώνονται ξανά οι δέκα περιοχές με την υψηλότερη πυκνότητα ταιριασμάτων χρησιμοποιώντας τον πίνακα PAM250 και στη συνέχεια ψαλιδίζονται οι άκρες κάθε περιοχής προκειμένου να επιλεχθούν μόνον εκείνα τα τμήματα που συμβάλλουν περισσότερο στο μέγιστο αποτέλεσμα (highest score). Το FASTA μπορεί να χρησιμοποιηθεί και για τη σύγκριση ακολουθιών DNA οπότε και κατασκευάζονται ειδικοί πίνακες. Για κάθε μία από τις καλύτερες περιοχές των διαγωνίων που σαρώθηκαν ξανά, έχει ανιχνευθεί μία υποπεριοχή με το μέγιστο αποτέλεσμα το οποίο και αναφέρεται ως αποτέλεσμα init1 (Σχήμα 3.10).

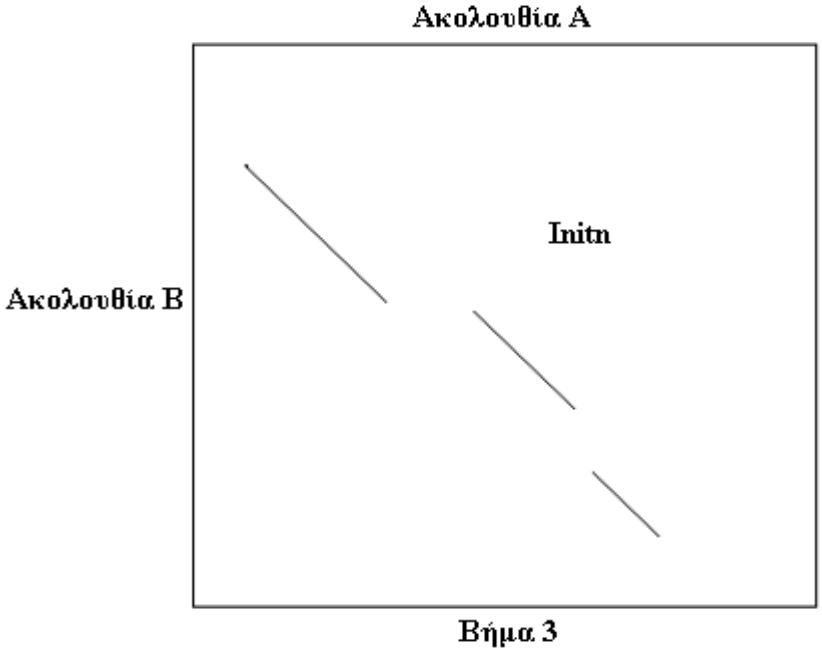


Σχήμα 3.10: FASTA Βήμα 2

- Βήμα 3

Εάν υπάρχουν αρκετές αρχικές περιοχές με αποτελέσματα μεγαλύτερα από κάποια τιμή κατωφλίου γίνεται έλεγχος για το αν οι ψαλιδισμένες αρχικές περιοχές μπορούν να ενωθούν για να σχηματιστεί μια προσεγγιστική ευθυγράμμιση με κενά. Υπολογίζεται ένα αποτέλεσμα ομοιότητας που ισούται με το άθροισμα των ενωμένων αρχικών περιοχών μείον κάποια ποινή για κάθε κενό (συνήθως 20). Αυτό το αρχικό σκορ ομοιότητας (initn) χρησιμοποιείται για την κατάταξη των ακολουθιών της βιβλιοθήκης.

Αναλυτικότερα, το FASTA κατά τη διάρκεια αναζήτησης σε βιβλιοθήκη ακολουθιών ελέγχει αν μερικές αρχικές περιοχές μπορούν να ενωθούν σε μία ευθυγράμμιση, προκειμένου να βελτιωθεί το αρχικό αποτέλεσμα. Το FASTA υπολογίζει μία βέλτιστη ευθυγράμμιση αρχικών περιοχών ως ένα συνδυασμό συμβατών περιοχών με μέγιστο αποτέλεσμα. Αυτή η βέλτιστη στοίχιση μπορεί να υπολογιστεί γρήγορα χρησιμοποιώντας έναν αλγόριθμο δυναμικού προγραμματισμού. Το FASTA χρησιμοποιεί το αποτέλεσμα που προκύπτει και αναφέρεται ως initn, για να κατατάξει τη βιβλιοθήκη ακολουθιών. Το τρίτο αυτό "ενωτικό" βήμα στον υπολογισμό του αρχικού αποτελέσματος αυξάνει την ευαισθησία της μεθόδου, καθώς επιτρέπονται εισαγωγές, διαγραφές, αλλά και συντηρητικές αντικαταστάσεις συμβόλων. Οι τροποποιήσεις αυτές παρόλα αυτά μειώνουν την επιλεκτικότητα. Ο περιορισμός της μείωσης της επιλεκτικότητας επιτυγχάνεται εισάγοντας στο βήμα βελτιστοποίησης μόνο όσες αρχικές περιοχές έχουν αποτέλεσμα πάνω από ένα εμπειρικά υπολογισμένο κατώφλι. (Για παράδειγμα για μία υπό έρευνα ακολουθία μήκους 200 συμβόλων και με τιμή ktup ίση με 2, η τιμή του κατωφλίου αυτού είναι 28).

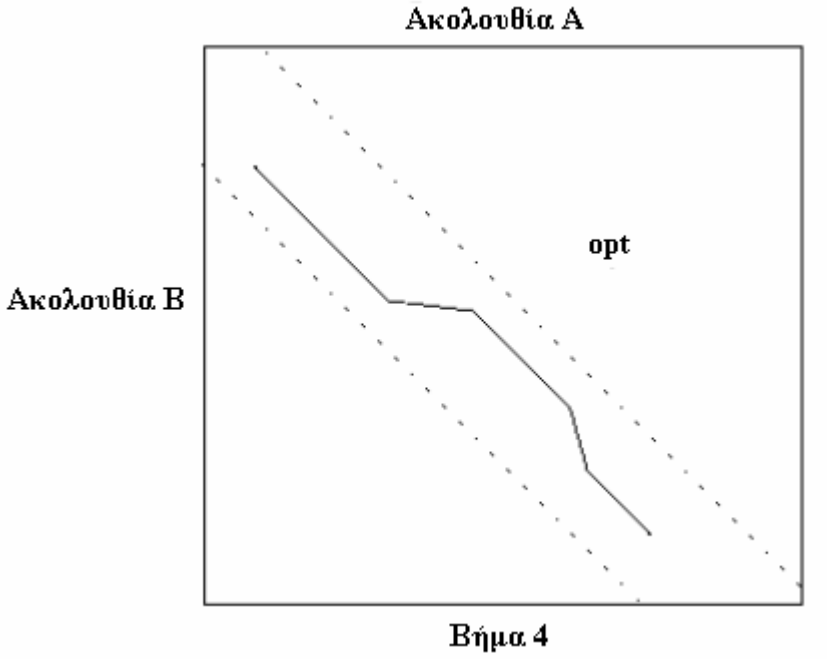


Σχήμα 3.11: FASTA Βήμα 3

- Βήμα 4

Κατασκευάζεται η NWS βέλτιστη στοίχιση (χρησιμοποιώντας δηλαδή τον αλγόριθμο των Needleman-Wunch-Sellers) γύρω από το κέντρο της init1 περιοχής, με τη δυνατότητα εισαγωγής κενών αλλά και τη δυνατότητα να επεκταθεί η ευθυγράμμιση κατά μήκος της ίδιας διαγωνίου, κατά ένα στενό εύρος συμβόλων. Το FASTA αναφέρει αυτό το αποτέλεσμα ως το βελτιστοποιημένο (opt).

Μετά από μια πλήρη έρευνα της βιβλιοθήκης το FASTA σχεδιάζει τα αρχικά αποτελέσματα κάθε ακολουθίας σε ένα ιστόγραμμα, υπολογίζει το μέσο αποτέλεσμα ομοιότητας για τη σύγκριση της υπό έρευνα ακολουθίας με κάθε ακολουθία της βιβλιοθήκης και καθορίζει την τυπική απόκλιση της κατανομής των αρχικών αποτελεσμάτων. Τα αρχικά αποτελέσματα χρησιμοποιούνται στην κατάταξη της βιβλιοθήκης ακολουθιών και στο τέταρτο και τελευταίο βήμα της σύγκρισης οι υψηλότερου αποτελέσματος ακολουθίες της βιβλιοθήκης στοιχίζονται με βάση μια τροποποιημένη εκδοχή της τυπικής μεθόδου βελτιστοποίησης NWS. Η μέθοδος αυτή εφαρμόζει τον ίδιο πίνακα αποτελεσμάτων που χρησιμοποιείται και στις αρχικές περιοχές, ενώ οι παραγόμενες βέλτιστες στοιχίσεις μελετώνται περαιτέρω για πιθανές συσχετίσεις και το βελτιστοποιημένο αποτέλεσμα ομοιότητας αναφέρεται.



Σχήμα 3.12: FASTA Βήμα 4

Αφού γίνουν τα παραπάνω βήματα για όλες τις ακολουθίες, αναφέρονται οι καλύτερου αποτελέσματος ευθυγραμμίσεις.

Όσον αφορά τα τρία αποτελέσματα που το πρόγραμμα υπολογίζει, υπάρχουν οι παρακάτω περιπτώσεις:

- Initn = init1 = opt. Δείχνει 100% ομολογία των ακολουθιών.

- Initn > init. Τότε υπάρχουν στην ακολουθία της βάσης δεδομένων περισσότερες της μίας περιοχές ταιριάσματος με την υπό έρευνα, οι οποίες διαχωρίζονται από φτωχά ταιριασμένες περιοχές.
- Opt > initn. Τότε οι περιοχές ταιριάσματος βελτιώνονται σημαντικά με την εισαγωγή κενών και πιθανότατα αναφερόμαστε σε μη ομόλογες ακολουθίες.
- Opt < initn. Το στενό εύρος συμβόλων γύρω από το οποίο μπορεί να γίνει η βέλτιστη ευθυγράμμιση αποκλείει κάποιες περιοχές ταιριάσματος.

Το FASTA (στην FASTA3 έκδοση που ερευνήσαμε) όπως και το BLAST έχει διάφορες εκδοχές οι σημαντικότερες των οποίων απεικονίζονται στον παρακάτω πίνακα:

fasta3	Συγκρίνει μία ακολουθία αμινοξέων έναντι πρωτεϊνικής βάσης δεδομένων ή μια ακολουθία νουκλεοτιδίων έναντι βάσης δεδομένων ακολουθιών DNA, χρησιμοποιώντας τον αλγόριθμο του FASTA των Pearson και Lipman. Η ταχύτητα έρευνας αλλά και η ευαισθησία ελέγχονται με την ktup (μήκος λέξης) παράμετρο. Για ακολουθίες αμινοξέων η προκαθορισμένη (default) τιμή ktup ισούται με 2, ενώ με ktup=1 έχουμε πιο ευαίσθητη αλλά και βραδύτερη έρευνα. Για ακολουθίες DNA η default τιμή είναι 6 ενώ οι τιμές ktup=3 και ktup=4 παρέχουν μεγαλύτερη ευαισθησία. Η ktup=1 χρησιμοποιείται για μικρού μήκους ακολουθίες (ολιγονουκλεοτίδια-μήκους μικρότερου των 20 συμβόλων).
ssearch3	Συγκρίνει μία ακολουθία αμινοξέων έναντι πρωτεΐνικής βάσης δεδομένων ή μια ακολουθία DNA έναντι βάσης δεδομένων ακολουθιών DNA χρησιμοποιώντας τον αλγόριθμο Smith-Waterman. Το ssearch3 είναι περίπου 10 φορές βραδύτερο από το fasta3, αλλά και πιο ευαίσθητο.
fastx3/ fasty3	Συγκρίνει μία ακολουθία DNA μεταφραζόμενης στα έξι πλαίσια ανάγνωσής της (βλέπε Σχήμα 4.8) έναντι πρωτεϊνικής βάσης δεδομένων, επιτρέποντας τη χρήση κενών. Το fastx3 χρησιμοποιεί έναν απλούστερο και ταχύτερο αλγόριθμο, ενώ το fasty3 είναι βραδύτερο αλλά παράγει καλύτερες ευθυγραμμίσεις.
tfastx3/ tfasty3	Συγκρίνει μία ακολουθία αμινοξέων με βάση δεδομένων ακολουθιών DNA οι οποίες μεταφράζονται και στα έξι πλαίσια ανάγνωσής τους (βλέπε Σχήμα 3.7).

Το FASTA, όπως και το BLAST, αναφέρει το σκορ της κάθε ευθυγράμμισης σε bits σύμφωνα με τον τύπο $S(bits) = \frac{\lambda \times S - \ln K}{\ln 2}$, όπου το S είναι το μη επεξεργασμένο σκορ (raw-που εδώ ισούται με το opt), και τα λ και K είναι οι γνωστές στατιστικές παράμετροι. Αφού λοιπόν υπολογιστεί το κανονικοποιημένο S(bits), γίνεται δυνατή όχι μόνο η σύγκριση διαφορετικών ερευνών του FASTA, αλλά και η σύγκριση αποτελεσμάτων μεταξύ FASTA και BLAST. Η

τιμή της παραμέτρου Ε είναι και σε αυτό το πρόγραμμα το σημαντικότερο κριτήριο της ομοιότητας δύο ακολουθιών. Συγκεκριμένα για τιμές του Ε μικρότερες του 0.01 οι ακολουθίες είναι σχεδόν πάντα ομόλογες, ενώ για τιμές από 1 έως 10 οι ακολουθίες είναι σε κάποιο βαθμό συσχετισμένες. Βεβαίως και σε αυτό το πρόγραμμα υπάρχει η πιθανότητα να υπερεκτιμάται η σημασία των αποτελεσμάτων ορισμένων ασυσχέτιστων ακολουθιών, κάτι που αντιμετωπίζεται ενεργοποιώντας την επιλογή βελτιστοποίησης "–ο".

3.4.2.1 *Εκτέλεση του FASTA*

Ένα παράδειγμα εκτέλεσης του προγράμματος FASTA φαίνεται παρακάτω.(Η εκτέλεση έγινε διαδραστικά μέσω του διαδικτυακού τόπου του EBI [Ebi07] και σε 24 δευτερόλεπτα λάβαμε την απάντησή μας.)

Η τυχαία νουκλεοτιδική ακολουθία που υποβάλλαμε προς έρευνα είναι η ATGTGCTGTAGC TGTAGTGCTGTAGTCGTA, για σύγκριση με την βάση νουκλεοτιδικών ακολουθιών Emfun (Fungi) όπου και είχαμε τα εξής αποτελέσματα:

SUBMISSION PARAMETERS			
Title	ATGTGCTGTAGCTGTA GTGCTGTAGTCGTA	Database	Emfun
Sequence length	30	Sequence type	N
Program	fasta	Version	3.4t25 Sept 2, 2005
Expectation upper value	10.0	Sequence range	1-
Number of scores	50	Number of alignments	50
Word size	6	Open gap penalty	-14
Gap extension penalty	-4	Histogram	False

Εμφανίζουμε το αποτέλεσμα επιλέγοντας Fasta Result και έχουμε τα παρακάτω

```
FASTA searches a protein or DNA sequence data bank
 version 3.4t25 Sept 2, 2005
Please cite:
 W.R. Pearson & D.J. Lipman PNAS (1988) 85:2444-2448
```

```
Query library @ vs +emfun library
searching /ebi/services/idata/v1679/fastadb/em_fun library

  1>>>ATGTGCTGTAGCTGTAGTGCTGTAGTCGTA - 30 nt
 vs  EMBL Fungi library

331241143 residues in 159875 sequences
 statistics sampled from 60000 to 161515 sequences
  Expectation_n fit: rho(ln(x))= 4.6585+/-0.000174; mu= 4.8810+/- 0.012
 mean_var=40.5785+/- 9.306, 0's: 4435 Z-trim: 4435  B-trim: 1271 in 2/73
 Lambda= 0.201338

FASTA (3.47 Mar 2004) function [optimized, +5/-4 matrix (5:-4)] ktup: 6
 join: 45, opt: 30, open/ext: -14/-4, width:  16
 Scan time: 23.480
```

The best scores are:	**opt**	**bits**	**E**
EM_FUN:CR380954; CR380954 Candida glabrata s...	92	32.1	2.2
EM_FUN:AP007166; AP007166 Aspergillus oryzae ...	91	31.8	2.7
EM_FUN:AE017345; AE017345 Cryptococcus neofor...	91	31.8	2.7
EM_FUN:CR382127; CR382127 Yarrowia lipolytica...	89	31.2	4.1
EM_FUN:AP007150; AP007150 Aspergillus oryzae ...	89	31.2	4.1
EM_FUN:BX294012; BX294012 Neurospora crassa D...	89	31.2	4.1
EM_FUN:AJ575660; AJ575660 Yarrowia lipolytica...	89	31.2	4.1
EM_FUN:AE017342; AE017342 Cryptococcus neofor...	86	30.3	7.4
EM_FUN:AF041976; AF041976 Emericella nidulans...	87	30.3	7.6
EM_FUN:AE017343; AE017343 Cryptococcus neofor...	85	30.0	9.1
EM_FUN:CR382129; CR382129 Yarrowia lipolytica...	85	30.0	9.1
EM_FUN:CR382130; CR382130 Yarrowia lipolytica...	85	30.0	9.1

```
>>EM_FUN:CR380954; CR380954 Candida glabrata strain CBS1  (1050361
nt)rev-comp initn: 78 init1: 75 opt: 92 Z-score: 104.2 bits: 32.1
E():  2.2
banded Smith-Waterman score: 92;  81.250% identity (81.250% similar)
in 32 nt overlap (30-3:603853-603884)

                                    30        20            10
ATGTG-                               TACGACTACAGC----ACTACAGCTACAGC
                                     :::: :::::::     ::::::::::::
EM_FUN GACACCCTTTACAATGAAGAATGTGAAGGTTACGGCTACAGCTACGGCTACAGCTACAGC
         603830    603840    603850    603860    603870    603880


ATGTG- ACAT
       ::
EM_FUN ACTCTGAAGTCCCCCTAGGTGATACTGACAGTTGTGCTCATCGCCTTACTTCACGGGTCA
         603890    603900    603910    603920    603930    603940

>>EM_FUN:AP007150; AP007150 Aspergillus oryzae RIB40 gen  (1913118
nt)
 initn:  89 init1:  89 opt:  89  Z-score: 99.5  bits: 31.2 E():  4.1
banded Smith-Waterman score: 89;  84.000% identity (84.000% similar)
in 25 nt overlap (2-26:154528-154552)

                                          10        20        30
ATGTGC                                ATGTGCTGTAGCTGTAGTGCTGTAGTCGTA
                                      ::::::::   ::::: ::::::::
EM_FUN GATCTTCCTCGACTCTGCATAGAAGTCAAGTGTGCTGTCCTTGTAGGGCTGTAGTGCTGA
    154500    154510    154520    154530    154540    154550

EM_FUN TTAGTAAGCCAGGGTTGGTTCATCCCTTTCCGATTACTTGACTACTACCAACAATTCCTT
    154560    154570    154580    154590    154600    154610
```

```
>>EM_FUN:CR382130; CR382130 Yarrowia lipolytica chromoso  (3633272
nt)
 initn:  61 init1:  61 opt:  85  Z-score: 93.2  bits: 30.0 E():  9.1
banded Smith-Waterman score: 85;  80.769% identity (80.769% similar)
in 26 nt overlap (4-29:3317847-3317872)

                                          10        20        30
ATGTGC                                ATGTGCTGTAGCTGTAGTGCTGTAGTCGTA
                                        ::  :::: ::::::: :::::: ::
EM_FUN TCGAAGGATTACGTGTACTGTCGTTGCTGTTGTAGTAGTTGTAGTGTTGTAGTTGTTGTT
    3317820   3317830   3317840   3317850   3317860   3317870

EM_FUN CAACGTCACCTAACTAAATGAGAAAGGGTAGCAACTACATCTTCAACTAGACTCACCCAA
    3317880   3317890   3317900   3317910   3317920   3317930


30 residues in 1 query   sequences
331241143 residues in 159875 library sequences
 Tcomplib [34t25] (4 proc)
 start: Sun Dec  3 23:41:29 2006 done: Sun Dec  3 23:41:42 2006
 Total Scan time: 23.480 Total Display time:  0.090

Function used was FASTA [version 3.4t25 Sept 2, 2005]
```

Alignment	DB:ID	Source	Length	Identity%	Similar%	Overlap	E()
1	EM_FUN:CR380954	Candida glabrata strain CBS1	1050361	81.250	81.250	32	2.2
2	EM_FUN:AP007166	Aspergillus oryzae RIB40 gen	1841713	95.000	95.000	20	2.7
3	EM_FUN:AE017345	Cryptococcus neoformans var.	1507550	95.000	95.000	20	2.7
4	EM_FUN:CR382127	Yarrowia lipolytica chromoso	2303261	84.000	84.000	25	4.1
5	EM_FUN:AP007150	Aspergillus oryzae RIB40 gen	1913118	84.000	84.000	25	4.1
6	EM_FUN:BX294012	Neurospora crassa DNA linkag	161126	88.462	88.462	26	4.1

7	EM_FUN:AJ575660	Yarrowia lipolytica iah1, mt	78189	84.000	84.000	25	4.1
8	EM_FUN:AE017342	Cryptococcus neoformans var.	1632307	82.759	82.759	29	7.4
9	EM_FUN:AF041976	Emericella nidulans nitrogen	2173	90.476	90.476	21	7.6
10	EM_FUN:AE017343	Cryptococcus neoformans var.	2105742	80.769	80.769	26	9.1
11	EM_FUN:CR382129	Yarrowia lipolytica chromoso	3272609	100.000	100.000	17	9.1
12	EM_FUN:CR382130	Yarrowia lipolytica chromoso	3633272	80.769	80.769	26	9.1

Για κάθε ακολουθία παρατίθενται τα αποτελέσματα initn, init1, opt, καθώς και το σκορ και η τιμή προσδοκίας. Για παράδειγμα η ακολουθία CR382130 (Yarrowia lipolytica chromoso) είχε αρχικό αποτέλεσμα init1 ίσο με 61, initn ίσο με 61, βελτιστοποιημένο αποτέλεσμα opt ίσο με 85, σκορ ίσο με 93.2 (ή 30.0 bits) και E ίσο με 9.1. Από την τιμή του E συμπεραίνουμε πως η ακολουθία αυτή είναι σε κάποιο μικρό βαθμό συσχετισμένη με την ακολουθία που ερευνούμε και πιθανότατα δεν είναι ομόλογες ακολουθίες.

Ο διαδικτυακός τόπος αυτός μας παρέχει πολλές επιλογές όπως

- η εμφάνιση των σχολίων μέσω εντολής Show Annotations

Παρατίθενται μερικά σχόλια για μία εκ των ακολουθιών

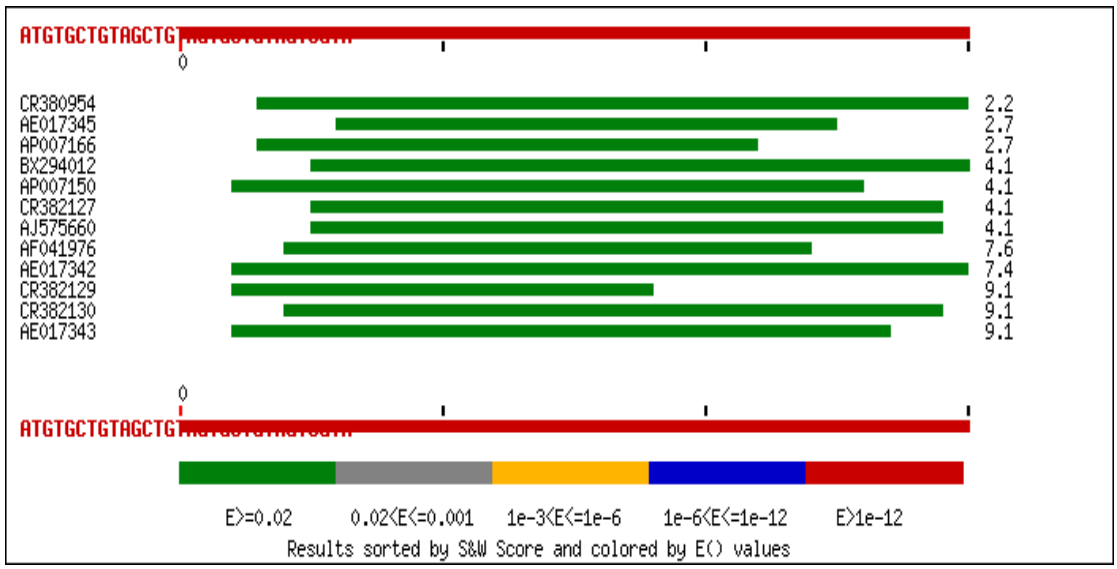
Alignment	DB:ID	Source	Length	Identity%	Similar%	Overlap	E()
1	EM_FUN:CR380954	Candida glabrata strain	1050361	81.250	81.250	32	2.2

		CBS1					

```
ID   CR380954; SV 1; linear; genomic DNA; STD; FUN;
1050361 BP.
AC   CR380954;
DT   30-JUN-2004 (Rel. 80, Created)
DT   14-NOV-2006 (Rel. 89, Last updated, Version 7)
DE   Candida glabrata strain CBS138 chromosome H complete
sequence.
KW   .
OS   Candida glabrata CBS 138
OC   Eukaryota; Fungi; Ascomycota; Saccharomycotina;
Saccharomycetes;
OC   Saccharomycetales; mitosporic Saccharomycetales;
Candida.
RN   [1]
RX   DOI; 10.1038/nature02579.
RX   PUBMED; 15229592.
RG   Genolevures
RA   Dujon B., Sherman D., Fischer G., Durrens P.,
Casaregola S., Lafontaine I.,
RA   de Montigny J., Marck C., Neuveglise C., Talla E.,
Goffard N., Frangeul L.,
RA   Aigle M., Anthouard V., Babour A., Barbe V., Barnay
S., Blanchin S.,
RA   Beckerich J.M., Beyne E., Bleykasten C., Boisrame A.,
Boyer J.,
RA   Cattolico L., Confanioleri F., de Daruvar A., Despons
L., Fabre E.,
RA   Fairhead C., Ferry-Dumazet H., Groppi A., Hantraye
F., Hennequin C.,
RA   Jauniaux N., Joyet P., Kachouri R., Kerrest A.,
Koszul R., Lemaire M.,
RA   Lesur I., Ma L., Muller H., Nicaud J.M., Nikolski M.,
Oztas S.,
RA   Ozier-Kalogeropoulos O., Pellenz S., Potier S.,
Richard G.F., Straub M.L.,
RA   Suleau A., Swennene D., Tekaia F., Wesolowski-Louvel
M., Westhof E.,
RA   Wirth B., Zeniou-Meyer M., Zivanovic I., Bolotin-
Fukuhara M., Thierry A.,
RA   Bouchier C., Caudron B., Scarpelli C., Gaillardin C.,
Weissenbach J.,
RA   Wincker P., Souciet J.L.;
RT   "Genome evolution in yeasts";
RL   Nature 430(6995):35-44(2004).
RN   [2]
RA   Frangeul L., Sherman D.;
RT   ;
RL   Submitted (05-MAY-2004) to the EMBL/GenBank/DDBJ
databases.
RL   B. Dujon, Institut Pasteur, Unite de Genetique
Moleculaire des Levures, 25
RL   rue du Docteur Roux, 75724 Paris Cedex 15, FRANCE. E-
mail:
RL   lfrangeu@pasteur.fr E-mail: david@labri.fr
DR   CABRI; CBS 138.
CC   This is the first release of the complete genome
sequence. The Candida
CC   glabrata sequence is made of 13 chromosomes (A to M).
```

	These sequences CC contains 6 gaps (artificially closed) indicated by "gap" features and CC about ten colinearity problems indicated by "misc_difference" features. CC An update of the sequence will be submitted in few weeks.

- **Γραφική απεικόνιση** του αποτελέσματος (**Visual Fasta**)



- **Εμφάνιση της έρευνας σε μορφή γλώσσας XML** (παραθέτουμε τμήμα της έρευνας σε γλώσσα XML λόγω του μεγάλου μήκους αυτής)

<EBIApplicationResultxsi:noNamespaceSchemaLocation="http://www.ebi.ac.uk/schema/ApplicationResult.xsd">

<Header>

<program name="fasta" version="3.4t25 Sept 2, 2005" citation="PMID:3162770"/>

<commandLine command="//ebi/extserv/bin/fasta34t25/fasta34_t -l /ebi/services/idata/v1679/fastacfg/fasta3db -Q -H -n -b 50 -d 50 -E 10.0 -f -14 -g -4 @:1- +emfun+ 6"/>

–

<parameters>

–

<sequences total="1">

<sequence number="1" name="ATGTGCTGTAGCTGTAGTGCTGTAGTCGTA" type="n" length="30"/>

</sequences>

–

<databases total="1" sequences="159875" letters="331241143">
<database number="1" name="emfun" type="n"/>
</databases>
<scores>50</scores>
<alignments>50</alignments>
<expectationUpper>10.0</expectationUpper>
<gapOpen>-14</gapOpen>
<gapExtension>-4</gapExtension>
<moleculeType>d</moleculeType>
<ktup>6</ktup>
<histogram>false</histogram>
<sequenceRange>1-</sequenceRange>
</parameters>
<timeInfo start="2006-12-03T23:41:29+00:00" end="2006-12-03T23:41:42+00:00" search="PT23.480S"/>
</Header>
–
<SequenceSimilaritySearchResult>
–
<hits total="12">
–
<hit number="1" database="emfun" id="CR380954" ac="CR380954" length="1050361" description="Candida glabrata strain CBS1">
–
<alignments total="1">
–
<alignment number="1">
<initn>78</initn>
<init1>75</init1>
<opt>92</opt>
<zScore>104.2</zScore>
<bits>32.1</bits>
<expectation>2.2</expectation>
<smithWatermanScore>92</smithWatermanScore>
<identity>81.250</identity>

<ungapped>81.250</ungapped>

<overlap>32</overlap>

και ούτω καθεξής…

Παρατηρήσαμε πως το πρόγραμμα FASTA εκτελείται σε χρόνο αρκετά ή μεγαλύτερο από έρευνες που κάναμε μέσω του συστήματος BLAST (σε οποιαδήποτε εκδοχή του), κάτι που αναμέναμε καθώς το πρόγραμμα FASTA είναι πιο ευαίσθητο, άρα και βραδύτερο.

3.4.3 *CLUSTAL*

Μία εναλλακτική τεχνική που χρησιμοποιείται ευρέως, είναι αυτή της πολλαπλής ευθυγραμμισης. Η μέθοδος αυτή χρησιμοποιείται για σκοπούς όπως ο χαρακτηρισμός οικογενειών πρωτεϊνών, και η ανίχνευση ή η επίδειξη ομολογίας μεταξύ νέων ακολουθιών και άλλων ήδη γνωστών οικογενειών ακολουθιών.

Στη μέθοδο αυτή ευθυγραμμίζουμε πολλές ακολουθίες και επιλέγουμε μία νέα η οποία έχει ως στοιχεία αυτά που εμφανίζονται περισσότερες φορές σε κάθε στήλη (Σχήμα 3.13).

A	C	G	T	T	-	G	T	A
C	G	T	A	G	G	C	A	A
C	A	T	G	T	A	C	T	C
G	T	C	A	G	G	T	A	C
T	G	T	C	A	G	T	G	C
C	G	A	A	G	-	T	A	T
C	**G**	**T**	**A**	**G**	**G**	**T**	**A**	**C**

Σχήμα 3.13: Πολλαπλή ευθυγραμμιση

Στο παράδειγμα αυτό, η ακολουθία που επιλέγεται είναι η CGTAGGTAC και ονομάζεται ομόφωνη ακολουθία (consensus sequence). Συνήθως σε αυτή τη διαδικασία θέτουμε ακολουθίες αμινοξέων που πιστεύουμε πως έχουν παρόμοια δομή αλλά και ακολουθίες νουκλεοτιδίων για σκοπούς όπως η διαπίστωση εξελικτικών αποστάσεων γονιδίων. Λόγω της μεγάλης πολυπλοκότητας αυτής της μεθόδου, χρησιμοποιούμε προσεγγιστικές τεχνικές, όπως η αναγωγή των πολλαπλών ευθυγραμμίσεων σε μία σειρά από ευθυγραμμίσεις ακολουθιών κατά ζεύγη και στη συνέχεια σύγκριση των αποτελεσμάτων, ή η χρήση μαρκοβιανών μοντέλων. Ένα πακέτο λογισμικού που παράγει πολλαπλές ευθυγραμμίσεις είναι το CLUSTAL και άλλα παρόμοια.

Το πρόγραμμα CLUSTAL υπολογίζει το καλύτερο ταίριασμα των ακολουθιών εισόδου και στη συνέχεια τις ευθυγραμμίζει καθ'όλο το μήκος τους (κάνει καθολική ευθυγράμμιση), έτσι ώστε οι ταυτοσημίες ή οι διαφορές συμβόλων να είναι ορατές. Η οπτική θεώρηση των αποτελεσμάτων είναι κριτικής σημασίας για την κατανόησή τους και τα προγράμματα που τα παρουσιάζουν κάνουν χρήση ποικιλίας χρωμάτων για να δώσουν έμφαση σε περιοχές που μας ενδιαφέρουν.

Στη συνέχεια, μέσω της ιστοσελίδας του EBI ([Ebi07]) εκτελέσαμε το πρόγραμμα clustal δίνοντας του ως είσοδο (input) πέντε τυχαίες νουκλεοτιδικές ακολουθίες.

3.4.3.1 *Εκτέλεση του CLUSTAL*

```
>seq1
ATCGTACGTAGCTAGTCGTAGCTACGATGCTAGCTAGCTGTA
>seq2
GCATGCTGATCGTGACTAGTACACATAGAAAACTGTACGTAG
>seq3
CGATACGTACGTACGATCGATCGATCGATCGTAGCTGATCGA
>seq4
CGTAGCTGTAGCTAGCTGATCGATAACGATAGCTGGGCTTAA
>seq5
GTGTGCTGTAGTAGATAGGTCGCATCGATTATCGATTGGCGA
```

Η έξοδος της εκτέλεσης(output) μετά από 11 δευτερόλεπτα ήταν η παρακάτω:

```
Sequence 4: seq4             42 bp
Sequence 5: seq5             42 bp
Start of Pairwise alignments
Aligning...
Sequences (1:2) Aligned. Score:  19
Sequences (1:3) Aligned. Score:  33
Sequences (1:4) Aligned. Score:  23
Sequences (1:5) Aligned. Score:  35
Sequences (2:3) Aligned. Score:  19
Sequences (2:4) Aligned. Score:  23
Sequences (2:5) Aligned. Score:  40
Sequences (3:4) Aligned. Score:  26
Sequences (3:5) Aligned. Score:  45
Sequences (4:5) Aligned. Score:  47
Guide tree        file created:   [/ebi/extserv/clustalw-
work/interactive/2006120717/clustalw-20061207-17583250.dnd]
Start of Multiple Alignment
There are 4 groups
Aligning...
Group 1: Sequences:   2      Score:411
Group 2: Sequences:   3      Score:454
Group 3: Sequences:   2      Score:423
Group 4: Sequences:   5      Score:344
Alignment Score 488
CLUSTAL-Alignment file created  [/ebi/extserv/clustalw-
work/interactive/2006120717/clustalw-20061207-17583250.aln]
```

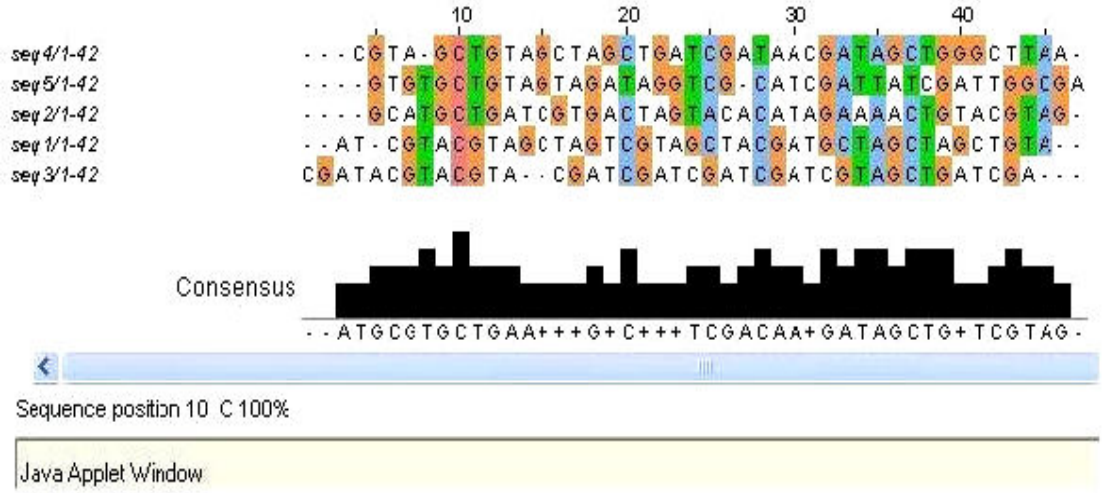
ενώ η απεικόνιση της ευθυγράμμισης είναι

Alignment

```
CLUSTAL W (1.83) multiple sequence alignment

seq4            ---CGTA-GCTGTAGCTAGCTGATCGATAACGATAGCTGGGCTTAA- 42
seq5            ----GTGTGCTGTAGTAGATAGGTCG-CATCGATTATCGATTGGCGA 42
seq2            ----GCATGCTGATCGTGACTAGTACACATAGAAAACTGTACGTAG- 42
seq1            --AT-CGTACGTAGCTAGTCGTAGCTACGATGCTAGCTAGCTGTA-- 42
seq3            CGATACGTACGTA--CGATCGATCGATCGATCGTAGCTGATCGA--- 42
                          *
```

Στην απεικόνιση αυτή παρατηρούμε την χρήση κενών για ευθυγράμμιση των ακολουθιών αλλά και την χρήση του συμβόλου * που υποδεικνύει την ταυτοσημία συμβόλων σε συγκεκριμένη θέση. Η γραφική απεικόνιση της ευθυγράμμισης είναι η παρακάτω και εμφανίζεται ως εφαρμογή Java.



Παρατηρούμε πως σε κάθε θέση γίνεται σύγκριση και το στοιχείο που εμφανίζεται τις περισσότερες φορές επιλέγεται ως στοιχείο της ομόφωνης ακολουθίας. Για παράδειγμα στη θέση 10 υπάρχει απόλυτη (100%) ομοφωνία συμβόλων C (κυτοσίνης), ενώ αντίθετα σε όσες θέσεις υπάρχει εμφάνιση διαφορετικών συμβόλων σε ίσο πλήθος τοποθετείται το σύμβολο + στην ομόφωνη ακολουθία. Τέλος παρέχεται η δυνατότητα απεικόνισης της πολλαπλής ευθυγράμμισης με τη μορφή φυλογενετικού δένδρου. Το CLUSTAL βασιζόμενο στις κατά ζεύγη ευθυγραμμίσεις ζευγών ακολουθιών, έχει υπολογίσει το βαθμό ομοιότητας κάθε ακολουθίας με οποιαδήποτε άλλη. Ο βαθμός ομοιότητας αυτός δείχνει και την εξελικτική απόσταση ανάμεσα σε δύο ακολουθίες, κάτι που τα φυλογενετικά δένδρα απεικονίζουν. Συγκεκριμένα το φυλογενετικό δένδρο των ακολουθιών ήταν:

```
(
(
seq1:0.35317,
seq3:0.31349)
:0.06845,
seq2:0.40774,
(
seq4:0.34226,
seq5:0.18155)
:0.00893);
```

3.5 *Σύνοψη και Συμπεράσματα*

Τα προγράμματα ευθυγράμμισης ακολουθιών που μελετήσαμε είναι τα πλέον δημοφιλή εργαλεία ευθυγραμμίσεων ακολουθιών στο χώρο των Βιοεπιστημών. Όσον αφορά τη μέθοδο

τοπικής ευθυγράμμισης βιολογικών ακολουθιών, περιγράψαμε και εκτελέσαμε τα προγράμματα BLAST και FASTA.

Το BLAST που δημοσιεύθηκε το 1990 δέχεται σε καθημερινή βάση δεκάδες χιλιάδες διαδικτυακές επισκέψεις και εκτελέσεις. Χαρακτηρίζεται ως αρκετά φιλικό προς το χρήστη, αρκετά ταχύ και εξαιρετικά αποτελεσματικό. Το πρόγραμμα FASTA που εμφανίστηκε το 1985, είναι επίσης δημοφιλές, όχι όμως βέβαια σε τόσο μεγάλο βαθμό όσο το BLAST, ενώ είναι και ιδιαίτερα φιλικό παρέχοντας δυνατότητες όπως η εμφάνιση σχολίων. Και τα δύο αυτά προγράμματα είναι ευριστικές μέθοδοι αναζήτησης τοπικών ευθυγραμμίσεων. Αυτό συνεπάγεται πως και τα δύο αυτά προγράμματα αδυνατούν να εγγυηθούν της εύρεσης της βέλτιστης τοπικής ευθυγράμμισης δύο ακολουθιών και συχνά χάνουν περιοχές σημαντικού ενδιαφέροντος. Παρά το γεγονός αυτό όμως, αποτελούν δύο ταχύτατα εργαλεία που εντοπίζουν μία λύση πολύ κοντά στη βέλτιστη, σε πολύ μικρό χρόνο. Το πρόγραμμα FASTA θα μπορούσε να χαρακτηριστεί ως λίγο πιο ευαίσθητο από το BLAST, ενώ το τελευταίο είναι λίγο ταχύτερο. Βεβαίως υπάρχουν και περιπτώσεις όπου το FASTA είναι ταχύτερο. Μάλιστα σε κάποιες εφαρμογές του FASTA υπάρχει η δυνατότητα να περιορίσουμε την έρευνα σε ένα υποσύνολο μιας βάσης δεδομένων, κάτι που δεν προσφέρει το BLAST. Με αυτόν τον τρόπο μπορούμε και να επιταχύνουμε μία έρευνα, αλλά και να ασχοληθούμε μόνο με συγκεκριμένες ακολουθίες που επιθυμούμε. Επίσης, είναι σημαντικό να τονιστεί πως σημαντικό ρόλο στο χρόνο εκτέλεσης αυτών των προγραμμάτων παίζει και το πόσο υπερφορτωμένοι (overloaded) θα είναι οι εξυπηρετητές (servers) των ιστοσελίδων που χρησιμοποιούμε (λόγω του πλήθους των ταυτόχρονων ερευνών, είναι σχεδόν πάντα υπερφορτωμένοι).

Από την άλλη, το πρόγραμμα CLUSTAL που επίσης μελετήθηκε, είναι το δημοφιλέστερο εργαλείο πολλαπλών ευθυγραμμίσεων, είναι ιδιαίτερα φιλικό και αρκετά ταχύ. Παρόλα αυτά λειτουργεί ικανοποιητικά κυρίως σε περιπτώσεις όπου οι ευθυγραμμιζόμενες ακολουθίες είναι αρκετά συσχετισμένες μεταξύ τους. Τα παραπάνω χαρακτηριστικά των προγραμμάτων απεικονίζονται στον Πίνακα 3.8.

Χαρακτηριστικά	FASTA	BLAST	CLUSTAL
Λειτουργία	Τοπική Ευθυγράμμιση κατά ζεύγη	Τοπική Ευθυγράμμιση κατά ζεύγη	Πολλαπλή Καθολική Ευθυγράμμιση
Ταχύτητα	√√	√√√	√√√
Ευαισθησία	√√√	√√	√
Φιλικότητα	√√√	√√	√√
Δημοφιλία	√√	√√√	√√√

Πίνακας 3.8: Χαρακτηριστικά προγραμμάτων ευθυγράμμισης βιολογικών ακολουθιών

4

Εξελικτικά Δένδρα

4.1 *Κατασκευή Εξελικτικών (φυλογενετικών) δένδρων*

Εφόσον η εξέλιξη διαδραματίζει πρωτεύοντα ρόλο στη βιολογία, είναι απολύτως φυσιολογική η προσπάθεια απεικόνισής της, χρησιμοποιώντας δένδρα που αναφέρονται ως φυλογενετικά δένδρα.

Τα φύλλα των δένδρων αυτών αναπαριστούν διαφόρους οργανισμούς, είδη, γονιδιακές ή πρωτεϊνικές ακολουθίες. Ένας εσωτερικός κόμβος P αντιστοιχεί σε έναν οργανισμό (είδος ή ακολουθία), του οποίου η ύπαρξη θεωρείται δεδομένη και που η εξέλιξή του οδήγησε στους οργανισμούς που είναι άμεσοι απόγονοί του και απεικονίζονται ως τα κλαδιά που γεννούνται από τον κόμβο P.

Το κίνητρο για τη δημιουργία των φυλογενετικών δένδρων είναι η έκφραση σε γραφική μορφή των αποτελεσμάτων πολλαπλών ευθυγραμμίσεων που δείχνουν τις συσχετίσεις μεταξύ ζευγών ή ομάδων ακολουθιών. Τα δένδρα αυτά είναι πιθανόν να αποκαλύπτουν εξελικτικές ασυνέπειες που πρέπει να διερευνηθούν. Με αυτήν την έννοια λοιπόν η κατασκευή των δένδρων αυτών καθιστά έγκυρες ή άκυρες υποθέσεις που έχουν γίνει για πιθανούς προγόνους διαφόρων οργανισμών.

Τα δένδρα αυτά μπορούν να προκύψουν μετά από ευθυγράμμιση ακολουθιών. Αν έχουμε μία πολλαπλή ευθυγράμμιση, τότε από όλες τις ακολουθίες μπορούν να επιλεχθούν δύο των οποίων η ευθυγράμμιση αποδίδει το μεγαλύτερο αποτέλεσμα (σκορ). Στη συνέχεια

δημιουργείται ένας κόμβος που αναπαριστά τον άμεσο πρόγονο των δύο αυτών ακολουθιών. Στη συνέχεια η δυσκολία εμφανίζεται στην ανακατασκευή, μεταξύ πολλών πιθανών ακολουθιών, της ακολουθίας που αναπαριστά καλύτερα τα παιδιά του κόμβου. Αυτό απαιτεί εφευρετικότητα και διαίσθηση. Αφού γίνει η επιλογή, η ακολουθία-πρόγονος αντικαθιστά τα δύο παιδιά της και ο αλγόριθμος συνεχίζεται αναδρομικά έως ότου καθορισθεί ένας κόμβος-ρίζα. Το αποτέλεσμα είναι ένα δυαδικό δένδρο του οποίου η ρίζα αναπαριστά την αρχέγονη ακολουθία που θεωρείται πως παρήγαγε όλες τις άλλες.

Υπάρχουν διάφοροι τύποι δένδρων που χρησιμοποιούνται στις Βιοεπιστήμες, οι σημαντικότεροι εκ των οποίων είναι οι παρακάτω:

- Δένδρα χωρίς ρίζα (αδόκιμος όρος) είναι αυτά που καθορίζουν τις αποστάσεις (διαφορές) μεταξύ ειδών. Το μήκος του μονοπατιού μεταξύ δύο οποιωνδήποτε φύλλων αναπαριστά τις συσσωρευμένες διαφορές.
- Κλαδογράμματα ονομάζονται τα δένδρα με ρίζα, όπου τα μήκη των κλαδιών δεν έχουν καμία σημασία.
- Τα φυλογράμματα είναι επεκτεταμένα κλαδογράμματα στα οποία το μήκος ενός κλαδιού ποσοτικοποιεί το πλήθος των γενετικών αλλαγών που συνέβησαν μεταξύ ενός δοσμένου κόμβου και του άμεσου προγόνου του.
- Πολυμετρικά δένδρα είναι τα φυλογράμματα στα οποία οι συσσωρευμένες αποστάσεις από μία ρίζα σε κάθε ένα από τα φύλλα ποσοτικοποιούνται από τον ίδιο αριθμό. Τα πολυμετρικά δένδρα είναι λοιπόν αυτά που παρέχουν την περισσότερη πληροφορία για τις εξελικτικές αλλαγές. Είναι επίσης τα δένδρα που κατασκευάζονται δυσκολότερα.

Οι παραπάνω ορισμοί υποδεικνύουν την ύπαρξη κάποιου είδους μοριακού ρολογιού με βάση το οποίο οι μεταλλάξεις συμβαίνουν με κάποιον προβλέψιμο ρυθμό, καθώς και την ύπαρξη γραμμικής εξάρτησης μεταξύ του χρόνου και του πλήθους των αλλαγών. Οι ρυθμοί αυτοί είναι γνωστό πως διαφέρουν για εκάστοτε οργανισμό, ή ακόμα και για τα διάφορα συστατικά των κυττάρων (π.χ. για το DNA και για τις πρωτεΐνες). Αυτό δείχνει τη σπουδαιότητα αλλά και την εξαιρετική δυσκολία που εμφανίζεται στη δημιουργία σωστών φυλογενετικών δένδρων.

4.2 *Αλγόριθμοι*

Λόγω της προαναφερθείσας δυσκολίας αλλά και σημασίας της δημιουργίας σωστών φυλογενετικών δένδρων απαιτείται η βοήθεια αλγορίθμων. Οι σημαντικότεροι αυτών, είναι οι βασιζόμενες σε αποστάσεις μέθοδοι (distance based methods), αλγόριθμοι UPGMA και Neighbor-Joining. Με τον όρο “distance based methods” εννοούμε πως αυτές οι μέθοδοι στηρίζονται σε εξελικτικές αποστάσεις και κατασκευάζουν δένδρα, όπου η απόσταση μεταξύ δύο στοιχείων στο δένδρο ισοδυναμεί με την εξελικτική τους απόσταση.

4.2.1 *Αλγόριθμος UPGMA*

Ένας αλγόριθμος που χρησιμοποιείται συχνά για να κατασκευαστούν δένδρα χωρίς ρίζα ονομάζεται UPGMA (Unweighted Pair Group Method using Arithmetic Averages - Μη Ζυγισμένη Μέθοδος Ομαδοποίησης Ζευγών με χρήση Αριθμητικών Μέσων), αναπτύχθηκε από τους Michener και Sokal το 1957 και έγινε αρκετά δημοφιλής, κυρίως λόγω της απλότητάς του. Στην κατανόηση της λειτουργίας του μας βοηθά το παράδειγμα των πολλαπλών ευθυγραμμίσεων. Ας υποθέσουμε λοιπόν πως μπορούμε να ποσοτικοποιήσουμε τις αποστάσεις μεταξύ δύο οποιωνδήποτε ευθυγραμμίσεων κατά ζεύγη (Το αποτέλεσμα-σκορ της μεθόδου του δυναμικού προγραμματισμού για τις ευθυγραμμίσεις αυτές θα απέδιδε την επιθυμητή πληροφορία για τις αποστάσεις: όσο υψηλότερο είναι το αποτέλεσμα, τόσο μικρότερη η απόσταση μεταξύ δύο ακολουθιών). Οι ποικίλες αποστάσεις μπορούν να συγκεντρωθούν σε έναν πίνακα τριγωνικής μορφής. Ο Πίνακας 4.1 παρουσιάζει τις εξελικτικές αποστάσεις μεταξύ κάποιων ειδών και δημιουργήθηκε από τον Sarich το 1969:

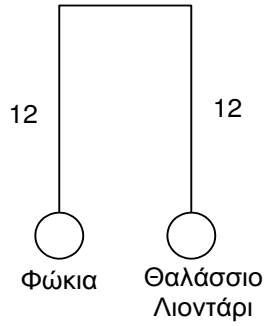
	Σκύλος	Αρκούδα	Ρακούν	Νυφίτσα	Φώκια	Θαλάσσιο λιοντάρι	Γάτα	Πίθηκος
Σκύλος	0	32	48	51	50	48	98	148
Αρκούδα		0	26	34	29	33	84	136
Ρακούν			0	42	44	44	92	152
Νυφίτσα				0	44	38	86	142
Φώκια					0	24	89	142
Θαλάσσιο λιοντάρι						0	90	142
Γάτα							0	148
Πίθηκος								0

Πίνακας 4.1: Αρχικός Πίνακας UPGMA

Ο αλγόριθμος UPGMA λειτουργεί ως εξής: Θεωρούμε δύο στοιχεία E_1 και E_2 με την ελάχιστη απόσταση D μεταξύ τους τα οποία στη συνέχεια ομαδοποιούνται σε ένα στοιχείο (E_1,E_2). Οι αντίστοιχες δύο στήλες και γραμμές συγχωνεύονται σε μία και στο δένδρο που δημιουργείται, απεικονίζονται τα στοιχεία να συνδέονται με κλαδιά μήκους $\frac{D}{2}$. Έπειτα κατασκευάζεται ένας ενημερωμένος πίνακας στον οποίον οι νέες αποστάσεις για τα ομαδοποιημένα στοιχεία προκύπτουν από τον τύπο $D_{(ij),k} = D_{k,(ij)} = \frac{w_i \times D_{ik} + w_j \times D_{jk}}{w_i + w_j}$,

όπου w είναι το βάρος του κάθε στοιχείου και ορίζεται ως το πλήθος των στοιχείων που εμπεριείχε και D είναι η απόσταση του κάθε στοιχείου προς κάποιο άλλο. Ο αλγόριθμος συνεχίζεται ώσπου όλοι οι κόμβοι να εκφυλιστούν (collapsed) σε έναν μοναδικό κόμβο.

Στο παράδειγμά μας λοιπόν παρατηρούμε πως η ελάχιστη απόσταση (ελάχιστο στοιχείο στον πίνακα) είναι μεταξύ φώκιας και θαλάσσιου λιονταριού, και είναι ίση με 24. Το δένδρο είναι:



Σχήμα 4.1: UPGMA Βήμα 1

Στο πρώτο βήμα λοιπόν τα στοιχεία Φώκια και Θαλάσσιο Λιοντάρι ενώνονται και οι νέες αποστάσεις για το στοιχείο (Φώκια, Θαλάσσιο Λιοντάρι) προκύπτουν ως εξής:

$$D_{(\text{Σκύλος},(\text{Φώκια-Θαλάσσιο Λιοντάρι}))} = \frac{1\times 50 + 1\times 48}{1+1} = \frac{50+48}{2} = 49$$

$$D_{(\text{Αρκούδα},(\text{Φώκια-Θαλάσσιο Λιοντάρι}))} = \frac{1\times 29 + 1\times 33}{1+1} = \frac{29+33}{2} = 31$$

$$D_{(\text{Ρακούν},(\text{Φώκια-Θαλάσσιο Λιοντάρι}))} = \frac{1\times 44 + 1\times 44}{1+1} = \frac{44+44}{2} = 44$$

$$D_{(\text{Νυφίτσα},(\text{Φώκια-Θαλάσσιο Λιοντάρι}))} = \frac{1\times 44 + 1\times 38}{1+1} = \frac{44+38}{2} = 41$$

$$D_{((\text{Φώκια-Θαλάσσιο Λιοντάρι}),\text{Γάτα})} = \frac{1\times 89 + 1\times 90}{1+1} = \frac{89+90}{2} = 89.5$$

$$D_{((\text{Φώκια-Θαλάσσιο Λιοντάρι}),\text{Πίθηκος})} = \frac{1\times 142 + 1\times 142}{1+1} = \frac{142+142}{2} = 142$$

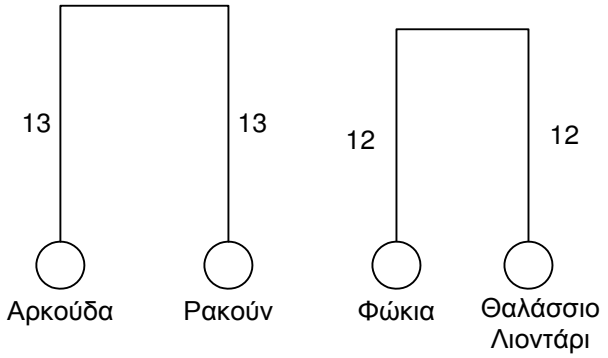
Έτσι η μορφή του πίνακα θα είναι :

	Σκύλος	Αρκούδα	Ρακούν	Νυφίτσα	Φώκια Θαλάσσιο λιοντάρι	Γάτα	Πίθηκος
Σκύλος	0	32	48	51	49	98	148
Αρκούδα		0	26	34	31	84	136
Ρακούν			0	42	44	92	152
Νυφίτσα				0	41	86	142
Φώκια Θαλάσσιο Λιοντάρι					0	89.5	142
Γάτα						0	148
Πίθηκος							0

Πίνακας 4.2: UPGMA Βήμα 1

Στο επόμενο βήμα η ελάχιστη απόσταση είναι μεταξύ αρκούδας και ρακούν, ίση με 26. Ακολουθείται η ίδια λογική:

Το δένδρο θα είναι



Σχήμα 4.2: UPGMA Βήμα 2

Τώρα οι αποστάσεις υπολογίζονται ως εξής:

$$D_{(\text{Σκύλος},(\text{Αρκούδα-Ρακούν}))} = \frac{1\times 32 + 1\times 48}{1+1} = \frac{32+48}{2} = 40$$

$$D_{((\text{Αρκούδα-Ρακούν}),\text{Νυφίτσα})} = \frac{1\times 34 + 1\times 42}{1+1} = \frac{34+42}{2} = 38$$

$$D_{((\text{Αρκούδα-Ρακούν}),(\text{Φώκια-Θαλάσσιο Λιοντάρι}))} = \frac{1\times 31 + 1\times 44}{1+1} = \frac{31+44}{2} = 37.5$$

$$D_{((\text{Αρκούδα-Ρακούν}),\text{Γάτα})} = \frac{1\times 84 + 1\times 92}{1+1} = \frac{84+92}{2} = 88$$

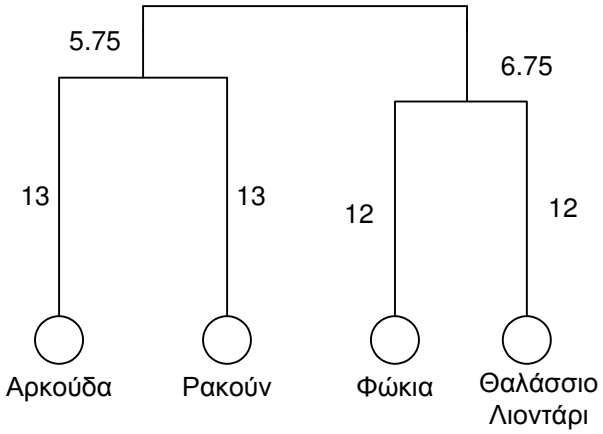
$$D_{((\text{Αρκούδα-Ρακούν}),\text{Πίθηκος})} = \frac{1\times 136 + 1\times 152}{1+1} = \frac{136+152}{2} = 144$$

και η μορφή του πίνακα θα είναι:

	Σκύλος	Αρκούδα Ρακούν	Νυφίτσα	Φώκια Θαλάσσιο λιοντάρι	Γάτα	Πίθηκος
Σκύλος	0	40	51	49	98	148
Αρκούδα Ρακούν		0	38	37. 5	88	144
Νυφίτσα			0	41	86	142
Φώκια Θαλάσσιο λιοντάρι				0	89.5	142
Γάτα					0	148
Πίθηκος						0

Πίνακας 4.3: UPGMA Βήμα 2

Εδώ η ελάχιστη απόσταση είναι μεταξύ των στοιχείων (Φώκια, Θαλάσσιο Λιοντάρι) και (Αρκούδα, Ρακούν) ίση με 37.5. Το δένδρο θα είναι:



Σχήμα 4.3: UPGMA Βήμα 3

και ο πίνακας θα είναι:

	Σκύλος	Αρκούδα Ρακούν Φώκια Θαλάσσιο Λ.	Νυφίτσα	Γάτα	Πίθηκος
Σκύλος	0	44.5	51	98	148
Αρκούδα Ρακούν Φώκια Θαλάσσιο Λ.		0	39.5	88.75	143
Νυφίτσα			0	86	142
Γάτα				0	148
Πίθηκος					0

Πίνακας 4.4: UPGMA Βήμα 3

όπου τα στοιχεία υπολογίστηκαν ως εξής:

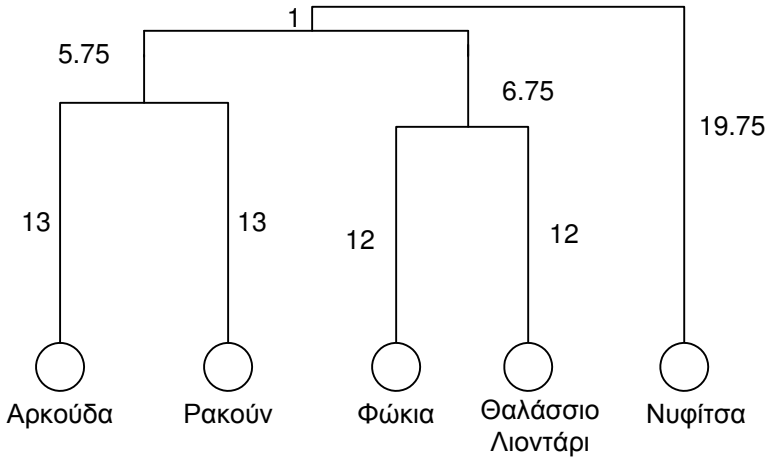
$$D_{(\Sigma\kappa\acute{\upsilon}\lambda\text{ος},(\text{ΑΡΦΘλ}))} = \frac{2\times 40 + 2\times 49}{2+2} = \frac{2\times(40+49)}{4} = \frac{40+49}{2} = 44.5$$

$$D_{((\text{ΑΡΦΘλ}),\text{Νυφίτσα})} = \frac{2\times 38 + 2\times 41}{2+2} = \frac{2\times(38+41)}{4} = \frac{38+41}{2} = 39.5$$

$$D_{((\text{ΑΡΦΘλ}),\Gamma\acute{\alpha}\tau\alpha)} = \frac{2\times 88 + 2\times 89.5}{2+2} = \frac{2\times(88+89.5)}{4} = \frac{88+89.5}{2} = 88.75$$

$$D_{((\text{ΑΡΦΘλ}),\text{Γάτα})} = \frac{2\times144+2\times142}{2+2} = \frac{2\times(144+142)}{4} = \frac{144+142}{2} = 143$$

Τώρα η ελάχιστη απόσταση είναι μεταξύ του στοιχείου (Αρκούδα, Ρακούν, Φώκια, Θαλάσσιο Λιοντάρι) και της Νυφίτσας, ίση με 39.5. Τότε το δένδρο είναι:



Σχήμα 4.4: UPGMA Βήμα 4

και ο πίνακας θα γίνει

	Σκύλος	Αρκούδα Ρακούν Νυφίτσα Φώκια Θαλάσσιο Λ.	Γάτα	Πίθηκος
Σκύλος	0	45.8	98	148
Αρκούδα Ρακούν Νυφίτσα Φώκια Θαλάσσιο Λ.		0	88.2	142.8
Γάτα			0	148
Πίθηκος				0

Πίνακας 4.5: UPGMA Βήμα 4

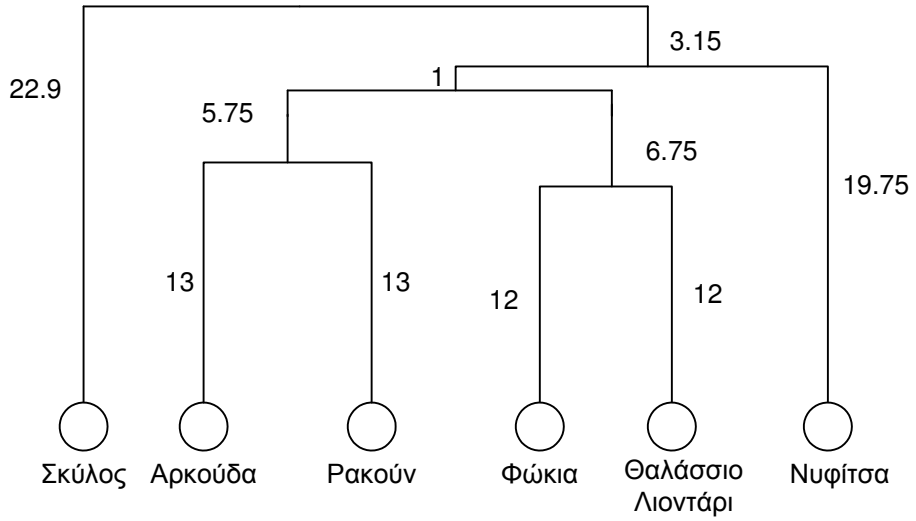
όπου τα στοιχεία υπολογίστηκαν ως εξής:

$$D_{(\text{Σκύλος},(\text{ΑΡΦΘλΝ}))} = \frac{4\times44.5+1\times51}{4+1} = \frac{178+51}{5} = \frac{229}{5} = 45.8$$

$$D_{((\text{ΑΡΦΘλΝ}),\text{Γάτα})} = \frac{4\times88.75+1\times86}{4+1} = \frac{355+86}{5} = \frac{441}{5} = 88.2$$

$$D_{((\text{ΑΡΦΘλΝ}),\text{Πίθηκος})} = \frac{4\times143+1\times142}{4+1} = \frac{572+142}{5} = \frac{714}{5} = 142.8$$

ενώ το δένδρο θα είναι



Σχήμα 4.5: UPGMA Βήμα 5

Τώρα η ελάχιστη απόσταση είναι το 45.8 και η εκτέλεση συνεχίζει ομοίως με πριν.

	Σκύλος Αρκούδα Ρακούν Νυφίτσα Φώκια Θαλάσσιο Λ.	Γάτα	Πίθηκος
Σκύλος Αρκούδα Ρακούν Νυφίτσα Φώκια Θαλάσσιο Λ.	0	89.833	143.66
Γάτα		0	148
Πίθηκος			0

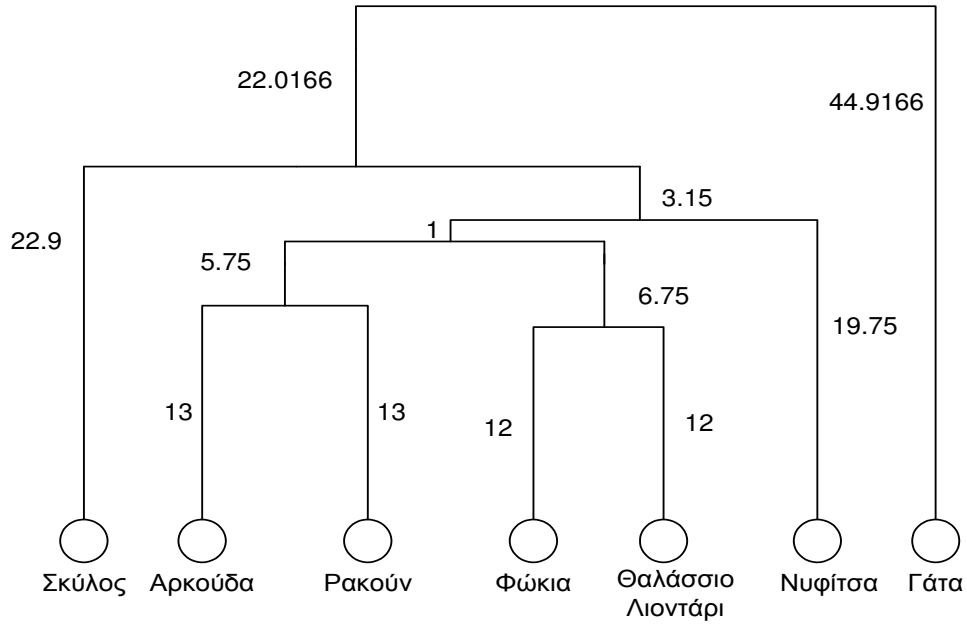
Πίνακας 4.6: UPGMA Βήμα 5

όπου οι αποστάσεις υπολογίστηκαν ως εξής

$$D_{((\Sigma A P \Phi \Theta \lambda N),\Gamma \acute{\alpha} \tau \alpha)} = \frac{1 \times 98 + 5 \times 88.2}{1+5} = \frac{98 + 441}{6} = \frac{539}{6} = 89.833$$

$$D_{((\Sigma A P \Phi \Theta \lambda N),\Pi \acute{\iota} \theta \eta \kappa o \varsigma)} = \frac{1 \times 148 + 5 \times 142.8}{1+5} = \frac{148 + 714}{6} = \frac{862}{6} = 143.66$$

ενώ το δένδρο είναι:



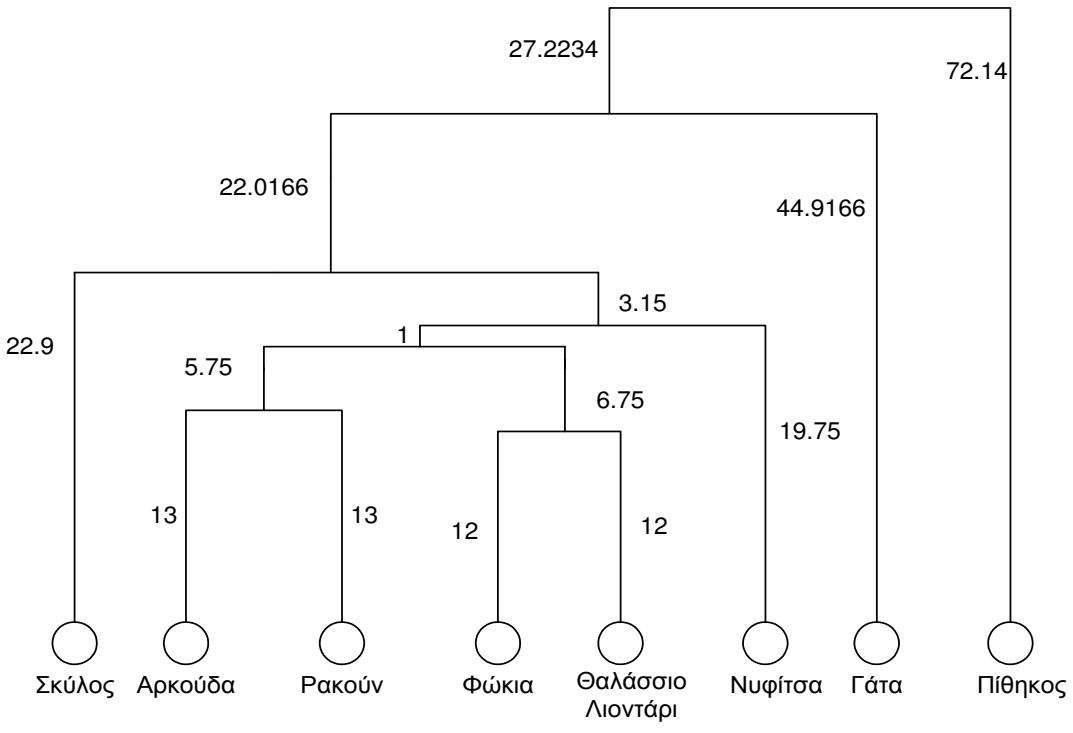
Σχήμα 4.6: UPGMA Βήμα 6

Τώρα η απόσταση, ο πίνακας και η τελική μορφή του δένδρου είναι:

$$D_{((\Sigma\Alpha\Rho\Phi\Theta\lambda N\Gamma),\Pi\acute{\iota}\theta\eta\kappa o\varsigma)} = \frac{6\times 143.66 + 1\times 148}{6+1} = \frac{861.96+148}{7} = \frac{1009.96}{7} = 144.28$$

	ΣΑΡΦΘλΝΓ	Πίθηκος
ΣΑΡΦΘλΝΓ	0	144.28
Πίθηκος		0

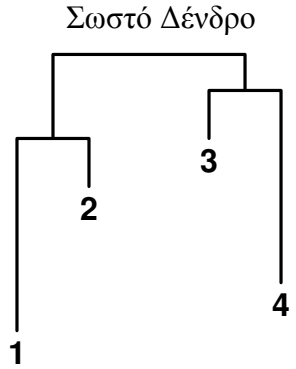
Πίνακας 4.7: UPGMA Βήμα 6



Σχήμα 4.7: UPGMA Τελικό δένδρο

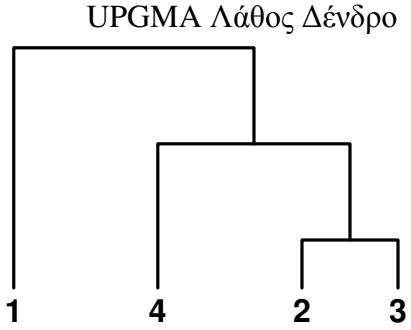
Έχουμε λοιπόν το τελικό δένδρο χωρίς ρίζα στο οποίο απεικονίζονται οι εξελικτικές αποστάσεις μεταξύ των διαφόρων ειδών.

Ο αλγόριθμος αυτός είναι ο απλούστερος και βασίζεται στην απλοϊκή υιοθέτηση ενός σταθερού μοριακού ρολογιού για όλα τα είδη. Αυτή η υπόθεση οδηγεί στο συχνά, αλλά όχι πάντα εσφαλμένο συμπέρασμα ότι η εξέλιξη όλων των ειδών-κλαδιών γίνεται με τον ίδιο ρυθμό. Αποτέλεσμα αυτού είναι ότι στα δένδρα που κατασκευάζει ο αλγόριθμος UPGMA, τα φύλλα βρίσκονται όλα στο ίδιο επίπεδο (βλέπε Σχήμα 4.7). Για παράδειγμα εάν γνωρίζουμε τον πίνακα αποστάσεων τεσσάρων ειδών και γνωρίζουμε ότι το σωστό δένδρο είναι το δένδρο του Σχήματος 4.8,



Σχήμα 4.8: Παράδειγμα δένδρου

ο αλγόριθμος UPGMA θα μας κατασκευάσει εσφαλμένα το παρακάτω δένδρο.



Πρόταση 1. *Προκειμένου ο αλγόριθμος UPGMA να κατασκευάζει το σωστό δένδρο, πρέπει οι αποστάσεις στον πίνακα να είναι πολυμετρικές (ultrametric).*

Μία απόσταση είναι πολυμετρική εάν για κάθε σημείο-στοιχείο i, j, k, ισχύουν τα εξής:

- Είτε οι τρεις αποστάσεις είναι ίσες: $d_{ij} = d_{ik} = d_{jk}$
- Είτε δύο από αυτές είναι ίσες και μία είναι μικρότερη: $d_{jk} < d_{ij} = d_{ik}$.

4.2.2 Αλγόριθμος Neighbor-Joining

Ο αλγόριθμος Ένωσης Γειτόνων-Neighbor-Joining είναι ο πιο δημοφιλής τρόπος κατασκευής δένδρων από μετρήσεις αποστάσεων. Η αρχή του αλγορίθμου αυτού είναι: «*Ενώνω τους κόμβους που είναι κοντά μεταξύ τους και μακριά από κάθε άλλον*». Ο αλγόριθμος αυτός

έρχεται ακριβώς να διορθώσει την συχνά εσφαλμένη υπόθεση του UPGMA περί ίδιου ρυθμού εξέλιξης όλων των κλαδιών ενός δένδρου, καθώς ο πίνακας αποστάσεων προσαρμόζεται στις διαφορές του ρυθμού εξέλιξης του κάθε είδους.

Ο αλγόριθμος αυτός εκτελείται ομοίως βάσει ενός πίνακα που απεικονίζει τις εξελικτικές αποστάσεις (distance-based μέθοδος) μεταξύ των ειδών και αποτελείται από τα εξής βήματα:

- 1) Για κάθε στοιχείο του πίνακα υπολογίζεται το $u_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{n-2}$ όπου D_{ij} είναι η εξελικτική απόσταση του στοιχείου i από το στοιχείο j και n το πλήθος των στοιχείων-ειδών του πίνακα .
- 2) Επιλέγονται τα i και j για τα οποία η παράσταση $D_{ij} - u_i - u_j$ είναι ελάχιστη.
- 3) Τα στοιχεία i και j ενώνονται . Υπολογίζεται το μήκος του κλαδιού από το i ως το νέο κόμβο (v_i) και από το j ως το νέο κόμβο (v_j) ως εξής:

  $v_i = \frac{1}{2} D_{ij} + \frac{1}{2}(u_i - u_j)$ και $v_j = \frac{1}{2} D_{ij} + \frac{1}{2}(u_j - u_i)$.
- 4) Υπολογίζεται η απόσταση μεταξύ του νέου κόμβου (ij) και οποιουδήποτε άλλου στοιχείου ως $D_{(ij),k} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$.
- 5) Τα στοιχεία i και j (οι αντίστοιχες γραμμές και στήλες) διαγράφονται από τον πίνακα και αντικαθίστανται από ένα νέο στοιχείο ij (μία νέα γραμμή και στήλη)
- 6) Αν έχουν απομείνει παραπάνω από δύο κόμβοι οδηγούμαστε στο βήμα 1 και ούτω καθεξής. Αλλιώς οι δύο εναπομείναντες κόμβοι συνδέονται με ένα κλαδί μήκους D_{ij} και η εκτέλεση του αλγορίθμου τερματίζεται.

Ως παράδειγμα για την κατανόηση του αλγορίθμου Neighbor-Joining χρησιμοποιούμε τον πίνακα του Sarich, όπως και στον UPGMA. Έχουμε λοιπόν:

	Σκύλος	Αρκούδα	Ρακούν	Νυφίτσα	Φώκια	Θαλάσσιο Λιοντάρι	Γάτα	Πίθηκος
Σκύλος	0	32	48	51	50	48	98	148
Αρκούδα	32	0	26	34	29	33	84	136
Ρακούν	48	26	0	42	44	44	92	152
Νυφίτσα	51	34	42	0	44	38	86	142
Φώκια	50	29	44	44	0	24	89	142
Θαλάσσιο Λιοντάρι	48	33	44	38	24	0	90	142
Γάτα	98	84	92	86	89	90	0	148
Πίθηκος	148	136	152	142	142	142	148	0

Πίνακας 4.8: Neighbor-Joining Αρχικός Πίνακας

Ακολουθούμε τα βήματα του αλγορίθμου:

1) Υπολογίζουμε τις ποσότητες $u_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{n-2}$ για κάθε i του πίνακα (δηλαδή για κάθε είδος).

Είναι λοιπόν για n=8

$$u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} = \frac{32+48+51+50+48+98+148}{6} = \frac{475}{6} = 79.166$$

$$u_{\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha} = \frac{32+26+34+29+33+84+136}{6} = \frac{374}{6} = 62.333$$

$$u_{\Rho\Alpha\Kappa\Omicron\Upsilon\Nu} = \frac{48+26+42+44+44+92+152}{6} = \frac{448}{6} = 74.666$$

$$u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = \frac{51+34+42+44+38+86+142}{6} = \frac{437}{6} = 72.833$$

$$u_{\Phi\Omega\Kappa\Iota\Alpha} = \frac{50+29+44+44+24+89+142}{6} = \frac{422}{6} = 70.333$$

$$u_{\substack{\Theta\Alpha\Lambda\Alpha\Sigma\Sigma\Iota\Omicron \\ \Lambda\Iota\Omicron\Nu\Tau\Alpha\Rho\Iota}} = \frac{48+33+44+38+24+90+142}{6} = \frac{419}{6} = 69.833$$

$$u_{\Gamma\Alpha\Tau\Alpha} = \frac{98+84+92+86+89+90+148}{6} = \frac{687}{6} = 114.5$$

$$u_{\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma} = \frac{148+136+152+142+142+142+148}{6} = \frac{1010}{6} = 168.333$$

2) Υπολογίζονται οι παραστάσεις $D_{ij} - u_i - u_j$, για όλα τα i και j και επιλέγονται αυτά που την ελαχιστοποιούν. Προφανώς λόγω της τριγωνικότητας του πίνακα ($D_{ij} = D_{ji}$) θα ισχύει ότι $D_{ij} - u_i - u_j = D_{ji} - u_j - u_i$.

Σκύλος-Αρκούδα (i=1, j=2)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha} = 32 - 79.166 - 62.333 = -109.499$$

Σκύλος-Ρακούν (i=1, j=3)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Rho\Alpha\Kappa\Omicron\Upsilon\Nu}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Rho\Alpha\Kappa\Omicron\Upsilon\Nu} = 48 - 79.166 - 74.666 = -105.832$$

Σκύλος-Νυφίτσα (i=1, j=4)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = 51 - 79.166 - 72.833 = -100.999$$

Σκύλος-Φώκια (i=1, j=5)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Phi\Omega\Kappa\Iota\Alpha}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Phi\Omega\Kappa\Iota\Alpha} = 50 - 79.166 - 70.333 = -99.499$$

Σκύλος-Θαλάσσιο Λιοντάρι (i=1, j=6)

$$D_{\substack{\text{ΣΚΥΛΟΣ}\\ \text{ΘΑΛΑΣΣΙΟ}\\ \text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΣΚΥΛΟΣ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\ \text{ΛΙΟΝΤΑΡΙ}}} = 48 - 79.166 - 69.833 = -100.99$$

Σκύλος-Γάτα (i=1, j=7)

$$D_{\substack{\text{ΣΚΥΛΟΣ}\\ \text{ΓΑΤΑ}}} - u_{\text{ΣΚΥΛΟΣ}} - u_{\text{ΓΑΤΑ}} = 98 - 79.166 - 114.5 = -95.666$$

Σκύλος-Πίθηκος (i=1, j=8)

$$D_{\substack{\text{ΣΚΥΛΟΣ}\\ \text{ΠΙΘΗΚΟΣ}}} - u_{\text{ΣΚΥΛΟΣ}} - u_{\text{ΠΙΘΗΚΟΣ}} = 148 - 79.166 - 168.333 = -99.499$$

Σύμφωνα με την προηγούμενη παρατήρησή μας, δεν χρειάζεται να υπολογιστεί η παράσταση για το ζεύγος (ΑΡΚΟΥΔΑ-ΣΚΥΛΟΣ) εφόσον έχει ήδη υπολογιστεί για το ζεύγος (ΣΚΥΛΟΣ –ΑΡΚΟΥΔΑ). Ομοίως αντιμετωπίζονται και τα υπόλοιπα «συμμετρικά» ζεύγη.

Αρκούδα-Ρακούν (i=2, j=3)

$$D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{ΡΑΚΟΥΝ}}} - u_{\text{ΑΡΚΟΥΔΑ}} - u_{\text{ΡΑΚΟΥΝ}} = 26 - 62.333 - 74.666 = -110.999$$

Αρκούδα-Νυφίτσα (i=2, j=4)

$$D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{ΝΥΦΙΤΣΑ}}} - u_{\text{ΑΡΚΟΥΔΑ}} - u_{\text{ΝΥΦΙΤΣΑ}} = 34 - 62.333 - 72.833 = -101.166$$

Αρκούδα-Φώκια (i=2, j=5)

$$D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{ΦΩΚΙΑ}}} - u_{\text{ΑΡΚΟΥΔΑ}} - u_{\text{ΦΩΚΙΑ}} = 29 - 62.333 - 70.333 = -103.666$$

Αρκούδα-Θαλάσσιο Λιοντάρι (i=2, j=6)

$$D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{ΘΑΛΑΣΣΙΟ}\\ \text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΑΡΚΟΥΔΑ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\ \text{ΛΙΟΝΤΑΡΙ}}} = 33 - 62.333 - 69.833 = -99.166$$

Αρκούδα-Γάτα (i=2, j=7)

$$D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{ΓΑΤΑ}}} - u_{\text{ΑΡΚΟΥΔΑ}} - u_{\text{ΓΑΤΑ}} = 84 - 62.333 - 114.5 = -92.833$$

Αρκούδα-Πίθηκος (i=2, j=8)

$$D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{ΠΙΘΗΚΟΣ}}} - u_{\text{ΑΡΚΟΥΔΑ}} - u_{\text{ΠΙΘΗΚΟΣ}} = 136 - 62.333 - 168.333 = -94.666$$

Ρακούν-Νυφίτσα (i=3, j=4)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\ \text{ΝΥΦΙΤΣΑ}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\text{ΝΥΦΙΤΣΑ}} = 42 - 74.666 - 72.833 = -105.499$$

Ρακούν-Φώκια (i=3, j=5)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\ \text{ΦΩΚΙΑ}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\text{ΦΩΚΙΑ}} = 44 - 74.666 - 70.333 = -100.999$$

Ρακούν-Θαλάσσιο Λιοντάρι (i=3, j=6)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} = 44 - 74.666 - 69.833 = -100.499$$

Ρακούν-Γάτα (i=3, j=7)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\\text{ΓΑΤΑ}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\text{ΓΑΤΑ}} = 92 - 74.666 - 114.5 = -97.166$$

Ρακούν-Πίθηκος (i=3, j=8)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\\text{ΠΙΘΗΚΟΣ}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\text{ΠΙΘΗΚΟΣ}} = 152 - 74.666 - 168.333 = -90.999$$

Νυφίτσα-Φώκια (i=4, j=5)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\\text{ΦΩΚΙΑ}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{ΦΩΚΙΑ}} = 44 - 72.833 - 70.333 = -99.166$$

Νυφίτσα-Θαλάσσιο Λιοντάρι (i=4, j=6)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} = 38 - 72.833 - 69.833 = -104.666$$

Νυφίτσα-Γάτα (i=4, j=7)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\\text{ΓΑΤΑ}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{ΓΑΤΑ}} = 86 - 72.833 - 114.5 = -101.333$$

Νυφίτσα-Πίθηκος (i=4, j=8)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\\text{ΠΙΘΗΚΟΣ}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{ΠΙΘΗΚΟΣ}} = 142 - 72.833 - 168.333 = -99.166$$

Φώκια-Θαλάσσιο Λιοντάρι (i=5, j=6)

$$D_{\substack{\text{ΦΩΚΙΑ}\\\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΦΩΚΙΑ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} = 24 - 70.333 - 69.833 = -116.166$$

Φώκια-Γάτα (i=5, j=7)

$$D_{\substack{\text{ΦΩΚΙΑ}\\\text{ΓΑΤΑ}}} - u_{\text{ΦΩΚΙΑ}} - u_{\text{ΓΑΤΑ}} = 89 - 70.333 - 114.5 = -95.833$$

Φώκια-Πίθηκος (i=5, j=8)

$$D_{\substack{\text{ΦΩΚΙΑ}\\\text{ΠΙΘΗΚΟΣ}}} - u_{\text{ΦΩΚΙΑ}} - u_{\text{ΠΙΘΗΚΟΣ}} = 142 - 70.333 - 168.333 = -96.666$$

Θαλάσσιο Λιοντάρι-Γάτα (i=6, j=7)

$$D_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}\\\text{ΓΑΤΑ}}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΓΑΤΑ}} = 90 - 69.833 - 114.5 = -94.333$$

Θαλάσσιο Λιοντάρι-Πίθηκος (i=6, j=8)

$$D_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}\\\text{ΠΙΘΗΚΟΣ}}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΠΙΘΗΚΟΣ}} = 142 - 69.833 - 168.333 = -96.166$$

Γάτα-Πίθηκος (i=7, j=8)

$$D_{\substack{\Gamma\Alpha\Tau\Alpha\\ \Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma}} - u_{\Gamma\Alpha\Tau\Alpha} - u_{\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma} = 148 - 114.5 - 168.333 = -134.833$$

Αφού λοιπόν υπολογίστηκε η ζητούμενη παράσταση για όλα τα ζεύγη παρατηρούμε πως αυτή ελαχιστοποιείται (-134,833) για το ζεύγος Γάτα-Πίθηκος (i=7, j=8).

3) Τώρα τα i και j ενώνονται. Υπολογίζεται το μήκος του κλαδιού από το i ως το νέο κόμβο (v_i) και από το j ως το νέο κόμβο (v_j) ως εξής:

$$v_i = \frac{1}{2}D_{ij} + \frac{1}{2}(u_i - u_j) \quad \text{και} \quad v_j = \frac{1}{2}D_{ij} + \frac{1}{2}(u_j - u_i)$$

Άρα για τη Γάτα:

$$\text{Μήκος} = \frac{1}{2}D_{\substack{\Gamma\Alpha\Tau\Alpha\\ \Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma}} + \frac{1}{2}(u_{\Gamma\Alpha\Tau\Alpha} - u_{\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma}) = \frac{1}{2}\times 148 + \frac{1}{2}\times(114.5 - 168.333) = 47.0835$$

και για τον πίθηκο

$$\text{Μήκος} = \frac{1}{2}D_{\substack{\Gamma\Alpha\Tau\Alpha\\ \Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma}} + \frac{1}{2}(u_{\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma} - u_{\Gamma\Alpha\Tau\Alpha}) = \frac{1}{2}\times 148 + \frac{1}{2}\times(168.333 - 114.5) = 100.9165$$

4) Στη συνέχεια υπολογίζεται η απόσταση μεταξύ του νέου κόμβου (ij) και οποιουδήποτε άλλου στοιχείου σύμφωνα με τον τύπο $D_{(ij),k} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$.

$$D_{(\Gamma\Alpha\Tau\Alpha-\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma),\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} = \frac{D_{\Gamma\Sigma} + D_{\Pi\Sigma} - D_{\Gamma\Pi}}{2} = \frac{98 + 148 - 148}{2} = 49$$

$$D_{(\Gamma\Alpha\Tau\Alpha-\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma),\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha} = \frac{D_{\Gamma\Alpha} + D_{\Pi\Alpha} - D_{\Gamma\Pi}}{2} = \frac{84 + 136 - 148}{2} = 36$$

$$D_{(\Gamma\Alpha\Tau\Alpha-\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma),\Rho\Alpha\Kappa\Omicron\Upsilon\Nu} = \frac{D_{\Gamma\Rho} + D_{\Pi\Rho} - D_{\Gamma\Pi}}{2} = \frac{92 + 152 - 148}{2} = 48$$

$$D_{(\Gamma\Alpha\Tau\Alpha-\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma),\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = \frac{D_{\Gamma\Nu} + D_{\Pi\Nu} - D_{\Gamma\Pi}}{2} = \frac{86 + 142 - 148}{2} = 40$$

$$D_{(\Gamma\Alpha\Tau\Alpha-\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma),\Phi\Omega\Kappa\Iota\Alpha} = \frac{D_{\Gamma\Phi} + D_{\Pi\Phi} - D_{\Gamma\Pi}}{2} = \frac{89 + 142 - 148}{2} = 41.5$$

$$D_{(\Gamma\Alpha\Tau\Alpha-\Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma),\Theta\Alpha\Lambda.\Lambda\Iota\Omicron\Nu\Tau\Alpha\Rho\Iota} = \frac{D_{\Gamma\Theta\Lambda} + D_{\Pi\Theta\Lambda} - D_{\Gamma\Pi}}{2} = \frac{90 + 142 - 148}{2} = 42$$

5) Πλέον υπολογίστηκαν οι νέες αποστάσεις μεταξύ του στοιχείου (ΓΑΤΑ-ΠΙΘΗΚΟΣ) και όλων των λοιπών στοιχείων του πίνακα. Αφού λοιπόν συγχωνεύονται οι αντίστοιχες στήλες και γραμμές ο πίνακας έχει την παρακάτω μορφή:

	Σκύλος	Αρκούδα	Ρακούν	Νυφίτσα	Φώκια	Θαλάσσιο Λιοντάρι	Γάτα Πίθηκος
Σκύλος	0	32	48	51	50	48	49
Αρκούδα	32	0	26	34	29	33	36
Ρακούν	48	26	0	42	44	44	48
Νυφίτσα	51	34	42	0	44	38	40
Φώκια	50	29	44	44	0	24	41.5
Θαλάσσιο Λιοντάρι	48	33	44	38	24	0	42
Γάτα Πίθηκος	49	36	48	40	41.5	42	0

Πίνακας 4.9: Neighbor-Joining Βήμα 1

Εφόσον έχουν απομείνει περισσότερα των δύο στοιχεία στον πίνακα επιστρέφουμε στο πρώτο βήμα.

1)Υπολογίζουμε τις ποσότητες $u_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{n-2}$ για κάθε i του πίνακα (δηλαδή για κάθε είδος).

Είναι λοιπόν για n=7:

$$u_{\Sigma K Y \Lambda O \Sigma} = \frac{32+48+51+50+48+49}{5} = \frac{278}{5} = 55.6$$

$$u_{A P K O Y \Delta A} = \frac{32+26+34+29+33+36}{5} = \frac{190}{5} = 38$$

$$u_{P A K O Y N} = \frac{48+26+42+44+44+48}{5} = \frac{252}{5} = 50.4$$

$$u_{N Y \Phi I T \Sigma A} = \frac{51+34+42+44+38+40}{5} = \frac{249}{5} = 49.8$$

$$u_{\Phi \Omega K I A} = \frac{50+29+44+44+24+41,5}{5} = \frac{232.5}{5} = 46.5$$

$$u_{\substack{\Theta A \Lambda A \Sigma \Sigma I O \\ \Lambda I O N T A P I}} = \frac{48+33+44+38+24+42}{5} = \frac{229}{5} = 45.8$$

$$u_{\substack{\Gamma A T A \\ \Pi I \Theta H K O \Sigma}} = \frac{49+36+48+40+41.5+42}{5} = \frac{256.5}{5} = 51.3$$

2) Υπολογίζονται και πάλι οι παραστάσεις $D_{ij} - u_i - u_j$ για όλα τα i και j (με τα νέα τώρα δεδομένα) και επιλέγονται αυτά που την ελαχιστοποιούν.

Σκύλος-Αρκούδα (i=1, j=2)

$$D_{\substack{\Sigma K Y \Lambda O \Sigma \\ A P K O Y \Delta A}} - u_{\Sigma K Y \Lambda O \Sigma} - u_{A P K O Y \Delta A} = 32 - 55.6 - 38 = -61.6$$

Σκύλος-Ρακούν (i=1, j=3)

$$D_{\substack{\Sigma K Y \Lambda O \Sigma \\ P A K O Y N}} - u_{\Sigma K Y \Lambda O \Sigma} - u_{P A K O Y N} = 48 - 55.6 - 50.4 = -58$$

Σκύλος-Νυφίτσα (i=1, j=4)

$$D_{\substack{\Sigma K Y \Lambda O \Sigma \\ N Y \Phi I T \Sigma A}} - u_{\Sigma K Y \Lambda O \Sigma} - u_{N Y \Phi I T \Sigma A} = 51 - 55.6 - 49.8 = -54.4$$

Σκύλος-Φώκια (i=1, j=5)

$$D_{\substack{\Sigma K Y \Lambda O \Sigma \\ \Phi \Omega K I A}} - u_{\Sigma K Y \Lambda O \Sigma} - u_{\Phi \Omega K I A} = 50 - 55.6 - 46.5 = -52.1$$

Σκύλος-Θαλάσσιο Λιοντάρι (i=1, j=6)

$$D_{\substack{\Sigma K Y \Lambda O \Sigma \\ \Theta A \Lambda A \Sigma \Sigma I O \\ \Lambda I O N T A P I}} - u_{\Sigma K Y \Lambda O \Sigma} - u_{\substack{\Theta A \Lambda A \Sigma \Sigma I O \\ \Lambda I O N T A P I}} = 48 - 55.6 - 45.8 = -53.4$$

Σκύλος- Γ.Π (i=1, j=7)

$$D_{\substack{\Sigma K Y \Lambda O \Sigma \\ \Gamma . \Pi}} - u_{\Sigma K Y \Lambda O \Sigma} - u_{\Gamma . \Pi} = 49 - 55.6 - 51.3 = -57.9$$

Αρκούδα-Ρακούν (i=2, j=3)

$$D_{\substack{A P K O Y \Delta A \\ P A K O Y N}} - u_{A P K O Y \Delta A} - u_{P A K O Y N} = 26 - 38 - 50.4 = -62.4$$

Αρκούδα-Νυφίτσα (i=2, j=4)

$$D_{\substack{A P K O Y \Delta A \\ N Y \Phi I T \Sigma A}} - u_{A P K O Y \Delta A} - u_{N Y \Phi I T \Sigma A} = 34 - 38 - 49.8 = -53.8$$

Αρκούδα-Φώκια (i=2, j=5)

$$D_{\substack{A P K O Y \Delta A \\ \Phi \Omega K I A}} - u_{A P K O Y \Delta A} - u_{\Phi \Omega K I A} = 29 - 38 - 46.5 = -55.5$$

Αρκούδα-Θαλάσσιο Λιοντάρι (i=2, j=6)

$$D_{\substack{A P K O Y \Delta A \\ \Theta A \Lambda A \Sigma \Sigma I O \\ \Lambda I O N T A P I}} - u_{A P K O Y \Delta A} - u_{\substack{\Theta A \Lambda A \Sigma \Sigma I O \\ \Lambda I O N T A P I}} = 33 - 38 - 45.8 = -50.8$$

Αρκούδα-Γ.Π (i=2, j=7)

$$D_{\substack{A P K O Y \Delta A \\ \Gamma . \Pi}} - u_{A P K O Y \Delta A} - u_{\Gamma . \Pi} = 36 - 38 - 51.3 = -53.3$$

Ρακούν-Νυφίτσα (i=3, j=4)

$$D_{\substack{P A K O Y N \\ N Y \Phi I T \Sigma A}} - u_{P A K O Y N} - u_{N Y \Phi I T \Sigma A} = 42 - 50.4 - 49.8 = -58.2$$

Ρακούν-Φώκια (i=3, j=5)

$$D_{\substack{P A K O Y N \\ \Phi \Omega K I A}} - u_{P A K O Y N} - u_{\Phi \Omega K I A} = 44 - 50.4 - 46.5 = -52.9$$

Ρακούν-Θαλάσσιο Λιοντάρι (i=3, j=6)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} = 44 - 50.4 - 45.8 = -52.2$$

Ρακούν-Γ.Π (i=3, j=7)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\\text{Γ.Π}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\text{Γ.Π}} = 48 - 50.4 - 51.3 = -53.7$$

Νυφίτσα-Φώκια (i=4, j=5)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\\text{ΦΩΚΙΑ}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{ΦΩΚΙΑ}} = 44 - 49.8 - 46.5 = -52.3$$

Νυφίτσα-Θαλάσσιο Λιοντάρι (i=4, j=6)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} = 38 - 49.8 - 45.8 = -57.6$$

Νυφίτσα-Γ.Π (i=4, j=7)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\\text{Γ.Π}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{Γ.Π}} = 40 - 49.8 - 51.3 = -61.1$$

Φώκια-Θαλάσσιο Λιοντάρι (i=5, j=6)

$$D_{\substack{\text{ΦΩΚΙΑ}\\\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΦΩΚΙΑ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} = 24 - 46.5 - 45.8 = -68.3$$

Φώκια-Γ.Π (i=5, j=7)

$$D_{\substack{\text{ΦΩΚΙΑ}\\\text{Γ.Π}}} - u_{\text{ΦΩΚΙΑ}} - u_{\text{Γ.Π}} = 41.5 - 46.5 - 51.3 = -56.3$$

Θαλάσσιο Λιοντάρι-Γ.Π (i=6, j=7)

$$D_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}\\\text{Γ.Π}}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{Γ.Π}} = 42 - 45.8 - 51.3 = -55.1$$

Παρατηρούμε λοιπόν πως η παράσταση ελαχιστοποιείται (-68.3) για i=5, j=6 δηλαδή για το ζεύγος Φώκια-Θαλάσσιο Λιοντάρι.

3) Τώρα τα i και j ενώνονται. Υπολογίζεται το μήκος του κλαδιού από το i ως το νέο κόμβο (v_i) και από το j ως το νέο κόμβο (v_j) ως εξής:

$$v_i = \frac{1}{2}D_{ij} + \frac{1}{2}(u_i - u_j) \quad \text{και} \quad v_j = \frac{1}{2}D_{ij} + \frac{1}{2}(u_j - u_i)$$

Άρα για τη φώκια

$$\text{Μ}\textit{ήκος} = \frac{1}{2}D_{\substack{\text{ΦΩΚΙΑ}\\\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}} + \frac{1}{2}(u_{\text{ΦΩΚΙΑ}} - u_{\substack{\text{ΘΑΛΑΣΣΙΟ}\\\text{ΛΙΟΝΤΑΡΙ}}}) = \frac{1}{2}\times 24 + \frac{1}{2}\times(46.5 - 45.8) = 12.35$$

ενώ για το θαλάσσιο λιοντάρι

$$\text{Μ}\eta\kappa\text{ος} = \frac{1}{2} D_{\substack{\text{ΦΩΚΙΑ} \\ \text{ΘΑΛΑΣΣΙΟ} \\ \text{ΛΙΟΝΤΑΡΙ}}} + \frac{1}{2}(u_{\substack{\text{ΘΑΛΑΣΣΙΟ} \\ \text{ΛΙΟΝΤΑΡΙ}}} - u_{\text{ΦΩΚΙΑ}}) = \frac{1}{2} \times 24 + \frac{1}{2} \times (45.8 - 46.5) = 11.65$$

4) Στη συνέχεια υπολογίζεται η απόσταση μεταξύ του νέου κόμβου (ij) και οποιουδήποτε άλλου στοιχείου σύμφωνα με τον τύπο $D_{(ij),k} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$.

$$D_{(\text{ΦΩΚΙΑ}-\text{ΘΑΛ.ΛΙΟΝΤΑΡΙ}),\text{ΣΚΥΛΟΣ}} = \frac{D_{\text{ΦΣ}} + D_{\text{ΘΛΣ}} - D_{\text{ΦΘΛ}}}{2} = \frac{50 + 48 - 24}{2} = 37$$

$$D_{(\text{ΦΩΚΙΑ}-\text{ΘΑΛ.ΛΙΟΝΤΑΡΙ}),\text{ΑΡΚΟΥΔΑ}} = \frac{D_{\text{ΦΑ}} + D_{\text{ΘΛΑ}} - D_{\text{ΦΘΛ}}}{2} = \frac{29 + 33 - 24}{2} = 19$$

$$D_{(\text{ΦΩΚΙΑ}-\text{ΘΑΛ.ΛΙΟΝΤΑΡΙ}),\text{ΡΑΚΟΥΝ}} = \frac{D_{\text{ΦΡ}} + D_{\text{ΘΛΡ}} - D_{\text{ΦΘΛ}}}{2} = \frac{44 + 44 - 24}{2} = 32$$

$$D_{(\text{ΦΩΚΙΑ}-\text{ΘΑΛ.ΛΙΟΝΤΑΡΙ}),\text{ΝΥΦΙΤΣΑ}} = \frac{D_{\text{ΦΝ}} + D_{\text{ΘΛΝ}} - D_{\text{ΦΘΛ}}}{2} = \frac{44 + 38 - 24}{2} = 29$$

$$D_{(\text{ΦΩΚΙΑ}-\text{ΘΑΛ.ΛΙΟΝΤΑΡΙ}),\text{Γ.Π}} = \frac{D_{\text{ΦΓ.Π}} + D_{\text{ΘΛΓ.Π}} - D_{\text{ΦΘΛ}}}{2} = \frac{41.5 + 42 - 24}{2} = 29.75$$

5) Πλέον υπολογίστηκαν οι νέες αποστάσεις μεταξύ του στοιχείου (Φώκια-Θαλάσσιο Λιοντάρι) και όλων των λοιπών στοιχείων του πίνακα. Αφού λοιπόν συγχωνεύονται οι αντίστοιχες στήλες και γραμμές ο πίνακας έχει την παρακάτω μορφή:

	Σκύλος	Αρκούδα	Ρακούν	Νυφίτσα	Φώκια Θαλάσσιο Λιοντάρι	Γάτα Πίθηκος
Σκύλος	0	32	48	51	37	49
Αρκούδα	32	0	26	34	19	36
Ρακούν	48	26	0	42	32	48
Νυφίτσα	51	34	42	0	29	40
Φώκια Θαλάσσιο Λιοντάρι	37	19	32	29	0	29.75
Γάτα Πίθηκος	49	36	48	40	29.75	0

Πίνακας 4.10: Neighbor-Joining Βήμα 2

Εφόσον έχουν απομείνει περισσότερα των δύο στοιχεία στον πίνακα επιστρέφουμε στο πρώτο βήμα.

1)Υπολογίζουμε τις ποσότητες $u_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{n-2}$ για κάθε i του πίνακα (δηλαδή για κάθε είδος).

Είναι λοιπόν για n=6

$$u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} = \frac{32+48+51+37+49}{4} = \frac{217}{4} = 54.25$$

$$u_{\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha} = \frac{32+26+34+19+36}{4} = \frac{147}{4} = 36.75$$

$$u_{\Rho\Alpha\Kappa\Omicron\Upsilon\Nu} = \frac{48+26+42+32+48}{4} = \frac{196}{4} = 49$$

$$u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = \frac{51+34+42+29+40}{4} = \frac{196}{4} = 49$$

$$u_{\substack{\Phi\Omega\Kappa\Iota\Alpha \\ \Theta\Alpha\Lambda\Lambda\Alpha\Sigma\Sigma\Iota\Omicron \\ \Lambda\Iota\Omicron\Nu\Tau\Alpha\Rho\Iota}} = \frac{37+19+32+29+29,75}{4} = \frac{146.75}{4} = 36.6875$$

$$u_{\substack{\Gamma\Alpha\Tau\Alpha \\ \Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma}} = \frac{49+36+48+40+29.75}{4} = \frac{202.75}{4} = 50.6875$$

2) Υπολογίζονται και πάλι οι παραστάσεις $D_{ij} - u_i - u_j$ για όλα τα i και j (με τα νέα τώρα δεδομένα) και επιλέγονται αυτά που την ελαχιστοποιούν.

Σκύλος-Αρκούδα (i=1, j=2)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha} = 32 - 54.25 - 36.75 = -59$$

Σκύλος-Ρακούν (i=1, j=3)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Rho\Alpha\Kappa\Omicron\Upsilon\Nu}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Rho\Alpha\Kappa\Omicron\Upsilon\Nu} = 48 - 54.25 - 49 = -55.25$$

Σκύλος-Νυφίτσα (i=1, j=4)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = 51 - 54.25 - 49 = -52.25$$

Σκύλος-Φ.ΘΛ (i=1, j=5)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Phi.\Theta\Lambda}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Phi.\Theta\Lambda} = 37 - 54.25 - 36.6875 = -53.9375$$

Σκύλος- Γ.Π (i=1, j=6)

$$D_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Gamma.\Pi}} - u_{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma} - u_{\Gamma.\Pi} = 49 - 54.25 - 50.6875 = -55.9375$$

Αρκούδα-Ρακούν (i=2, j=3)

$$D_{\substack{\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha \\ \Rho\Alpha\Kappa\Omicron\Upsilon\Nu}} - u_{\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha} - u_{\Rho\Alpha\Kappa\Omicron\Upsilon\Nu} = 26 - 36.75 - 49 = -59.75$$

Αρκούδα-Νυφίτσα (i=2, j=4)

$$D_{\substack{\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha \\ \Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha}} - u_{\Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = 34 - 36.75 - 49 = -51.75$$

Αρκούδα-Φ.ΘΛ (i=2, j=5)

$$D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{Φ.ΘΛ}}} - u_{\text{ΑΡΚΟΥΔΑ}} - u_{\text{Φ.ΘΛ}} = 19 - 36.75 - 36.6875 = -54.4375$$

Αρκούδα-Γ.Π (i=2, j=6)

$$D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{Γ.Π}}} - u_{\text{ΑΡΚΟΥΔΑ}} - u_{\text{Γ.Π}} = 36 - 36.75 - 50.6875 = -51.4375$$

Ρακούν-Νυφίτσα (i=3, j=4)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\ \text{ΝΥΦΙΤΣΑ}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\text{ΝΥΦΙΤΣΑ}} = 42 - 49 - 49 = -56$$

Ρακούν-Φ.ΘΛ (i=3, j=5)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\ \text{Φ.ΘΛ}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\text{Φ.ΘΛ}} = 32 - 49 - 36.6875 = -53.6875$$

Ρακούν-Γ.Π (i=3, j=6)

$$D_{\substack{\text{ΡΑΚΟΥΝ}\\ \text{Γ.Π}}} - u_{\text{ΡΑΚΟΥΝ}} - u_{\text{Γ.Π}} = 48 - 49 - 50.6875 = -51.6875$$

Νυφίτσα-Φ.ΘΛ (i=4, j=5)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\ \text{Φ.ΘΛ}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{Φ.ΘΛ}} = 29 - 49 - 36.6875 = -56.6875$$

Νυφίτσα-Γ.Π (i=4, j=6)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\ \text{Γ.Π}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{Γ.Π}} = 40 - 49 - 50.6875 = -59.6875$$

Φ.ΘΛ-Γ.Π (i=5, j=6)

$$D_{\substack{\text{Φ.ΘΛ}\\ \text{Γ.Π}}} - u_{\text{Φ.ΘΛ}} - u_{\text{Γ.Π}} = 29.75 - 36.6875 - 50.6875 = -57.625$$

Παρατηρούμε λοιπόν πως η παράσταση ελαχιστοποιείται (-59.75) για i=2, j=3 δηλαδή για το ζεύγος Αρκούδα-Ρακούν.

3) Τώρα τα i και j ενώνονται. Υπολογίζεται το μήκος του κλαδιού από το i ως το νέο κόμβο (v_i) και από το j ως το νέο κόμβο (v_j) ως εξής:

$$v_i = \frac{1}{2}D_{ij} + \frac{1}{2}(u_i - u_j) \quad \text{και} \quad v_j = \frac{1}{2}D_{ij} + \frac{1}{2}(u_j - u_i)$$

Άρα για την αρκούδα

$$\text{Μήκος} = \frac{1}{2}D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{ΡΑΚΟΥΝ}}} + \frac{1}{2}(u_{\text{ΑΡΚΟΥΔΑ}} - u_{\text{ΡΑΚΟΥΝ}}) = \frac{1}{2}\times 26 + \frac{1}{2}\times(36.75 - 49) = 6.875$$

ενώ για το ρακούν

$$\text{Μήκος} = \frac{1}{2}D_{\substack{\text{ΑΡΚΟΥΔΑ}\\ \text{ΡΑΚΟΥΝ}}} + \frac{1}{2}(u_{\text{ΡΑΚΟΥΝ}} - u_{\text{ΑΡΚΟΥΔΑ}}) = \frac{1}{2}\times 26 + \frac{1}{2}\times(49 - 36.75) = 19.125$$

4) Στη συνέχεια υπολογίζεται η απόσταση μεταξύ του νέου κόμβου (ij) και οποιουδήποτε άλλου στοιχείου σύμφωνα με τον τύπο $D_{(ij),k} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$.

$$D_{(\text{ΑΡΚΟΥΔΑ-ΡΑΚΟΥΝ}),\text{ΣΚΥΛΟΣ}} = \frac{D_{\text{ΑΣ}} + D_{\text{ΡΣ}} - D_{\text{ΑΡ}}}{2} = \frac{32 + 48 - 26}{2} = 27$$

$$D_{(\text{ΑΡΚΟΥΔΑ-ΡΑΚΟΥΝ}),\text{ΝΥΦΙΤΣΑ}} = \frac{D_{\text{ΑΝ}} + D_{\text{ΡΝ}} - D_{\text{ΑΡ}}}{2} = \frac{34 + 42 - 26}{2} = 25$$

$$D_{(\text{ΑΡΚΟΥΔΑ-ΡΑΚΟΥΝ}),\text{ΦΘΛ}} = \frac{D_{\text{Α.ΦΘΛ}} + D_{\text{Ρ.ΦΘΛ}} - D_{\text{ΑΡ}}}{2} = \frac{19 + 32 - 26}{2} = 12.5$$

$$D_{(\text{ΑΡΚΟΥΔΑ-ΡΑΚΟΥΝ}),\text{ΓΠ}} = \frac{D_{\text{Α.ΓΠ}} + D_{\text{Ρ.ΓΠ}} - D_{\text{ΑΡ}}}{2} = \frac{36 + 48 - 26}{2} = 29$$

5) Πλέον υπολογίστηκαν οι νέες αποστάσεις μεταξύ του στοιχείου (Αρκούδα-Ρακούν) και όλων των λοιπών στοιχείων του πίνακα. Αφού λοιπόν συγχωνεύονται οι αντίστοιχες στήλες και γραμμές ο πίνακας έχει την παρακάτω μορφή:

	Σκύλος	Αρκούδα Ρακούν	Νυφίτσα	Φώκια Θαλάσσιο Λιοντάρι	Γάτα Πίθηκος
Σκύλος	0	27	51	37	49
Αρκούδα Ρακούν	27	0	25	12.5	29
Νυφίτσα	51	25	0	29	40
Φώκια Θαλάσσιο Λιοντάρι	37	12.5	29	0	29.75
Γάτα Πίθηκος	49	29	40	29.75	0

Πίνακας 4.11: Neighbor-Joining Βήμα 3

Εφόσον έχουν απομείνει περισσότερα των δύο στοιχεία στον πίνακα επιστρέφουμε στο πρώτο βήμα.

1)Υπολογίζουμε τις ποσότητες $u_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{n-2}$ για κάθε i του πίνακα (δηλαδή για κάθε είδος). Είναι λοιπόν για n=5:

$$u_{\text{ΣΚΥΛΟΣ}} = \frac{27 + 51 + 37 + 49}{3} = \frac{164}{3} = 54.667$$

$$u_{\substack{\text{ΑΡΚΟΥΔΑ} \\ \text{ΡΑΚΟΥΝ}}} = \frac{27 + 25 + 12.5 + 29}{3} = \frac{93.5}{3} = 31.1667$$

$$u_{\text{ΝΥΦΙΤΣΑ}} = \frac{51 + 25 + 29 + 40}{3} = \frac{145}{3} = 48.333$$

$$u_{\substack{\text{ΦΩΚΙΑ}\\ \text{ΘΑΛΑΣΣΙΟ}\\ \text{ΛΙΟΝΤΑΡΙ}}} = \frac{37 + 12.5 + 29 + 29.75}{3} = \frac{108.25}{3} = 36.0833$$

$$u_{\substack{\text{ΓΑΤΑ}\\ \text{ΠΙΘΗΚΟΣ}}} = \frac{49 + 29 + 40 + 29.75}{3} = \frac{147.75}{3} = 49.25$$

2) Υπολογίζονται και πάλι οι παραστάσεις $D_{ij} - u_i - u_j$ για όλα τα i και j (με τα νέα τώρα δεδομένα) και επιλέγονται αυτά που την ελαχιστοποιούν.

Σκύλος-Α.Ρ (i=1, j=2)

$$D_{\substack{\text{ΣΚΥΛΟΣ}\\ \text{Α.Ρ}}} - u_{\text{ΣΚΥΛΟΣ}} - u_{\text{Α.Ρ}} = 27 - 54.667 - 31.1667 = -58.8337$$

Σκύλος-Νυφίτσα (i=1, j=3)

$$D_{\substack{\text{ΣΚΥΛΟΣ}\\ \text{ΝΥΦΙΤΣΑ}}} - u_{\text{ΣΚΥΛΟΣ}} - u_{\text{ΝΥΦΙΤΣΑ}} = 51 - 54.25 - 49 = -52.25$$

Σκύλος-Φ.ΘΛ (i=1, j=4)

$$D_{\substack{\text{ΣΚΥΛΟΣ}\\ \text{Φ.ΘΛ}}} - u_{\text{ΣΚΥΛΟΣ}} - u_{\text{Φ.ΘΛ}} = 37 - 54.667 - 36.0833 = -53.7503$$

Σκύλος- Γ.Π (i=1, j=5)

$$D_{\substack{\text{ΣΚΥΛΟΣ}\\ \text{Γ.Π}}} - u_{\text{ΣΚΥΛΟΣ}} - u_{\text{Γ.Π}} = 49 - 54.667 - 49.25 = -54.917$$

Α.Ρ-Νυφίτσα (i=2, j=3)

$$D_{\substack{\text{Α.Ρ}\\ \text{ΝΥΦΙΤΣΑ}}} - u_{\text{Α.Ρ}} - u_{\text{ΝΥΦΙΤΣΑ}} = 25 - 31.1667 - 48.333 = -54.4997$$

Α.Ρ-Φ.ΘΛ (i=2, j=4)

$$D_{\substack{\text{Α.Ρ}\\ \text{Φ.ΘΛ}}} - u_{\text{Α.Ρ}} - u_{\text{Φ.ΘΛ}} = 12.5 - 31.1667 - 36.0833 = -54.75$$

Α.Ρ-Γ.Π (i=2, j=5)

$$D_{\substack{\text{Α.Ρ}\\ \text{Γ.Π}}} - u_{\text{Α.Ρ}} - u_{\text{Γ.Π}} = 29 - 31.1667 - 49.25 = -51.4167$$

Νυφίτσα-Φ.ΘΛ (i=3, j=4)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\ \text{Φ.ΘΛ}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{Φ.ΘΛ}} = 29 - 48.333 - 36.0833 = -55.4163$$

Νυφίτσα-Γ.Π (i=3, j=5)

$$D_{\substack{\text{ΝΥΦΙΤΣΑ}\\ \text{Γ.Π}}} - u_{\text{ΝΥΦΙΤΣΑ}} - u_{\text{Γ.Π}} = 40 - 48.333 - 49.25 = -57.583$$

Φ.ΘΛ-Γ.Π (i=4, j=5)

$$D_{\substack{\text{Φ.ΘΛ}\\ \text{Γ.Π}}} - u_{\text{Φ.ΘΛ}} - u_{\text{Γ.Π}} = 29.75 - 36.0833 - 49.25 = -55.5833$$

Παρατηρούμε λοιπόν πως η παράσταση ελαχιστοποιείται (-58.8337) για i=1, j=2 δηλαδή για το ζεύγος Σκύλος-Α.Ρ.

3) Τώρα τα i και j ενώνονται. Υπολογίζεται το μήκος του κλαδιού από το i ως το νέο κόμβο (v_i) και από το j ως το νέο κόμβο (v_j) ως εξής:

$$v_i = \frac{1}{2}D_{ij} + \frac{1}{2}(u_i - u_j) \quad \text{και} \quad v_j = \frac{1}{2}D_{ij} + \frac{1}{2}(u_j - u_i)$$

Άρα για το σκύλο

$$\text{Μήκος} = \frac{1}{2}D_{\substack{\Sigma\text{ΚΥΛΟΣ}\\ \text{Α.Ρ}}} + \frac{1}{2}(u_{\Sigma\text{ΚΥΛΟΣ}} - u_{\text{Α.Ρ}}) = \frac{1}{2}\times 27 + \frac{1}{2}\times(54.667 - 31.1667) = 25.25015$$

ενώ για το Α.Ρ

$$\text{Μήκος} = \frac{1}{2}D_{\substack{\Sigma\text{ΚΥΛΟΣ}\\ \text{Α.Ρ}}} + \frac{1}{2}(u_{\text{Α.Ρ}} - u_{\Sigma\text{ΚΥΛΟΣ}}) = \frac{1}{2}\times 27 + \frac{1}{2}\times(31.1667 - 54.667) = 1.74985$$

4) Στη συνέχεια υπολογίζεται η απόσταση μεταξύ του νέου κόμβου (ij) και οποιουδήποτε άλλου στοιχείου σύμφωνα με τον τύπο $D_{(ij),k} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$.

$$D_{(\Sigma.\text{ΑΡ}),\text{ΝΥΦΙΤΣΑ}} = \frac{D_{\Sigma.\text{Ν}} + D_{\text{ΑΡ.Ν}} - D_{\Sigma.\text{ΑΡ}}}{2} = \frac{51 + 25 - 27}{2} = 24.5$$

$$D_{(\Sigma.\text{ΑΡ}),\Phi\Theta\Lambda} = \frac{D_{\Sigma.\Phi\Theta\Lambda} + D_{\text{ΑΡ.}\Phi\Theta\Lambda} - D_{\Sigma.\text{ΑΡ}}}{2} = \frac{37 + 12.5 - 27}{2} = 11.25$$

$$D_{(\Sigma.\text{ΑΡ}),\Gamma\Pi} = \frac{D_{\Sigma.\Gamma\Pi} + D_{\text{ΑΡ.}\Gamma\Pi} - D_{\Sigma.\text{ΑΡ}}}{2} = \frac{49 + 29 - 27}{2} = 25.5$$

5) Πλέον υπολογίστηκαν οι νέες αποστάσεις μεταξύ του στοιχείου (Σκύλος-ΑΡ) και όλων των λοιπών στοιχείων του πίνακα. Αφού λοιπόν συγχωνεύονται οι αντίστοιχες στήλες και γραμμές ο πίνακας έχει την παρακάτω μορφή:

	Σκύλος Αρκ-Ρακ.	Νυφίτσα	Φώκια Θαλ.Λιον.	Γάτα Πίθηκος
Σκύλος Αρκ.-Ρακ.	0	24.5	11.25	25.5
Νυφίτσα	24.5	0	29	40
Φώκια Θαλ.Λιον.	11.25	29	0	29.75
Γάτα Πίθηκος	25.5	40	29.75	0

Πίνακας 4.12: Neighbor-Joining Βήμα 4

Εφόσον έχουν απομείνει περισσότερα των δύο στοιχεία στον πίνακα επιστρέφουμε στο πρώτο βήμα.

1)Υπολογίζουμε τις ποσότητες $u_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{n-2}$ για κάθε i του πίνακα (δηλαδή για κάθε είδος).

Είναι λοιπόν για n=4

$$u_{\substack{\Sigma\Kappa\Upsilon\Lambda\Omicron\Sigma \\ \Alpha\Rho\Kappa\Omicron\Upsilon\Delta\Alpha \\ \Rho\Alpha\Kappa\Omicron\Upsilon\Nu}} = \frac{24.5 + 11.25 + 25.5}{2} = \frac{61.25}{2} = 30.625$$

$$u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = \frac{24.5 + 29 + 40}{2} = \frac{93.5}{2} = 46.75$$

$$u_{\substack{\Phi\Omega\Kappa\Iota\Alpha \\ \Theta\Alpha\Lambda\Alpha\Sigma\Sigma\Iota\Omicron \\ \Lambda\Iota\Omicron\Nu\Tau\Alpha\Rho\Iota}} = \frac{11.25 + 29 + 29.75}{2} = \frac{70}{2} = 35$$

$$u_{\substack{\Gamma\Alpha\Tau\Alpha \\ \Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma}} = \frac{25.5 + 40 + 29.75}{2} = \frac{95.25}{2} = 47.625$$

2) Υπολογίζονται και πάλι οι παραστάσεις $D_{ij} - u_i - u_j$ για όλα τα i και j (με τα νέα τώρα δεδομένα) και επιλέγονται αυτά που την ελαχιστοποιούν.

Σ.ΑΡ-Νυφίτσα (i=1, j=2)

$$D_{\substack{\Sigma.\Alpha\Rho \\ \Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha}} - u_{\Sigma.\Alpha\Rho} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = 24.5 - 30.625 - 46.75 = -52.875$$

Σ.ΑΡ-Φ.ΘΛ (i=1, j=3)

$$D_{\substack{\Sigma.\Alpha\Rho \\ \Phi.\Theta\Lambda}} - u_{\Sigma.\Alpha\Rho} - u_{\Phi.\Theta\Lambda} = 11.25 - 30.625 - 35 = -54.375$$

Σ.ΑΡ-Γ.Π (i=1, j=4)

$$D_{\substack{\Sigma.\Alpha\Rho \\ \Gamma.\Pi}} - u_{\Sigma.\Alpha\Rho} - u_{\Gamma.\Pi} = 25.5 - 30.625 - 47.625 = -52.75$$

Νυφίτσα-Φ.ΘΛ (i=2, j=3)

$$D_{\substack{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha \\ \Phi.\Theta\Lambda}} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} - u_{\Phi.\Theta\Lambda} = 29 - 46.75 - 35 = -52.75$$

Νυφίτσα-Γ.Π (i=2, j=4)

$$D_{\substack{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha \\ \Gamma.\Pi}} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} - u_{\Gamma.\Pi} = 40 - 46.75 - 47.625 = -54.375$$

Φ.ΘΛ-Γ.Π (i=3, j=4)

$$D_{\substack{\Phi.\Theta\Lambda \\ \Gamma.\Pi}} - u_{\Phi.\Theta\Lambda} - u_{\Gamma.\Pi} = 29.75 - 35 - 47.625 = -52.875$$

Παρατηρούμε λοιπόν πως η παράσταση ελαχιστοποιείται (-54.375) για i=1, j=3 δηλαδή για το ζεύγος Σ.ΑΡ-Φ.ΘΛ καθώς και για το ζεύγος Νυφίτσα-Γ.Π με i=2, j=4. Επιλέγουμε το

πρώτο εκ των δύο ζευγών και συνεχίζουμε κανονικά, ενώ αποδεικνύεται πως αν επιλέγαμε το δεύτερο, το τελικό αποτέλεσμα θα ήταν πρακτικά το ίδιο.

3) Τώρα τα i και j ενώνονται. Υπολογίζεται το μήκος του κλαδιού από το i ως το νέο κόμβο (v_i) και από το j ως το νέο κόμβο (v_j) ως εξής:

$$v_i = \frac{1}{2}D_{ij} + \frac{1}{2}(u_i - u_j) \quad \text{και} \quad v_j = \frac{1}{2}D_{ij} + \frac{1}{2}(u_j - u_i)$$

Άρα για το Σ.ΑΡ

$$\text{Μήκος} = \frac{1}{2}D_{\substack{\Sigma.\text{ΑΡ} \\ \Phi.\Theta\Lambda}} + \frac{1}{2}(u_{\Sigma.\text{ΑΡ}} - u_{\Phi.\Theta\Lambda}) = \frac{1}{2}\times 11.25 + \frac{1}{2}\times(30.625 - 35) = 3.4375$$

ενώ για το Φ.ΘΛ

$$\text{Μήκος} = \frac{1}{2}D_{\substack{\Sigma.\text{ΑΡ} \\ \Phi.\Theta\Lambda}} + \frac{1}{2}(u_{\Phi.\Theta\Lambda} - u_{\Sigma.\text{ΑΡ}}) = \frac{1}{2}\times 11.25 + \frac{1}{2}\times(35 - 30.625) = 7.8125$$

4) Στη συνέχεια υπολογίζεται η απόσταση μεταξύ του νέου κόμβου (ij) και οποιουδήποτε άλλου στοιχείου σύμφωνα με τον τύπο $D_{(ij),k} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$.

$$D_{(\Sigma\text{ΑΡ.}\Phi\Theta\Lambda),\text{ΝΥΦΙΤΣΑ}} = \frac{D_{\Sigma\text{ΑΡ.Ν}} + D_{\Phi\Theta\Lambda\text{.Ν}} - D_{\Sigma\text{ΑΡ.}\Phi\Theta\Lambda}}{2} = \frac{24.5 + 29 - 11.25}{2} = 21.125$$

$$D_{(\Sigma\text{ΑΡ.}\Phi\Theta\Lambda),\Gamma.\Pi} = \frac{D_{\Sigma\text{ΑΡ.}\Gamma\Pi} + D_{\Phi\Theta\Lambda\text{.}\Gamma\Pi} - D_{\Sigma\text{ΑΡ.}\Phi\Theta\Lambda}}{2} = \frac{25.5 + 29.75 - 11.25}{2} = 22$$

5) Πλέον υπολογίστηκαν οι νέες αποστάσεις μεταξύ του στοιχείου (Σ.ΑΡ-Φ.ΘΛ) και όλων των λοιπών στοιχείων του πίνακα. Αφού λοιπόν συγχωνεύονται οι αντίστοιχες στήλες και γραμμές ο πίνακας έχει την παρακάτω μορφή:

	Σ.ΑΡ Φ.ΘΛ	Νυφίτσα	Γάτα Πίθηκος
Σ.ΑΡ Φ.ΘΛ	0	21.125	22
Νυφίτσα	21.125	0	40
Γάτα Πίθηκος	22	40	0

Πίνακας 4.13: Neighbor-Joining Βήμα 5

Εφόσον έχουν απομείνει περισσότερα των δύο στοιχεία στον πίνακα επιστρέφουμε στο πρώτο βήμα.

1)Υπολογίζουμε τις ποσότητες $u_i = \sum_{j \neq i}^{n} \frac{D_{ij}}{n-2}$ για κάθε i του πίνακα (δηλαδή για κάθε είδος).

Είναι λοιπόν για n=3

$$u_{\substack{\Sigma.\Alpha\Rho \\ \Phi.\Theta\Lambda}} = \frac{21.125 + 22}{1} = \frac{43.125}{1} = 43.125$$

$$u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = \frac{21.125 + 40}{1} = \frac{61.125}{1} = 61.125$$

$$u_{\substack{\Gamma\Alpha\Tau\Alpha \\ \Pi\Iota\Theta\Eta\Kappa\Omicron\Sigma}} = \frac{22 + 40}{1} = \frac{62}{1} = 62$$

2) Υπολογίζονται και πάλι οι παραστάσεις $D_{ij} - u_i - u_j$, για όλα τα i και j (με τα νέα τώρα δεδομένα) και επιλέγονται αυτά που την ελαχιστοποιούν.

ΣΑΡ.ΦΘΛ-Νυφίτσα (i=1, j=2)

$$D_{\substack{\Sigma\Alpha\Rho.\Phi\Theta\Lambda \\ \Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha}} - u_{\Sigma\Alpha\Rho.\Phi\Theta\Lambda} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} = 21.125 - 43.125 - 61.125 = -83.125$$

ΣΑΡ.ΦΘΛ-Γ.Π (i=1, j=3)

$$D_{\substack{\Sigma\Alpha\Rho.\Phi\Theta\Lambda \\ \Gamma.\Pi}} - u_{\Sigma\Alpha\Rho.\Phi\Theta\Lambda} - u_{\Gamma.\Pi} = 22 - 43.125 - 62 = -83.125$$

Νυφίτσα-Γ.Π (i=2, j=3)

$$D_{\substack{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha \\ \Gamma.\Pi}} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} - u_{\Gamma.\Pi} = 40 - 61.125 - 62 = -83.125$$

Παρατηρούμε λοιπόν πως η παράσταση ελαχιστοποιείται (-83.125) για όλα τα i και j, εκ των οποίων επιλέγω τα i=1 και j=2 δηλαδή το ζεύγος ΣΑΡ.ΦΘΛ-Νυφίτσα.

3) Τώρα τα i και j ενώνονται. Υπολογίζεται το μήκος του κλαδιού από το i ως το νέο κόμβο (v_i) και από το j ως το νέο κόμβο (v_j) ως εξής:

$$v_i = \frac{1}{2} D_{ij} + \frac{1}{2}(u_i - u_j) \quad \text{και} \quad v_j = \frac{1}{2} D_{ij} + \frac{1}{2}(u_j - u_i)$$

Άρα για το ΣΑΡ.ΦΘΛ

$$\text{Μήκος} = \frac{1}{2} D_{\substack{\Sigma\Alpha\Rho.\Phi\Theta\Lambda \\ \Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha}} + \frac{1}{2}(u_{\Sigma\Alpha\Rho.\Phi\Theta\Lambda} - u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha}) = \frac{1}{2} \times 21.125 + \frac{1}{2} \times (43.125 - 61.125) = 1.5625$$

ενώ για τη Νυφίτσα

$$\text{Μήκος} = \frac{1}{2} D_{\substack{\Sigma\Alpha\Rho.\Phi\Theta\Lambda \\ \Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha}} + \frac{1}{2}(u_{\Nu\Upsilon\Phi\Iota\Tau\Sigma\Alpha} - u_{\Sigma\Alpha\Rho.\Phi\Theta\Lambda}) = \frac{1}{2} \times 21.125 + \frac{1}{2} \times (61.125 - 43.125) = 19.5625$$

4) Στη συνέχεια υπολογίζεται η απόσταση μεταξύ του νέου κόμβου (ij) και οποιουδήποτε άλλου στοιχείου σύμφωνα με τον τύπο $D_{(ij),k} = \frac{D_{ik} + D_{jk} - D_{ij}}{2}$.

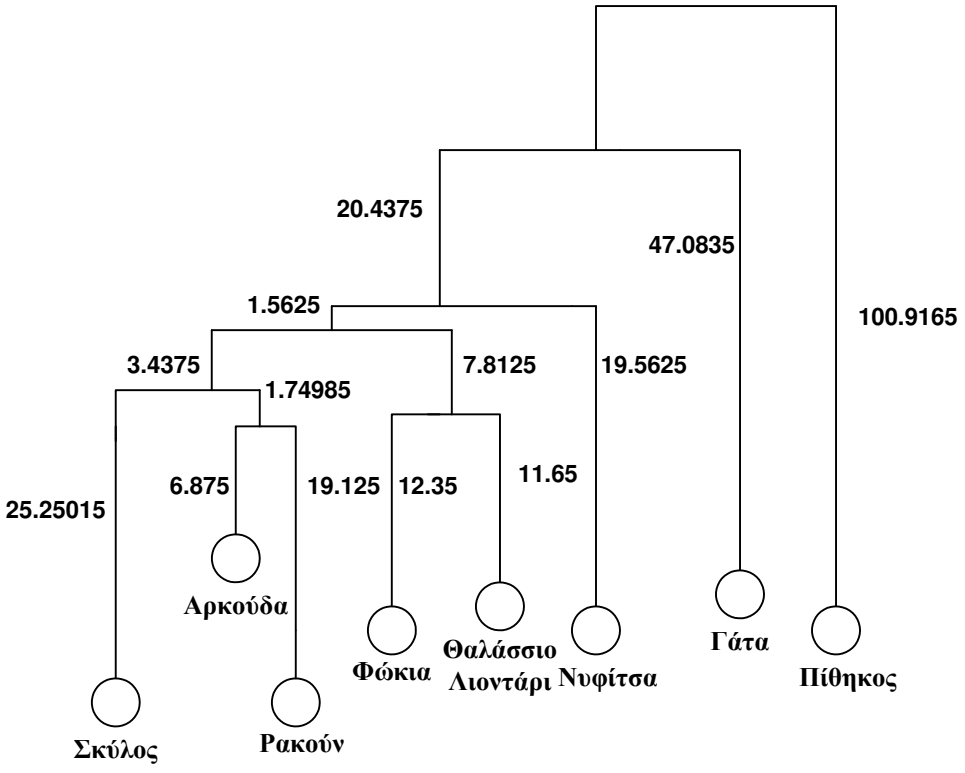
$$D_{(\Sigma\Lambda\Phi\Theta\Lambda.\Nu),\Gamma.\Pi} = \frac{D_{\Sigma\Lambda\Phi\Theta\Lambda.\Gamma\Pi} + D_{\Nu.\Gamma\Pi} - D_{\Sigma\Lambda\Phi\Theta\Lambda.\Nu}}{2} = \frac{22 + 40 - 21.125}{2} = 20.4375$$

5) Πλέον υπολογίστηκαν οι νέες αποστάσεις μεταξύ του στοιχείου (ΣΑΡΦΘΛ.Ν) και όλων των λοιπών στοιχείων του πίνακα, που έχει την παρακάτω μορφή:

	ΣΑΡΦΘΛ Νυφίτσα	Γάτα Πίθηκος
ΣΑΡΦΘΛ Νυφίτσα	0	20.4375
Γάτα Πίθηκος	20.4375	0

Πίνακας 4.14: Neighbor-Joining Βήμα 6

Εφόσον φτάσαμε σε δύο κόμβους, οι εναπομείναντες κόμβοι συνδέονται με ένα κλαδί μήκους D_{ij} και η εκτέλεση του αλγορίθμου τερματίζεται. Η τελική μορφή του δένδρου λοιπόν είναι:



Σχήμα 4.9: Neighbor-Joining Τελικό δένδρο

Στη συνέχεια θα περιγραφεί το PHYLIP, ένα από τα πλέον εύχρηστα και περιεκτικά πακέτα φυλογενετικής ανάλυσης που δημιουργήθηκε από τον Joseph Felsenstein στο πανεπιστήμιο της Ουάσινγκτον.

4.3 *Συστήματα Φυλογενετικής Ανάλυσης (Πακέτο PHYLIP)*

Με το πακέτο PHYLIP μπορούμε να κάνουμε πολλές εκ των διαθεσίμων στη βιβλιογραφία αναλύσεις, με χρήση διαφόρων μεθόδων και να χειριστούμε ποικίλους τύπους δεδομένων.

Υπάρχουν τρεις διαφορετικές μέθοδοι ανάλυσης δεδομένων ακολουθιών DNA ή αμινοξέων.

- Μέθοδος Αποστάσεων, στην οποία συνοψίζονται οι διαφορές μεταξύ ακολουθιών υπολογίζοντας ένα μέτρο αποστάσεων κατά ζεύγη, μεταξύ όλων των ευθυγραμμισμένων ακολουθιών. Τα προγράμματα αυτής της μεθόδου είναι σχεδιασμένα έτσι ώστε να χρησιμοποιούνται ακολουθιακά. Αρχικώς κατασκευάζεται ένας πίνακας αποστάσεων από τα προγράμματα Dnadist και Protdist με βάση την πολλαπλή ευθυγράμμιση ακολουθιών. Ο πίνακας αυτός στη συνέχεια μετατρέπεται σε ένα δένδρο χρησιμοποιώντας το πρόγραμμα Fitch, Kitsch ή Neighbor. Αναλυτικότερα τα προγράμματα αυτής της κατηγορίας (την οποία θα ερευνήσουμε με παράδειγμα) και οι αντίστοιχες λειτουργίες τους καταγράφονται στον Πίνακα 4.15.

Dnadist	Υπολογισμός πίνακα αποστάσεων DNA
Protdist	Υπολογισμός πίνακα αποστάσεων πρωτεϊνών
Fitch	Σχεδίαση δένδρου με τη μέθοδο Fitch-Margoliash χωρίς μοριακό ρολόι
Kitsch	Σχεδίαση δένδρου με τη μέθοδο Fitch-Margoliash με μοριακό ρολόι
Neighbor	Σχεδίαση δένδρου με τη μέθοδο Neighbor Joining και UPGMA

Πίνακας 4.15: Προγράμματα στη μέθοδο αποστάσεων

- Μέθοδος Φειδωλότητας (Parsimony) η οποία είναι μέθοδος χαρακτήρων (character based) κάτι που σημαίνει πως χειρίζεται κάθε περιοχή της πολλαπλής ευθυγράμμισης ανεξαρτήτως. Η έννοια της φειδωλότητας επικαλείται συχνά στην κατασκευή φυλογραμμάτων και βασίζεται στην υπόθεση ότι οι μεταλλάξεις συμβαίνουν σπάνια. Επομένως ο συνολικός αριθμός των μεταλλάξεων που υποθέτουμε πως συνέβησαν κατά τη διάρκεια των εξελικτικών βημάτων πρέπει να είναι ελάχιστος. Βεβαίως ως μετάλλαξη δεν θεωρούμε για παράδειγμα την αλλαγή ενός και μόνου νουκλεοτιδίου, γιατί τότε η κατασκευή δένδρων από ακολουθίες θα ήταν εξαιρετικά πολύπλοκη. Ένα παράδειγμα διαφωτίζει τη δυσκολία αυτή. Ας υποθέσουμε ένα δένδρο χωρίς ρίζα με δύο είδη ετικετών (labels). Οι ακολουθίες είναι ετικέτες στους κόμβους ενώ οι αριθμοί στα κλαδιά. Οι αριθμοί προδιαγράφουν τις μεταλλάξεις (αλλαγές συμβόλου σε συγκεκριμένη θέση της ακολουθίας) που συμβαίνουν μεταξύ ακολουθιών που είναι ετικέτες σε γειτονικούς κόμβους. Σε αυτήν την περίπτωση λοιπόν, θα είχαμε μεγάλο πλήθος ακολουθιών-κόμβων και αριθμών-κλαδιών και άρα μεγάλη δυσκολία στην κατασκευή του δένδρου.

Το πρόβλημα της κατασκευής ενός δένδρου με τη χρήση της φειδωλότητας τίθεται ως εξής: Με δεδομένες λίγες νουκλεοτιδικές ακολουθίες μικρού μήκους, να τοποθετηθούν αυτές σε κόμβους και φύλλα ενός δένδρου χωρίς ρίζα, έτσι ώστε ο συνολικός αριθμός των μεταλλάξεων να είναι ελάχιστος. Η απλούστερη λύση που κάποιος θα μπορούσε να φανταστεί είναι

να κατασκευαστούν όλα τα πιθανά δένδρα και στη συνέχεια να επιλεγεί αυτό με τον ελάχιστο αριθμό μεταλλάξεων. Αυτή η προσέγγιση βεβαίως είναι απαγορευτική για πολλές και μεγάλου μήκους ακολουθίες και αποτελεί ακόμα ένα παράδειγμα της συχνής εμφάνισης δύσκολων συνδυαστικών προβλημάτων στις Βιοεπιστήμες.

Τα προγράμματα που χρησιμοποιούνται εδώ απεικονίζονται στον Πίνακα 4.16.

Dnapars	Φειδωλότητα DNA
Protpars	Φειδωλότητα Πρωτεϊνών
Dnapenny	Φειδωλότητα DNA με χρήση αλγορίθμου επέκτασης και οριοθέτησης

Πίνακας 4.16: Προγράμματα στη μέθοδο φειδωλότητας

- Μέθοδος Μεγίστης Πιθανοφάνειας (Maximum Likelihood) που είναι ομοίως μέθοδος χαρακτήρων (character based). Τα προγράμματα που χρησιμοποιούνται εδώ είναι τα Dnaml, Proml, Dnamlk, Promlk και περιέχονται στον Πίνακα 4.17.

Dnaml	Μέθοδος Μέγιστης Πιθανοφάνειας DNA χωρίς μοριακό ρολόι
Proml	Μέθοδος Μέγιστης Πιθανοφάνειας Πρωτεϊνών χωρίς μοριακό ρολόι
Dnamlk	Μέθοδος Μέγιστης Πιθανοφάνειας DNA με μοριακό ρολόι
Promlk	Μέθοδος Μέγιστης Πιθανοφάνειας Πρωτεϊνών με μοριακό ρολόι

Πίνακας 4.17: Προγράμματα στη μέθοδο μέγιστης πιθανοφάνειας

Τα υπόλοιπα προγράμματα-εργαλεία που χρησιμοποιεί το PHYLIP (όπως π.χ. τα Drawgram και Drawtree) κυρίως χρησιμεύουν στη σχεδίαση δένδρων. Για παράδειγμα το πρόγραμμα Drawtree σχεδιάζει δένδρα χωρίς ρίζα, ενώ το πρόγραμμα Drawgram σχεδιάζει δένδρα με ρίζα. Το πακέτο PHYLIP καθώς και βοηθητικά έγγραφα διατίθενται δωρεάν στην ιστοσελίδα [Phy07], ενώ παρέχεται η δυνατότητα και διαδραστικής (interactive) εκτέλεσης μέσω της ιστοσελίδας [Pab07] (την οποία και χρησιμοποιήσαμε).

Προκειμένου να εκτελέσουμε τις έρευνες μας με το πακέτο PHYLIP απομονώσαμε τμήματα πρωτεϊνών από τη βάση δεδομένων Swiss-Prot και θέσαμε στο πρόγραμμα Protdist την παρακάτω είσοδο

>9KD_HUMAN

MEPPSPSPTHLSCIFFLLITVSPLEASSTRARVFPCLPLYAECPEQSLAQGKEKSHPGGGGERPGLAGQGEPD
HPAGARDGR

>CYB_SPHLE

MAIFIRKMHPLLKIMNHALVDLPAPSNISLWWNFGSLLGLCLIIQILTGLFLAMHYTADVSMAFSSVVHIC
RDVNYGWLIRNIHANGASLFFICVYLHIA

>BL1S2_MACFA

MAAAAEGVLATRRDESARDDAAVETAEEAKEPAEADITELCRDMFSKMATYLTGELTATSEDYKLLEN
MNKLTSLKYLEMKDIAINISRNLKDLNQKYAG

>A4_BOVIN

ISEVKMDAEFRHDSGYEVHHQKLVFFAEDVGSNKGAIIGLMVGGVVIATVIVITLVMLK

>HBA_AQUCH

MVLSANDKTNVKNVFTKISGHAEDYGAEALERMFTTYPPTKTYFPHFDLHHGSAQIKAHGKKVVGALIE
AVNHIDDMAGALSKLSDLHAQKLRVDPVNFK

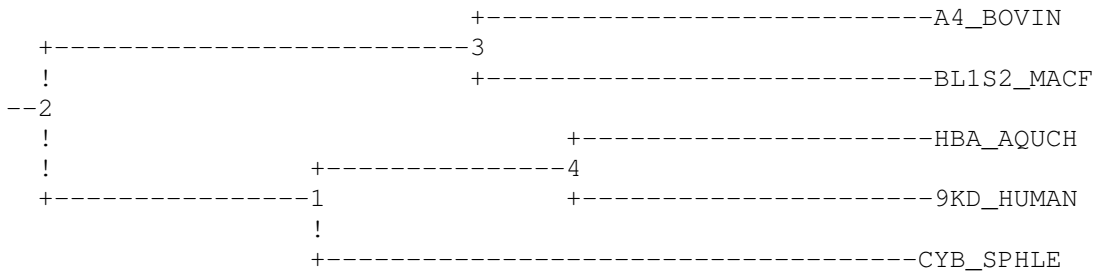
Ενδεικτικά αναφέρουμε κάποια στοιχεία για ορισμένες εκ των πρωτεϊνών που χρησιμοποιήσαμε. Η πρώτη πρωτεΐνη καταχωρημένη ως 9KD_HUMAN και με αριθμό πρόσβασης P13994, εμφανίζεται στον άνθρωπο, σχετίζεται με τον ιό της ασθένειας του Νιουκάστλ (Newcastle disease virus), ενσωματώθηκε στην πρωτεϊνική βάση δεδομένων UniProtKB /Swiss-Prot τον Ιανουάριο του 1990 και η τελευταία φορά που διαμορφώθηκε ήταν στις 23 Ιανουαρίου του 2007. Η πρωτεΐνη με όνομα CYB_SPHLE και με αριθμό πρόσβασης P34874 εντάχθηκε στην ίδια βάση δεδομένων το Φεβρουάριο του 1994, διαμορφώθηκε για τελευταία φορά τον Οκτώβριο του 2006 και εμφανίζεται στον σφυροκέφαλο καρχαρία (Sphyrna lewini-Hammerhead shark).

Το πρόγραμμα Protdist λοιπόν, αρχικώς σχημάτισε τον πίνακα αποστάσεων των πρωτεϊνών που είναι ο πίνακας:

```
9KD_HUMAN   0.0000 13.0034 40.0853 43.9885  5.9877
CYB_SPHLE  13.0034  0.0000 47.4792  8.7908  8.8688
BL1S2_MACF 40.0853 47.4792  0.0000  7.6224 39.4340
A4_BOVIN   43.9885  8.7908  7.6224  0.0000 39.5936
HBA_AQUCH   5.9877  8.8688 39.4340 39.5936  0.0000
```

Πίνακας 4.18: Πίνακας του Protdist

Στη συνέχεια εκτελέσαμε το πρόγραμμα Kitsch που σχεδίασε το παρακάτω δένδρο



Σχήμα 4.10: Δένδρο του Kitsch

όπου οι αποστάσεις μεταξύ των κόμβων (μήκη των κλαδιών) στο Σχήμα 4.10 απεικονίζονται στον Πίνακα 4.19.

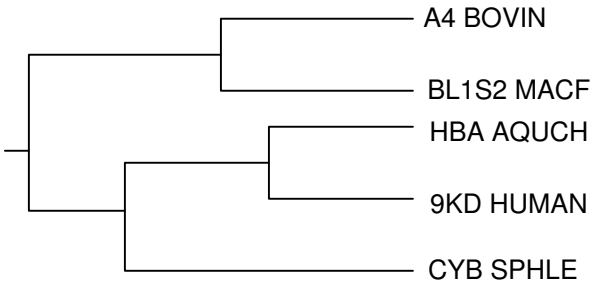
From Από	To Προς	Length Μήκος
3	A4_BOVIN	3.81120
2	3	3.56392
3	BL1S2_MACF	3.81120
4	HBA_AQUCH	2.99385
1	4	2.09689
4	9KD_HUMAN	2.99385
2	1	2.28438
1	CYB_SPHLE	5.09074

Πίνακας 4.19: Αποστάσεις μεταξύ των κόμβων του σχήματος 4.10

ενώ σε μια άλλη μορφή το δένδρο (και τα μήκη μεταξύ των κόμβων) παρουσιάζεται ως εξής:

```
((A4_BOVIN:3.81120,BL1S2_MACF:3.81120):3.56392,((HBA_AQUCH:2.99385,
9KD_HUMAN:2.99385):2.09689,CYB_SPHLE:5.09074):2.28438);
```

Στη συνέχεια εκτελέσαμε το πρόγραμμα drawgram που δημιούργησε το δένδρο του Σχήματος 4.11.



Σχήμα 4.11: Δένδρο του drawgram

4.4 Σύνοψη και Συμπεράσματα

Οι αλγόριθμοι κατασκευής φυλογενετικών δενδρών που εξετάστηκαν είναι οι πλεόν δημοφιλείς. Ο αλγόριθμος UPGMA είναι δημοφιλής λόγω της απλότητάς του, έχοντας ως μεγάλο μειονέκτημα την κατασκευή λάθος δένδρου, σε περίπτωση που τα αναπαριστώμενα είδη δεν εξελίσσονται με τον ίδιο ρυθμό, κάτι που ισχύει πολύ συχνά. Το πρόβλημα τότε λύνεται από τον αλγόριθμο Neighbor-Joining, ο οποίος λαμβάνει υπόψιν του τη διαφορά ρυθμού εξέλιξης των ειδών, κατασκευάζοντας έτσι το σωστό φυλογενετικό δένδρο.

Οι δύο αυτοί αλγόριθμοι εκτελούνται σε ίσο αριθμό βημάτων (N-2, όπου N το πλήθος των στοιχείων), με τον Neighbor-Joining όμως, να υπολογίζει πολύ περισσότερες παραμέτρους, ακριβώς για να συνυπολογίσει την ανισότητα των ρυθμών εξέλιξης των ειδών. Έτσι, είναι μεν λίγο βραδύτερος, αλλά κατασκευάζει το σωστό δένδρο. Ο Neighbor-Joining είναι δημοφιλέστερος του UPGMA, όχι λόγω της απλότητάς του (και οι δύο είναι σχετικά απλοί και πολύ φιλικοί), αλλά λόγω της ορθότητάς του. Τα χαρακτηριστικά τους συνοψίζονται στον Πίνακα 4.20.

Χαρακτηριστικά	UPGMA	Neighbor-Joining
Φιλικότητα	√√√	√√√
Ορθότητα	√	√√√√
Δημοφιλία	√√√	√√√√
Ταχύτητα	√√√√	√√√

Πίνακας 4.20: Χαρακτηριστικά μεθόδων αποστάσεων

Το πακέτο PHYLIP που επίσης μελετήθηκε, είναι μακράν το δημοφιλέστερο πακέτο φυλογενετικής ανάλυσης. Είναι ιδιαίτερα φιλικό, παρέχει πολλές επιλογές και δυνατότητες (όπως η επιλογή της μεθόδου ανάλυσης και η επιλογή των προγραμμάτων της μεθόδου) και είναι γενικά εύκολο στη χρήση.

5

Χρήση Σεναριακών Γλωσσών

5.1 *Εισαγωγή*

Η ανάγκη χρήσης σεναριακών γλωσσών είναι μεγάλη και φαίνεται από το παρακάτω παράδειγμα. Έστω ότι έχουμε στην κατοχή μας ένα αρχείο νουκλεοτιδικής ακολουθίας σε μορφή FASTA, το οποίο εκτός από την ίδια την ακολουθία μπορεί να περιέχει διάφορα σχόλια ή επικεφαλίδες που δεν μας ενδιαφέρουν. Μια πρώτη λοιπόν εργασία που θα ήταν απαραίτητη, είναι η απομόνωση της ακολουθίας και η απαλλαγή από οποιαδήποτε περιττή πληροφορία, όπως αριθμοί, σχόλια κλπ. Στη συνέχεια, υποθέτουμε πως σε αυτήν την ακολουθία θέλουμε να προσαρτήσουμε μία δεύτερη ακολουθία, προκειμένου να υπολογίσουμε τη συμπληρωματική ακολουθία της τελικής. Έπειτα έστω ότι θέλουμε να μεταγράψουμε την ακολουθία DNA σε ακολουθία RNA και να αναζητήσουμε σε αυτήν ένα συγκεκριμένο μοτίβο. Οι κανονικές εκφράσεις (regular expressions) χρησιμοποιούνται συχνά στον καθορισμό αυτών των μοτίβων. Τέλος υπάρχει η ανάγκη πρόσβασης μέσω του διαδικτύου σε προγράμματα χρήσιμα στις Βιοεπιστήμες όπως το BLAST.

Η αλυσιδωτή εκτέλεση των παραπάνω ενεργειών θα απαιτούσε από τον χρήστη-ερευνητή να διαδραματίζει ενεργό και κουραστικό χειρωνακτικό ρόλο, με το να ανοίγει αρχεία, να μελετά σύμβολο προς σύμβολο τις υπό έρευνα ακολουθίες, να πραγματοποιεί τις αναγκαίες διαμορφώσεις και να επιθεωρεί ή να απορρίπτει κάποια δεδομένα και ούτω καθεξής. Οι σεναριακές γλώσσες λοιπόν επιτρέπουν στους χρήστες να γράφουν προγράμματα που πραγματοποιούν αυτομάτως όλες τις παραπάνω ενέργειες.

Οι πλέον εύχρηστες σεναριακές γλώσσες στις βιοεπιστήμες είναι οι Perl και Python. Η Perl είναι παλαιότερη και έχει πολλά έτοιμα πακέτα διαθέσιμα για έρευνα ιστοσελίδων και εξαγωγή αποτελεσμάτων. Η Python είναι πιο πρόσφατη και σταδιακά κερδίζει σημαντικό έδαφος σε εφαρμογές βιοεπιστημών.

5.2 *Γλώσσα Perl*

Η γλώσσα Perl είναι μία διάσημη γλώσσα προγραμματισμού που έγινε διαθέσιμη στην πρώτη έκδοσή της το 1987 από τον Larry Wall. Χρησιμοποιείται ευρέως σε τομείς όπως οι Βιοεπιστήμες και ο δικτυακός προγραμματισμός. Έγινε δημοφιλής μεταξύ των βιολόγων επειδή είναι κατάλληλη για πολλές εργασίες των Βιοεπιστημών, διατίθεται δωρεάν και εκτελείται στα περισσότερα λειτουργικά συστήματα (Unix και Linux, Windows, Macintosh, VMS και άλλα). Όπως και κάθε άλλη γλώσσα προγραμματισμού η Perl έχει μία εφαρμογή μεταφραστή (γνωστή και ως μεταγλωττιστής) που μετατρέπει τα προγράμματα (ή αλλιώς σενάρια-scripts) σε εντολές που ο υπολογιστής μπορεί να εκτελέσει.

5.2.1 *Εισαγωγή στη Γλώσσα Perl*

Η Perl έχει πολλά πλεονεκτήματα, σημαντικότερα των οποίων είναι η ευκολία στον προγραμματισμό, η συμβατότητα με τα περισσότερα λειτουργικά συστήματα και η ικανότητα μεταφοράς προγραμμάτων μεταξύ λειτουργικών συστημάτων, η ταχύτητα εκτέλεσης και η δυνατότητα εύκολης συντήρησης και τροποποίησης προγραμμάτων.

Διαθέσιμη είναι επίσης μία συλλογή κώδικα της Perl που περιέχει πληθώρα χρήσιμων συναρτήσεων που σχετίζονται με τις βιοεπιστήμες και ονομάζεται Bioperl. Στη συνέχεια θα παραθέσουμε παραδείγματα και παρατηρήσεις για τη χρήση της γλώσσας Perl επικεντρωμένοι βέβαια κυρίως σε χειρισμό δεδομένων σχετικών με τις Βιοεπιστήμες.

- Τα σύμβολα που χρησιμοποιούνται στα παραδείγματα είναι οι γνωστές βάσεις νουκλεοτιδίων και τα αμινοξέα. Υπενθυμίζουμε πως οι βάσεις νουκλεοτιδίων είναι οι αδενίνη (A), κυτοσίνη (C), γουανίνη (G), θυμίνη (T) και ουρακίλη (U) ενώ τα αμινοξέα είναι 20 και συνθέτουν τις πρωτεΐνες.
- Η χρήση του συμβόλου # (δίεση) στην αρχή κάθε γραμμής σηματοδοτεί την εισαγωγή σχολίων. (Εξαίρεση αποτελεί η χρήση του συμβόλου # στην πρώτη γραμμή πολλών προγραμμάτων της Perl που μπορεί να είναι ως εξής: #!/usr/bin/perl;).
- Η χρήση του συμβόλου $ σημαίνει μεταβλητή (variable) στη γλώσσα Perl. Για παράδειγμα η $DNA είναι μεταβλητή με όνομα $DNA. Παρατηρούμε πως το όνομα των μεταβλητών στην Perl εμπεριέχει το σύμβολο του δολαρίου ($). Το σύμβολο αυτό μπορεί να ακολουθείται από γράμματα του αλφαβήτου (μικρά ή κεφαλαία), από ψηφία και από τον χαρακτήρα υπογράμμισης underscore (_). Ο μόνος περιορισμός που υπάρχει είναι ότι ο πρώτος χαρακτήρας δεν μπορεί να είναι ψηφίο. Η μεταβλητή αυτή μάλιστα είναι

μονόμετρη (scalar) κάτι που σημαίνει πως αντιστοιχεί σε ένα μόνο αντικείμενο δεδομένων. Αυτό το είδος μεταβλητών χρησιμοποιείται κυρίως για δεδομένα όπως συμβολοσειρές και αριθμούς των διαφόρων μορφών (όπως 3, 9,123 ή 3,5E10).

- Η ανάθεση τιμής σε μία συγκεκριμένη μεταβλητή γίνεται με το σύμβολο της ισότητας = (γνωστό και ως τελεστής ανάθεσης). Αμέσως μετά από μία εντολή ανάθεσης η μεταβλητή μπορεί πλέον να χρησιμοποιηθεί με τη νέα της τιμή.
- Οι συμβολοσειρές (strings) αρχίζουν και τερματίζονται είτε με μονά είτε με διπλά εισαγωγικά (quotes).
- Οι εντολές τερματίζονται με το σύμβολο ; (ελληνικό ερωτηματικό ή αλλιώς semicolon).

5.2.2 *Χειρισμός Ακολουθιών*

Το παράδειγμα 5.2.1 είναι ένα απλό πρόγραμμα που αποθηκεύει μία ακολουθία DNA σε μία μεταβλητή και στη συνέχεια τυπώνει αυτήν την ακολουθία στην οθόνη.

```
#!/usr/bin/perl –w
# Αποθήκευση ακολουθίας DNA και εμφάνιση στην οθόνη
# Αρχικώς αποθηκεύεται η ακολουθία σε μεταβλητή με όνομα $DNA
$DNA = 'AGTGCGATGCATGCAGTCGATCCGATCAGGCATGCTA';
# Στην συνέχεια τυπώνεται η ακολουθία DNA στην οθόνη
print $DNA;
# Τέλος ζητούμε από το πρόγραμμα να εξέλθει
exit;
```

Παράδειγμα 5.2.1

Η πρώτη γραμμή του προγράμματος 5.2.1 είναι ειδική και χρησιμοποιείται για να δείξει στον υπολογιστή που εκτελεί Unix και Linux ότι πρόκειται για πρόγραμμα της Perl. Σε εκτέλεση σε Windows ή Macintosh δεν απαιτείται η ειδική γραμμή, αλλά επειδή χρησιμοποιείται συχνά την καταγράφουμε. Το τέλος της γραμμής αυτής (-w) σημαίνει τη χρήση προειδοποιήσεων και έχει ως αποτέλεσμα την εκτύπωση μηνυμάτων στην περίπτωση λάθους(error). Το γράμμα w συμβολίζει τη λέξη warnings (προειδοποιήσεις).

- Η εντολή print χειρίζεται scalar μεταβλητές τυπώνοντας την τιμή τους. Στο πρόγραμμα 5.2.1 λοιπόν τυπώνεται το string που η μεταβλητή περιέχει (AGTGCGATGCATGCAGTCGATCCGATCAGGCATGCTA).
- Η εντολή exit λέει στον υπολογιστή να εξέλθει του προγράμματος. Στην Perl η εντολή αυτή δεν απαιτείται στο τέλος κάθε προγράμματος εφόσον τα προγράμματα εξέρχονται αυτομάτως όταν φτάσουν εκεί. Παρόλα αυτά δε βλάπτει η προσθήκη αυτής της εντολής σε αυτό το σημείο, καθώς με αυτήν γίνεται και εμφανές το τέρμα ενός προγράμματος.

Το παράδειγμα 5.2.2 αποτελεί μία απλή τροποποίηση του προγράμματος 5.2.1 και η λειτουργία του είναι η αλυσιδωτή σύνδεση (concatenation) δύο τμημάτων DNA (διαδικασία αρκετά κοινότυπη στη βιολογία). Με τον όρο αλυσιδωτή σύνδεση εννοούμε την προσκόλληση ενός αντικειμένου στο τέλος ενός άλλου.

```
#!/usr/bin/perl -w
# Αλυσιδωτή σύνδεση τμημάτων DNA
# Αποθήκευσε δύο τμήματα DNA σε δύο μεταβλητές  $DNA1 και $DNA2
$DNA1 = 'AGTGCGATGCATGCAGTCGATCCGATCAGGCATGCTA';
$DNA2 = 'ATAGTGCCGTGAGAGTGATGTAGTA';
# Εμφάνισε τις αρχικές ακολουθίες DNA στην οθόνη
print " DNA fragments:\n\n";
print $DNA1, "\n";
print $DNA2, "\n\n";
# Συνέδεσε αλυσιδωτά τα τμήματα DNA σε μία τρίτη μεταβλητή και εμφάνισέ τα
# στην οθόνη
# Χρησιμοποιώντας παρεμβολή string
$DNA3 = "$DNA1$DNA2";
print "Το τελικό DNA είναι (εκδοχή 1):\n\n";
print "$DNA3\n\n";
# Εναλλακτικά με χρήση της τελείας ( . )
$DNA3 = $DNA1 . $DNA2;
print " Το τελικό DNA είναι (εκδοχή 2):\n\n";
print "$DNA3\n\n";
# Χωρίς χρήση τρίτης μεταβλητής
print "Το τελικό DNA είναι (εκδοχή 3):\n\n";
print $DNA1, $DNA2, "\n";
exit;
```

Παράδειγμα 5.2.2

Η έξοδος (output) του προγράμματος αυτού είναι η παρακάτω:

```
DNA fragments

AGTGCGATGCATGCAGTCGATCCGATCAGGCATGCTA
ATAGTGCCGTGAGAGTGATGTAGTA
```

Το τελικό DNA είναι (εκδοχή 1):

AGTGCGATGCATGCAGTCGATCCGATCAGGCATGCTAATAGTGCCGTGAGAGTGATGTAGTA

Το τελικό DNA είναι (εκδοχή 2):

AGTGCGATGCATGCAGTCGATCCGATCAGGCATGCTAATAGTGCCGTGAGAGTGATGTAGTA

Το τελικό DNA είναι (εκδοχή 3):

AGTGCGATGCATGCAGTCGATCCGATCAGGCATGCTAATAGTGCCGTGAGAGTGATGTAGTA

Στο πρόγραμμα 5.2.2 παρατηρούμε τη χρήση της παράστασης "\n" και "\n\n" που προκαλούν μετακίνηση του κέρσορα κατά μία και δύο γραμμές κάτω αντιστοίχως. Η εντολή print χρησιμοποιεί σε αυτήν την περίπτωση διπλά quotes ακριβώς επειδή θέλουμε τη μετακίνηση του κέρσορα. Αν για παράδειγμα αντί για την εντολή

print "Το τελικό DNA είναι (εκδοχή 1):\n\n";

είχαμε την εντολή

print 'Το τελικό DNA είναι (εκδοχή 1):\n\n';

θα είχαμε ως αποτέλεσμα την εμφάνιση στην οθόνη της παρακάτω γραμμής

Το τελικό DNA είναι (εκδοχή 1):\n\n

και όχι την εμφάνιση της γραμμής

Το τελικό DNA είναι (εκδοχή 1):

και στη συνέχεια την επιθυμητή μετακίνηση του κέρσορα κατά δύο γραμμές κάτω.

Παρατηρούμε επίσης στο πρόγραμμα τους τρεις τρόπους με τους οποίους γίνεται η αλυσιδωτή σύνδεση δύο ακολουθιών (με παρεμβολή των δύο strings, με χρήση της τελείας και με τη χρήση του κόμματος).

Το παράδειγμα 5.2.3 μεταγράφει μία ακολουθία DNA σε RNA (Transcription-Μεταγραφή DNA σε RNA). Ως γνωστόν σε αυτήν τη βιολογική διαδικασία όλες οι βάσεις θυμίνης (T) μετατρέπονται σε βάσεις ουρακίλης (U).

```
#!/usr/bin/perl -w
#Μεταγραφή DNA σε RNA
#Η ακολουθία DNA είναι
$DNA = 'ACGGGAGGACGGGAAAATTACTACGGCATTAGC';
```

```
# Εμφάνισε το DNA στην οθόνη
print "Η αρχική ακολουθία DNA είναι:\n\n";
print "$DNA\n\n";
# Μετέγραψε το DNA σε RNA αντικαθιστώντας τις βάσεις T με βάσεις U
$RNA = $DNA;
$RNA =~ s/T/U/g;
# Εμφάνισε το RNA στην οθόνη
print "Το αποτέλεσμα της μεταγραφής του DNA σε RNA είναι:\n\n";
print "$RNA\n";
exit;
```

Παράδειγμα 5.2.3

Η έξοδος (output) του παραδείγματος 5.2.3 είναι:

Η αρχική ακολουθία DNA είναι

ACGGGAGGACGGGAAAATTACTACGGCATTAGC

Το αποτέλεσμα της μεταγραφής του DNA σε RNA είναι

ACGGGAGGACGGGAAAAUUACUACGGCAUUAGC

Στο πρόγραμμα 5.2.3 εμφανίζεται η εντολή $RNA =~ s/T/U/g; . Σε αυτήν την εντολή παρατηρούμε τον συνδετικό τελεστή (=~) καθώς και την εντολή αντικατάστασης (substitute command) s/T/U/g. Ο τελεστής (=~) χρησιμοποιείται σε περιπτώσεις μεταβλητών που περιέχουν strings και σημαίνει: «*Εφάρμοσε την λειτουργία που βρίσκεται στα δεξιά του τελεστή στο string της μεταβλητής που βρίσκεται αριστερά του*». Η εντολή αντικατάστασης αποτελείται από τμήματα που μεταξύ τους χωρίζονται με την πλάγια γραμμή (/). Αρχικώς το γράμμα s δείχνει πως πρόκειται για αντικατάσταση (substitution). Μετά την πρώτη πλάγια γραμμή / υπάρχει το T που αναπαριστά το στοιχείο στο string που θα αντικατασταθεί. Μετά τη δεύτερη γραμμή / υπάρχει το U το οποίο είναι αυτό που θα αντικαταστήσει το T. Τέλος μετά την τρίτη πλάγια γραμμή υπάρχει το γράμμα g που σημαίνει πως η αντικατάσταση θα γίνει σε όλο το μήκος του string, δηλαδή πως θα γίνει καθολικά (το γράμμα g προέρχεται από το global που σημαίνει ολικός). Η εντολή αυτή αντικατάστασης είναι ένα ενδεικτικό παράδειγμα της χρήσης των κανονικών εκφράσεων, οι οποίες κυρίως χρησιμεύουν στη διαχείριση κειμένου (text manipulation).

5.2.3 *Κανονικές Εκφράσεις (Regular Expressions)*

Το παράδειγμα 5.2.4 με δεδομένη μια ακολουθία νουκλεοτιδίων υπολογίζει την αντίστροφη συμπληρωματική της. Όπως είναι γνωστό στη διπλή έλικα του DNA υπάρχουν δύο νουκλεοτιδικές ακολουθίες οι οποίες είναι αντίστροφες και συμπληρωματικές. Με τον όρο συμπληρωματικές εννοούμε πως απέναντι από τα νουκλεοτίδια που περιέχουν τις αζωτούχες βάσεις αδενίνη, θυμίνη, γουανίνη, κυτοσίνη τοποθετούνται νουκλεοτίδια που φέρουν, αντίστοιχα, τις βάσεις θυμίνη, αδενίνη, κυτοσίνη, γουανίνη. Με τον όρο αντίστροφες εννοούμε πως η αρχή της μίας βρίσκεται απέναντι από το τέλος της άλλης. Ουσιαστικά λοιπόν προκειμένου να υπολογιστεί η αντίστροφη συμπληρωματική ακολουθία, πρέπει να αντιστραφεί η δοσμένη (η αρχή της να πάει στο τέλος) και στη συνέχεια να γίνουν οι αντικαταστάσεις των συμπληρωματικών βάσεων. Το ίδιο αποτέλεσμα θα έχουμε και αν πρώτα αντικαταστήσουμε τις βάσεις και στη συνέχεια αντιστρέψουμε τα άκρα. Ο υπολογισμός αυτός είναι πολύ συχνός και αποτελεί σημαντικό τμήμα πολλών εφαρμογών των Βιοεπιστημών.

```
#!/usr/bin/perl -w
# Υπολογισμός της αντίστροφης συμπληρωματικής ακολουθίας DNA
# Η ακολουθία DNA είναι
$DNA = 'ACGGGAGGACGGGAAAATTACTACGGCATTAGC';
# Εμφάνισε  το DNA στην οθόνη
print "Η αρχική ακολουθία DNA είναι:\n\n";
print "$DNA\n\n";
# Υπολόγισε την αντίστροφη συμπληρωματική ακολουθία
# Λάθος Μέθοδος
# Αρχικά  αντέστρεψε το αρχικό DNA και αποθήκευσέ το στη μεταβλητή $revcom
$revcom = reverse $DNA;
# Εν συνεχεία αντικατέστησε τις βάσεις με τις συμπληρωματικές τους
# A->T, T->A, G->C, C->G
$revcom =~ s/A/T/g;
$revcom =~ s/T/A/g;
$revcom =~ s/G/C/g;
$revcom =~ s/C/G/g;
# Εμφάνισε την αντίστροφη συμπληρωματική ακολουθία DNA
print "Η αντίστροφη συμπληρωματική ακολουθία DNA είναι:\n\n";
```

```
print "$revcom\n";
print "\nΛάθος αλγόριθμος\n";
print "Δεύτερη Προσπάθεια - Σωστή Μέθοδος\n\n";
# Αντέστρεψε το αρχικό DNA και αποθήκευσέ το στη μεταβλητή $revcom
$revcom = reverse $DNA;
$revcom =~ tr/ACGTacgt/TGCAtgca/;
print "Η αντίστροφη συμπληρωματική ακολουθία DNA είναι:\n\n";
print "$revcom\n";
print "\nΕπιτυχής Προσπάθεια!\n\n";
exit;
```

Παράδειγμα 5.2.4

Η έξοδος του προγράμματος αυτού είναι η παρακάτω:

Η αρχική ακολουθία DNA είναι:

ACGGGAGGACGGGAAAATTACTACGGCATTAGC

Η αντίστροφη συμπληρωματική ακολουθία DNA είναι:

GGAAAAGGGGAAGAAAAAAAGGGGAGGAGGGGA

Λάθος αλγόριθμος

Δεύτερη Προσπάθεια - Σωστή Μέθοδος

Η αντίστροφη συμπληρωματική ακολουθία DNA είναι:

GCTAATGCCGTAGTAATTTTCCCGTCCTCCCGT

Επιτυχής Προσπάθεια!

Στο πρόγραμμα 5.2.4 πρωτοεμφανίζεται η συνάρτηση reverse η οποία αντιστρέφει τη σειρά των στοιχείων μιας δομής συμπεριλαμβανομένων και των συμβολοσειρών. Η πρώτη λανθασμένη μέθοδος αντικαθιστά τα σύμβολα αδενίνης (A) με αυτά της θυμίνης (T), με την εντολή $revcom =~ s/A/T/g; Στη συνέχεια με την εντολή $revcom =~ s/T/A/g; αντικαθιστά τα σύμβολα της θυμίνης με αυτά της αδενίνης. Μετά το πέρας των δύο εντολών αυτών και

επειδή οι εντολές στην Perl εκτελούνται σειριακά, δεν θα υπάρχουν καθόλου σύμβολα θυμίνης. Τα αρχικά δηλαδή A και T έγιναν όλα A. Ομοίως οι εντολές $revcom =~ s/G/C/g; και $revcom =~ s/C/G/g; έχουν ως αποτέλεσμα την ύπαρξη μόνο συμβόλων G στη θέση των αρχικών C και G.

Η λύση δίνεται από τον τελεστή tr και την εντολή $revcom =~ tr/ACGTacgt/TGCAtgca/; . Η συντομογραφία tr προέρχεται από τη λέξη translation που σημαίνει μετάφραση. Ο τελεστής tr μεταφράζει ταυτοχρόνως ένα σύνολο χαρακτήρων σε νέους χαρακτήρες. Το σύνολο των νέων χαρακτήρων που αντικαθιστούν τους αρχικούς βρίσκεται μεταξύ της δεύτερης και τρίτης πλάγιας εμπρόσθιας γραμμής (/). Επίσης επειδή η ακολουθία DNA μπορεί να παρέχεται είτε με κεφαλαία είτε με μικρά γράμματα και οι δύο περιπτώσεις προβλέπονται στην εντολή αυτή.

Το παράδειγμα 5.2.5 διαβάζει από ένα αρχείο μια πρωτεϊνική ακολουθία την αποθηκεύει σε έναν πίνακα (array) και στη συνέχεια την εμφανίζει στην οθόνη. Με χρήση του text editor μπορούμε να δημιουργήσουμε ένα αρχείο που περιέχει μία πρωτεϊνική ακολουθία και να την ονομάσουμε NM_021964fragment.pep (εναλλακτικά το κατεβάζουμε από το διαδίκτυο). Η ακολουθία αυτή είναι η MNIDDKLEGLFLKCGGIDEMQSSRTMVVMGGVSGQSTVSGE LQDSVLQDRSMPHQEILAADEVLQESEMRQQDMISHDELMVHEETKNDEEQMETHE RLPQGLQYALNVPISVKQEITFTDVSEQLMRDKKQIR. Το πρόγραμμα λοιπόν είναι:

```
#!/usr/bin/perl -w
# Ο φάκελος που περιέχει την πρωτεϊνική ακολουθία είναι
$proteinfilename = 'NM_021964fragment.pep';
# Άνοιξε τον φάκελο
open(PROTEINFILE, $proteinfilename);
# Διάβασε την ακολουθία και αποθήκευσέ την σε array μεταβλητή
@protein = <PROTEINFILE>;
# Εμφάνισε την πρωτεΐνη στην οθόνη
print @protein;
# Κλείσε τον φάκελο
close PROTEINFILE;
exit;
```

Παράδειγμα 5.2.5

Η έξοδος (output) του προγράμματος είναι

MNIDDKLEGLFLKCGGIDEMQSSRTMVVMGGVSGQSTVSGELQDSVLQDRSMPHQEILA ADEVLQESEMRQQDMISHDELMVHEETVKNDEEQMETHERLPQGLQYALNVPISVKQEI TFTDVSEQLMRDKKQIR

Στο πρόγραμμα αυτό αρχικά αποθηκεύεται σε μία μεταβλητή το αρχείο NM_021964fragment.pep με την εντολή $proteinfilename = 'NM_021964fragment.pep'; και στη συνέχεια ανοίγεται ο φάκελος και συσχετίζεται με τον filehandle PROTEINFILE με την εντολή open(PROTEINFILE, $proteinfilename); .Με το σύμβολο (@) δηλώνονται οι πίνακες (arrays), ενώ οι αγκύλες (< >) χρησιμοποιούνται για την εισαγωγή δεδομένων από κάποια πηγή εξωτερικά του προγράμματος. Στη συνέχεια παρατίθενται παραδείγματα εντολών που χειρίζονται πίνακες.

- # Δήλωση ενός πίνακα που περιέχει 4 μονόμετρες μεταβλητές

  @bases = ('A', 'C', 'G', 'T');

  # Εμφάνιση του πρώτου στοιχείου

  print $bases[0];

Το τμήμα προγράμματος αυτό έχει ως αποτέλεσμα την εμφάνιση στην οθόνη του συμβόλου (A).

- Αντίθετα με το τμήμα

  @bases = ('A', 'C', 'G', 'T');

  print @bases;

εμφανίζονται στην οθόνη όλα τα στοιχεία του πίνακα δηλαδή ACGT ενώ με το τμήμα

  @bases = ('A', 'C', 'G', 'T');

  print "@bases";

εμφανίζονται τα στοιχεία του πίνακα χωριζόμενα από κενά (A C G T).

- Ένα στοιχείο μπορεί να εξαχθεί από το τέλος του πίνακα με την εντολή pop

  @bases = ('A', 'C', 'G', 'T');

  $base1 = pop @bases;

που στο παραπάνω παράδειγμα εξάγει το στοιχείο T.

- Στοιχείο εξάγεται από την αρχή ενός πίνακα με την εντολή shift

  @bases = ('A', 'C', 'G', 'T');

  $base2 = shift @bases;

που εδώ εξάγει το A.

- Στοιχείο εισάγεται στην αρχή ενός πίνακα με την εντολή unshift

  @bases = ('A', 'C', 'G', 'T');

  $base1 = pop @bases;

  unshift (@bases, $base1);

Το τμήμα αυτό προγράμματος αρχικά εξάγει το T από το τέλος ενός πίνακα και στη συνέχεια το εισάγει στην αρχή του. Ο τελικός πίνακας είναι λοιπόν T A C G.

- Στοιχείο εισάγεται στο τέλος ενός πίνακα με την εντολή push.

```
@bases = ('A', 'C', 'G', 'T');
$base2 = shift @bases;
push (@bases, $base2);
```

Το τμήμα αυτό εξάγει το A από την αρχή του πίνακα και στη συνέχεια το εισάγει στο τέλος του. Ο τελικός πίνακας είναι λοιπόν C G T A.

- Στοιχείο εισάγεται σε συγκεκριμένη θέση με τη συνάρτηση splice

```
@bases = ('A', 'C', 'G', 'T');
splice ( @bases, 2, 0, 'X');
```

που στη συγκεκριμένη περίπτωση θα εισαγάγει το σύμβολο X μετά το δεύτερο στοιχείο του πίνακα. Η τελική μορφή λοιπόν του πίνακα θα είναι A C X G T.

- Με τη συνάρτηση reverse που αναφέραμε στο παράδειγμα 5.2.4 μπορούμε να αντιστρέψουμε τα στοιχεία του πίνακα

```
@bases = ('A', 'C', 'G', 'T');
@reverse = reverse @bases;
```

Ο πίνακας @reverse τελικώς θα είναι η T G C A.

- Το μήκος ενός πίνακα εξάγεται για παράδειγμα με τη συνάρτηση scalar

```
@bases = ('A', 'C', 'G', 'T');
print scalar @bases, "\n";
```

που θα εμφανίσει τον αριθμό 4, ή με το παρακάτω τμήμα

```
@bases = ('A', 'C', 'G', 'T');
$a = @bases;
print $a, "\n";
```

που εμφανίζει και πάλι το 4.

- Αντιθέτως το τμήμα

```
@bases = ('A', 'C', 'G', 'T');
($a) = @bases;
print $a, "\n";
```

θα εμφανίσει το πρώτο στοιχείο του πίνακα που είναι το A.

5.2.4 *Αναζήτηση Μοτίβων*

Το παράδειγμα προγράμματος 5.2.6 εκτελεί μία από τις πλέον κοινές εφαρμογές των βιο-επιστημών, που είναι η αναζήτηση μοτίβων τα οποία μπορεί να είναι μικρού μήκους τμήματα DNA ή πρωτεϊνών. Τα τμήματα αυτά είναι ειδικού ενδιαφέροντος για τους ερευνητές και συνήθως είναι είτε ρυθμιστικά τμήματα DNA είτε μικρά κομμάτια πρωτεΐνης που αποδεδειγμένα διατηρούνται σε διάφορα είδη του ζωικού βασιλείου. Τα μοτίβα αυτά, που αναζητούνται σε βιολογικές ακολουθίες, έχουν διάφορες παραλλαγές. Συνήθως δεν είναι αυτούσιες ακολουθίες αφού για παράδειγμα σε κάποιες θέσεις η εκεί παρούσα νουκλεοτιδική βάση ή το αμινοξύ είναι αδιάφορα. Συχνά λοιπόν τα μοτίβα μπορούν να αναπαρασταθούν ως κανονικές εκφράσεις.

Το πρόγραμμα 5.2.6 διαβάζει μια πρωτεϊνική ακολουθία από κάποιον φάκελο, την τοποθετεί σε μια συμβολοσειρά για λόγους ευκολίας και στη συνέχεια αναζητά σε αυτήν μοτίβα που ο χρήστης πληκτρολογεί.

```
#!/usr/bin/perl -w
# Αναζήτηση μοτίβων
# Ζήτα από τον χρήστη το φάκελο με την πρωτεϊνική ακολουθία
print "Πληκτρολόγησε το όνομα του φακέλου: ";
$proteinfilename = <STDIN>;
chomp $proteinfilename;
# Άνοιξε τον φάκελο ή κάνε έξοδο
unless ( open(PROTEINFILE, $proteinfilename) ) {
exit; }
@protein = <PROTEINFILE>;
close PROTEINFILE;
# Τοποθέτησε την ακολουθία σε μία συμβολοσειρά ώστε
# η αναζήτηση μοτίβων να  είναι ευκολότερη
$protein = join( '', @protein);
# Μέσα σε βρόχο ζήτα από το χρήστη να εισάγει ένα μοτίβο
# και ανέφερε την εύρεση αλλιώς κάνε έξοδο.
do {
print "Δώσε το μοτίβο προς αναζήτηση: ";
$motif = <STDIN>;
chomp $motif;
```

```
if ( $protein =~ /$motif/ ) {
print "Το μοτίβο ανιχνεύθηκε.\n\n";
} else {
print "Το μοτίβο δεν ανιχνεύθηκε.\n\n";
}
} until ( $motif =~ /^\s*$/ );
exit;
```

Πρόγραμμα 5.2.6

Η έξοδος του προγράμματος 5.2.6 μπορεί να είναι

Πληκτρολόγησε το όνομα του φακέλου: NM_021964fragment.pep

Δώσε το μοτίβο προς αναζήτηση: SVLQ

Το μοτίβο ανιχνεύθηκε.

Δώσε το μοτίβο προς αναζήτηση: JKL

Το μοτίβο δεν ανιχνεύθηκε.

Δώσε το μοτίβο προς αναζήτηση: QDSV

Το μοτίβο ανιχνεύθηκε.

Δώσε το μοτίβο προς αναζήτηση: HERLPQGLQ

Το μοτίβο ανιχνεύθηκε.

Η χρησιμότητα του προγράμματος αυτού είναι προφανής, αφού κάνει την αναζήτηση μοτίβων ευκολότερη και γλιτώνει το χρήστη από την αντίστοιχη κουραστική χειρωνακτική εργασία. Στο πρόγραμμα αυτό το όνομα του φακέλου λαμβάνεται από τον χρήστη μέσω του filehandle STDIN (συντομογραφία του standard input). Με τη συνάρτηση chomp αφαιρούνται τα newlines από το τέλος του ονόματος του φακέλου που ο χρήστης εισάγει (οι χαρακτήρες νέας γραμμής εμφανίζονται όταν μετά από την πληκτρολόγηση του ονόματος ο χρήστης πατήσει το πλήκτρο ENTER), ενώ με τη συνάρτηση join τα στοιχεία του πίνακα @protein, που μπορεί να χωρίζονται από κενά, ενώνονται χωρίς κενά σε μια συμβολοσειρά που αποθηκεύεται στη μονόμετρη μεταβλητή $protein.

Η αναζήτηση για το αν το μοτίβο που πληκτρολογεί ο χρήστης γίνεται με τη γραμμή κώδικα if ($protein =~ /$motif/), όπου η κανονική έκφραση που είναι αποθηκευμένη ως τιμή της μεταβλητής $motif αναζητείται στην πρωτεΐνη $protein. Η αναζήτηση για την ύπαρξη ή όχι του μοτίβου τερματίζεται όταν πληκτρολογηθεί κενό μοτίβο και γίνεται με τη γραμμή κώδικα

until ($motif =~ /^\s*$/). Η κανονική έκφραση σε αυτήν την περίπτωση είναι η /^\s*$/ και σημαίνει μια συμβολοσειρά που στην αρχή της (που σηματοδοτείται με το σύμβολο ^) έχει μηδέν ή περισσότερους (σηματοδοτείται με το σύμβολο *) κενούς-whitespace χαρακτήρες όπως το κενό ή το tab (σηματοδοτούνται με το \s), έως και το τέλος της (που σηματοδοτείται με το σύμβολο $).

Συνεπώς εφόσον η αναζήτηση έγινε στην πρωτεϊνική ακολουθία MNIDDKLEGLFLKCG GIDEMQSSRTMVVMGGVSGQSTVSGELQD**SVLQ**DRSMPHQEILAADEVLQESEMRQ QDMISHDELMVHEETVKNDEEQMET**HERLPQGLQ**YALNVPISVKQEITFTDVSEQL MRDKKQIR είχαμε τα εξής αποτελέσματα :

- Όταν ο χρήστης αναζήτησε τη συμβολοσειρά χαρακτήρων SVLQ τότε αυτή ανιχνεύεται.
- Όταν αναζητά τη συμβολοσειρά HERLPQGLQ τότε αυτή ανιχνεύεται.
- Όταν αναζητά τη συμβολοσειρά JKL τότε αυτή δεν ανιχνεύεται.

Οι συμβολοσειρές χαρακτήρων ανήκουν στην απλούστερη μορφή μοτίβων που μπορούν να εισαχθούν από το χρήστη. Περισσότερο σύνθετες μορφές μοτίβων είναι οι παρακάτω :

- Αν ο χρήστης πληκτρολογήσει το μοτίβο **A[DS]V** τότε αναζητά το σύμβολο A ακολουθούμενο από το D ή το S και στη συνέχεια ένα V. Το μοτίβο αυτό στη συγκεκριμένη περίπτωση δε θα ανιχνευόταν.
- Αν ο χρήστης αναζητήσει το **KND*E{2,}** τότε αναζητά το KN ακολουθούμενο από μηδέν ή περισσότερα (*) D και στη συνέχεια δύο ή περισσότερα (**{2,}**) E. Το μοτίβο αυτό ανιχνεύεται (KNDEE).
- Αν ο χρήστης αναζητήσει το **EE.*EE** τότε ζητά δύο E ακολουθούμενα από οτιδήποτε (**.***), και στη συνέχεια άλλα δύο E. Το μοτίβο αυτό ανιχνεύεται (EETVKNDEE).

5.2.5 *Υπορουτίνες (Subroutines)*

Η χρήση των υπορουτινών αποτελεί μία σημαντική μέθοδο για την καλύτερη οργάνωση των προγραμμάτων σε μια γλώσσα προγραμματισμού. Μία υπορουτίνα είναι ουσιαστικά ένα σύνολο κώδικα (υποπρόγραμμα) το οποίο χρησιμοποιείται από το υπόλοιπο πρόγραμμα. Ένα πρόγραμμα λοιπόν καλεί μια υπορουτίνα με το όνομα της, δίνοντάς της ταυτόχρονα κάποιες τιμές ως ορίσματα. Η υπορουτίνα επεξεργάζεται τα ορίσματα αυτά και στη συνέχεια παρέχει στο πρόγραμμα που την κάλεσε, τα αποτελέσματα της επεξεργασίας αυτής.

Ένα απλό παράδειγμα της χρήσης, καθώς και του τρόπου ορισμού υπορουτινών είναι το 5.2.7. Το πρόγραμμα αυτό προσαρτεί στο τέλος μιας ακολουθίας DNA ένα άλλο τμήμα DNA (το ACGT).

```
#!/usr/bin/perl -w

# Πρόγραμμα με υπορουτίνα που προσαρτεί το ACGT στο DNA
```

```
# Το αρχικό DNA είναι:
$dna = 'CGACGTCTTCTCAGGCGA';
# Κλήση της υπορουτίνας "addACGT".
# Το όρισμά της είναι το $dna και το αποτέλεσμα θα αποθηκευθεί στο $neo_dna
$neo_dna = addACGT($dna);
print "Προσάρτησα το ACGT στο $dna και έλαβα  το $neo_dna\n\n";
exit;
# Ορισμός της υπορουτίνας "addACGT"
sub addACGT {
my($dna) = @_;
$dna .= 'ACGT';
return $dna;
}
```

Παράδειγμα 5.2.7

Το παράδειγμα αυτό έχει την ακόλουθη έξοδο (output).

Προσάρτησα το ACGT στο CGACGTCTTCTCAGGCGA και έλαβα το CGACGTCTTCTCAGGCGAACGT

Το πρόγραμμα αυτό αποτελείται από δύο μέρη. Το πρώτο και κύριο αρχίζει από την αρχή και τερματίζεται με την εντολή exit, ενώ το δεύτερο είναι ο ορισμός της υπορουτίνας addACGT. Η κλήση της γίνεται με την εντολή addACGT($dna);. Καλείται με το όνομά της, ακολουθούμενο από παρενθέσεις εντός των οποίων εμφανίζονται τα ορίσματα της. Είναι δυνατόν και να μην υπάρχουν ορίσματα, ή να υπάρχουν περισσότερα του ενός οπότε και χωρίζονται από κόμματα (,).

Ο ορισμός της υπορουτίνας γίνεται με τη δεσμευμένη λέξη sub ακολουθούμενη από το όνομα της υπορουτίνας (εδώ addACGT) και στη συνέχεια ένα τμήμα κώδικα που εσωκλείεται στα σύμβολα ({ }). Οι μεταβλητές που δηλώνονται με το τμήμα my στην αρχή τους, περιορίζονται στα όρια της υπορουτίνας (είναι τοπικές μεταβλητές) και οι τιμές των ορισμάτων περνούν στην υπορουτίνα μέσω της ειδικής μεταβλητής πίνακα @_ . Τέλος, οι περισσότερες υπορουτίνες επιστρέφουν τα αποτελέσματά τους μέσω της συνάρτησης return.

Η υπορουτίνα 5.2.8 που ακολουθεί είναι εξαιρετικά χρήσιμη επειδή χειρίζεται δεδομένα που βρίσκονται σε μορφή FASTA (FASTA format). Η μορφή FASTA χρησιμοποιείται από πολλά προγράμματα εξαιτίας της απλότητάς της, μεταξύ των οποίων και το BLAST. Τα δεδομένα της μορφής αυτής είναι ουσιαστικά γραμμές δεδομένων ακολουθιών με χαρακτήρες newline στο τέλος κάθε γραμμής, ώστε να εκτυπώνονται ή να εμφανίζονται καλύτερα στην οθόνη του

υπολογιστή. Τα δεδομένα αυτά έχουν στην αρχή τους επικεφαλίδα που ξεκινά με το σύμβολο του μεγαλύτερου (>). Το σύμβολο αυτό ακολουθείται προαιρετικά από το όνομα της ακολουθίας, το οποίο με τη σειρά του μπορεί να ακολουθείται από την κατακόρυφη γραμμή (|), ώστε μετά από αυτήν να παρέχονται επιπλέον πληροφορίες. Επίσης στα αρχεία αυτά μπορεί να υπάρχουν γραμμές σχολίων οι οποίες όπως και πριν ξεκινούν με το σύμβολο της δίεσης (#). Χαρακτηριστικό παράδειγμα αρχείου σε μορφή FASTA είναι το αρχείο sample.dna που ακολουθεί.

> sample dna | …*Επιπλέον Πληροφορίες*

agatggcggcgctgaggggtcttgggggctctaggccggccacctactggtttgcagcggagacgacgc

atggggcctgcgcaataggagtacgctgcctgggaggcgtgactagaagcggaagtagttgtgggcgcc

tttgcaaccgcctgggacgccgccgagtggtctgtgcaggttcgcgggtcgctggcgggggtcgtgagg

gagtgcgccgggagcggagatatggagggagatggttcagacccagagcctccagatgccggggagaca

```
# Υπορουτίνα : Ανάγνωση και εξαγωγή ακολουθίας από δεδομένα σε μορφή FASTA
sub extract_sequence_from_fasta_data {
my(@fasta_file_data) = @_;
use warnings;
# Αρχικοποίηση μεταβλητών
my $sequence = '';
foreach my $line (@fasta_file_data) {
# Απόρριψη κενών γραμμών
if ($line =~ /^\s*$/) {
next;
# Απόρριψη γραμμών σχολίων
} elsif($line =~ /^\s*#/) {
next;
# Απόρριψη επικεφαλίδας μορφής FASTA
} elsif($line =~ /^>/) {
next;
# Φύλαξε τη γραμμή και προσέθεσέ την στη συμβολοσειρά της ακολουθίας
} else {
$sequence .= $line;
}
```

```
}
# Αφαίρεσε τα κενά και στη συνέχεια επέστρεψε την ακολουθία
from $sequence string
$sequence =~ s/\s//g;
return $sequence;
}
```

Υπορουτίνα 5.2.8

Η υπορουτίνα επεξεργάζεται το αρχείο FASTA ως εξής: Για κάθε γραμμή του αρχείου ελέγχει εάν πρόκειται για κενή γραμμή, γραμμή σχολίων, ή γραμμή επικεφαλίδας. Εάν δεν πρόκειται για γραμμή που ανήκει σε μια από τις τρεις κατηγορίες γραμμών, τότε πρόκειται για γραμμή ακολουθίας, οπότε και προστίθεται σε μια αρχικά κενή συμβολοσειρά. Έπειτα ελέγχεται εάν υπάρχουν κενά μέσα στη συμβολοσειρά τα οποία και διαγράφονται. Τελικά η συμβολοσειρά επιστρέφεται ως αποτέλεσμα της υπορουτίνας.

5.2.6 *Χρήση του BioPerl*

Η αυτοματοποίηση χρήσης του προγράμματος BLAST μέσω της γλώσσας Perl διευκολύνει πολύ τον χρήστη καθώς τον απαλάσσει από το άνοιγμα και την επεξεργασία διαφορετικών ιστοσελίδων και διαδικτυακών συνδέσμων (links). Για να εκτελέσουμε έρευνες του BLAST χρησιμοποιούμε το BioPerl μέρος (module) της Perl, δηλαδή τους έτοιμους τύπους και συναρτήσεις που έχουν αναπτυχθεί για τις Βιοεπιστήμες. Ορισμένοι εκ των τύπων αυτών απεικονίζονται στον πίνακα

Bio::Seq	Κύριο αντικείμενο ακολουθιών της BioPerl
Bio::SeqIO	Παρέχει αρχεία ακολουθιών ως είσοδο ή εξοδο
Bio::DB::GenBank	Παρέχει πρόσβαση στη βάση δεδομένων GenBank. Παρόμοια modules είναι διαθέσιμα για άλλες βάσεις δεδομένων
Bio::Tools::Run::StandAloneBlast	Εκτελεί τοπικά (χωρίς σύνδεση internet) το BLAST
Bio::Tools::Run::RemoteBlast	Εκτελεί διαδικτυακά το BLAST
Bio::Clustalw	Παρέχει πρόσβαση στο πακέτο πολλαπλών ευθυγραμμίσεων Clustalw

Πίνακας 5.1: Τύποι του BioPerl module

Έτσι το πρόγραμμα 5.2.9 χρησιμοποιεί έτοιμες συναρτήσεις για να εκτελέσει μέσω internet το πρόγραμμα blastp, ρυθμίζοντας παραμέτρους, όπως η τιμή E ($1\times e^{-10}$), ή η βάση δεδομένων στην οποία θα γίνει η έρευνα (ecoli). Η ευκολία που παρέχει η εκτέλεση του BLAST μέσω ενός σεναρίου της Perl έγκειται ακριβώς στο ότι είναι απλούστερο να αλλάξει κάποιος μία παράμετρο στο έτοιμο σενάριο και μετά να το εκτελέσει, παρά να επεξεργάζεται ιστοσελίδες, να ακολουθεί διαδικτυακούς συνδέσμους (links) κ.λ.π.

```
use Bio::Tools::Run::RemoteBlast;
my $prog = 'blastp';
my $db = 'ecoli';
my $e_val= '1e-10';
my $remote_blast = Bio::Tools::Run::RemoteBlast->new(
-prog => $prog,
-data => $db,
-expect => $e_val);
my $r = $remote_blast->submit_blast($inputfilename);
while( my @rids = $remote_blast->each_rid ) {
for my $rid ( @rids ) {
my $rc = $remote_blast->retrieve_blast($rid);
if( ! ref($rc) ) {
if( $rc < 0 ) { $remote_blast->remove_rid($rid); }
print STDERR ".”; sleep(10);
} else {
$remote_blast->remove_rid($rid);
my $result = $rc->next_result;
while( my $hit = $result->next_hit ) {
print $hit->name, “ “, $hit->significance, “\n”;
}
}
```

Πρόγραμμα 5.2.9

5.3 *Γλώσσα Python*

Η Python είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού που δημοσιοποιήθηκε για πρώτη φορά το 1991 από τον Guido van Rossum. Διατίθεται δωρεάν μέσω διαδικτύου από την επίσημη ιστοσελίδα της Python ([Pyt07]) και η πλέον πρόσφατη έκδοσή της, (Python 2.5) είναι αυτή που μελετούμε. Η Python είναι περισσότερο ευανάγνωστη και ευκολότερα διαμορφώσιμη σε σχέση με την Perl. Έχει τη δυνατότητα εύκολης πρόσβασης σε κώδικες άλλων γλωσσών προγραμματισμού όπως στη C, C++ και Fortran, είναι και αυτή συμβατή με όλα σχεδόν τα λειτουργικά συστήματα, ενώ πληθαίνει και η κοινότητα των προγραμματιστών που χρησιμοποιούν τη γλώσσα Python για βιολογικές εφαρμογές.

5.3.1 *Εισαγωγή στη Γλώσσα Python*

Η Python προσφέρει όπως και η Perl ευκολία στον προγραμματισμό, συμβατότητα με πολλά λειτουργικά συστήματα και ικανότητα μεταφοράς προγραμμάτων μεταξύ αυτών. Είναι επίσης ταχεία στην εκτέλεση και παρέχει τη δυνατότητα εύκολης συντήρησης και τροποποίησης προγραμμάτων. Διαθέσιμη είναι επίσης μία συλλογή κώδικα της Python που περιέχει χρήσιμες, για τις Βιοεπιστήμες, συναρτήσεις και ονομάζεται Biopython (αντίστοιχη της Bioperl της γλώσσας Perl).

Στη συνέχεια παρατίθενται παρατηρήσεις επί των βασικών στοιχείων της γλώσσας Python:

- Τα τρία σύμβολα του μεγαλυτέρου (>) εμφανίζονται στην αρχή κάθε γραμμής εντολών.

- Η εισαγωγή σχολίων γίνεται με τη χρήση του συμβόλου της δίεσης (**#**) στην αρχή της γραμμής, ενώ οι εντολές τερματίζονται (προαιρετικά) με το ελληνικό ερωτηματικό (**;**).
- Οι συμβολοσειρές (strings) μπορούν να αποτελούνται από οποιοδήποτε σύνολο αριθμητικών ή αλφαβητικών χαρακτήρων που εσωκλείονται σε μονά ή διπλά εισαγωγικά (quotes) και δεν μπορούν να μεταβληθούν, παρά μόνο να οριστούν νέες.
- Τα μπλοκ κώδικα ξεκινούν με τη χρήση του συμβόλου (**:**) και ορίζονται βάσει της στοίχισης και όχι με τη χρήση συμβόλων. Η θέση της πρώτης εντολή ενός μπλοκ μπορεί να επιλεχθεί αλλά οι επόμενες πρέπει υποχρεωτικά να στοιχίζονται κάτω από την πρώτη.
- Στη γλώσσα Python χρησιμοποιείται η δομή των λιστών, τα στοιχεία των οποίων μπορούν να μεταβληθούν, καθώς και η δομή των tuples τα στοιχεία των οποίων δε μεταβάλλονται. Τα ακόλουθα παραδείγματα δείχνουν τις δύο δομές.

```
>>>#Λίστα a
>>>a=[1, 5, 9];
>>>#Εδώ το πρώτο στοιχείο της λίστας a, a[0]=1, οπότε με την εντολή που ακολουθεί
>>>a[0]=10;
>>>#Η λίστα έχει πλέον τα στοιχεία [10, 5, 9]
>>>a.append(0);
>>>#Η λίστα έχει πλέον τα στοιχεία [10, 5, 9, 0]
>>>del a[1];
>>>#Η λίστα είναι πλέον η [10, 9, 0]
>>>a.reverse( );
>>>#Τώρα a = [0, 9, 10]
>>>a.pop( );
>>>#a = [0, 9]
>>>a.insert(2, “GTGTGT”);
>>>#a = [0, 9, “GTGTGT”]
>>>a = a + [“ATGTGCT”, “CGCGC”];
>>>#a = [0, 9, “GTGTGT”, “ATGTGCT”, “CGCGC”]
>>>a = a*2;
>>>#a = [0, 9, “GTGTGT”, “ATGTGCT”, “CGCGC”, 0,
>>># 9, “GTGTGT”, “ATGTGCT”, “CGCGC”]
>>>#Tuple b. Η χρησιμότητα των tuples οφείλεται ακριβώς στο αμετάβλητο του
```

>>>#χαρακτήρα τους

>>>b=(2, 7, 8);

>>>#Εδώ λοιπόν η ακόλουθη εντολή δεν θα έχει καμία επίδραση στη δομή μας και θα

>>>#προκαλέσει την εμφάνιση μηνύματος συντακτικού λάθους

>>>b(1)=5;

Syntax error

5.3.2 *Χειρισμός Ακολουθιών*

Οι ακολουθίες αποθηκεύονται στη γλώσσα Python ως συμβολοσειρές. Για παράδειγμα με την εντολή s="GATTACA" αποθηκεύεται η συμβολοσειρά s, το πρώτο της στοιχείο της οποίας είναι το G (s[0]=G) και το τελευταίο είναι το A (s[6]=A=s[-1]). Παραδείγματα και τρόποι πρόσβασης και κτήσης υποσυμβολοσειρών (substrings) δίνονται παρακάτω. Για παράδειγμα με την εντολή s[3:5] παίρνουμε το τμήμα της συμβολοσειράς από το τέταρτο έως και το έκτο της σύμβολο (η αρίθμηση ξεκινά από το 0), ενώ με την εντολή s[4:] παίρνουμε το τμήμα της συμβολοσειράς από το πέμπτο της σύμβολο, έως και το τελευταίο της.

>>>s[1:3]

AT

>>>s[:3]

GAT

>>>s[4:]

ACA

>>>s[3:5]

TA

>>>s[:]

GATTACA

>>>s[::2]

GTAA

>>>s[-2:2:-1]

CAT

ενώ κάποιες εκ των σημαντικότερων μεθόδων χειρισμού συμβολοσειρών είναι οι εξής:

>>>len(“GATTACA”) #Μήκος

7

>>>“GAT“ + “TACA“ #Αλυσιδωτή Σύνδεση

GATTACA

```
>>>“A” * 10                    #Επανάληψη
AAAAAAAAAA
>>>“G“ in “GATTACA“           #Ελεγχος Υποσυμβολοσειρών
True
>>>“GAT“ in “GATTACA“         #Ελεγχος Υποσυμβολοσειρών
True
>>>“AGT“ in “GATTACA“         #Ελεγχος Υποσυμβολοσειρών
False
>>>“GATTACA“.find(“ATT“)      #Τοποθεσία Υποσυμβολοσειράς
1
>>>“GATTACA“.count(“T“)       #Μέτρηση Παρουσίας Υποσυμβολοσειράς
2
>>>”GATTACA”.lower( )          #Μετατροπή από Κεφαλαία σε πεζά
gattaca
>>>”gattaca”.upper( )          #Μετατροπή από πεζά σε Κεφαλαία
GATTACA
>>>”GATTACA“.replace(“G“, “U”)  #Αντικατάσταση Στοιχείων
UATTACA
>>>”GATTACA“.startswith(”G”)    #Ελεγχος Στοιχείου Έναρξης Συμβολοσειράς
True
>>>”GATTACA“.startswith(”g”)    #Ελεγχος Στοιχείου Έναρξης Συμβολοσειράς
False
```

Το πρόγραμμα 5.3.1α λοιπόν χρησιμοποιεί ορισμένες εκ των μεθόδων που αναφέρθηκαν, προκειμένου να υπολογίσει το ποσοστό ύπαρξης της δινουκλεοτιδικής ακολουθίας “GC”, εντός μιας συμβολοσειράς που εισάγεται από τον χρήστη, μέσω της συνάρτησης raw_input. Ο βαθμός παρουσίας της ‘GC’ ποικίλλει ανά οργανισμό και είναι υπεύθυνος για διαφορές στην τρισδιάστατη δομή του DNA καθώς και στην αντιγραφή και μεταγραφή του. Κατά συνέπεια είναι αρκετά σημαντικός και αξίζει να υπολογιστεί.

```
>>>sequence = raw_input("Δώσε την ακολουθία προς μελέτη : ");
>>>#Το string εντός των παρενθέσεων της raw_input θα εμφανιστεί στην οθόνη
>>>#Το μήκος της ακολουθίας εισόδου είναι:
>>>length = len(sequence);
>>>#ενώ το πλήθος εμφανίσεων της ακολουθίας GC προκύπτει από την εντολή
>>>numb = sequence.count("GC");
>>>#και το επί τοις εκατό ποσοστό παρουσίας της υποσυμβολοσειράς είναι
>>>percentage = (float(numb)/length)*200;
>>>#όπου ο πολλαπλασιασμός επί 200 γίνεται επειδή θέλουμε να κάνουμε το κλάσμα
```

```
>>>#επί τοις εκατό και η ακολουθία προς αναζήτηση έχει μήκος 2, άρα επί 200
>>>print percentage;
```

Πρόγραμμα 5.3.1α

Η εκτέλεση του προγράμματος αυτού μπορεί να είναι η εξής:

Δώσε την ακολουθία προς μελέτη : GCTGCTAGCTAGCTAGCTACGATGTCCGTA

33. 333333333333329

όπου η ακολουθία εισόδου είναι μήκους 30 χαρακτήρων, η ακολουθία GC εμφανίζεται 5 φορές, άρα όντως το ποσοστό εμφάνισής της είναι $\frac{5\times 2\times 100}{30} = \frac{1000}{30} = 33.333...\%$.

Το πρόγραμμα αυτό μπορεί εύκολα να τροποποιηθεί στη μορφή 5.3.1β προκειμένου να υπολογίζει το ποσοστό παρουσίας οποιουδήποτε τμήματος DNA σε ακολουθία που εισάγεται από τον χρήστη.

```
>>>study_sequence = raw_input("Δώσε την ακολουθία προς μελέτη");
>>>#Μήκος ακολουθίας εισόδου
>>>length1 = len(study_sequence);
>>>search_sequence = raw_input("Δώσε την ακολουθία προς αναζήτηση");
>>>#Μήκος ακολουθίας που θα αναζητήσουμε
>>>length2 = len(search_sequence);
>>>#Πλήθος εμφανίσεων της αναζητούμενης
>>>numb = study_sequence.count(search_sequence);
>>>#Τότε το ποσοστό παρουσίας της αναζητούμενης υπολογίζεται από τον τύπο
>>>percentage = (float(numb)/length1)*length2*100;
>>>print percentage;
```

Πρόγραμμα 5.3.1β

Το πρόγραμμα 5.3.2 που ακολουθεί αφορά τμήματα DNA που αντιστοιχούν σε τοποθεσίες περιοριστικών ενζύμων (restriction enzyme sites-*Τα περιοριστικά ένζυμα είναι πρωτεΐνες που τέμνουν ακολουθίες DNA, με αποτέλεσμα οι θέσεις τομής να μένουν ελεύθερες ώστε ο βιολόγος να μπορεί να εισάγει συγκεκριμένες ακολουθίες που επιθυμεί*). Οι τοποθεσίες αυτές είναι εκατοντάδες, αλλά οι γνωστότερες αυτών είναι οι ECORI (GATTC), HindIII(AAGCTT), BamHI(GGATCC). Είναι ευνόητη λοιπόν η ανάγκη δημιουργίας του προγράμματος που αναζητά την ύπαρξη ή όχι των τοποθεσιών αυτών σε ακολουθίες DNA.

Το πρόγραμμα αυτό αρχικώς ορίζει μία λίστα με όνομα restriction_sites η οποία έχει ως στοιχεία τις συμβολοσειρές των περιοριστικών ενζύμων και στη συνέχεια ελέγχει την ύπαρξη κάθε στοιχείου της λίστας σε ακολουθία που εισάγεται από τον χρήστη. Στο τέλος εμφανίζει

στην οθόνη μηνύματα παρουσίας ή απουσίας των τοποθεσιών αυτών. Το πρόγραμμα αυτό είναι εξαιρετικά εύχρηστο καθώς μεταβάλλοντας τα στοιχεία της λίστας μπορούμε να προσθέτουμε ή να αφαιρούμε τοποθεσίες περιοριστικών ενζύμων προς αναζήτηση, ενώ αλλάζοντας τα περιεχόμενα της λίστας καθώς και τις εντολές εμφάνισης μηνυμάτων στην οθόνη (print) μπορούμε να αναζητούμε οποιουδήποτε είδους και ενδιαφέροντος ακολουθίες.

```
>>>restriction_sites = [ "GAATTC", # EcoRI
                         "GGATCC", # BamHI
                         "AAGCTT", # HindIII
                         ];
>>>sequence = raw_input("Δώσε ακολουθία DNA προς μελέτη: ");
>>>for site in restriction_sites:
        if site in sequence:
           print "Η ακολουθία", site, "υπάρχει ως περιοριστικό ένζυμο.";
        else:
           print "Η ακολουθία", site, "δεν εμφανίζεται.";
```

Πρόγραμμα 5.3.2

Αν λοιπόν δοθεί ως ακολουθία εισόδου η "ATGTCGTAGCTA**GAATTC**GTAGATAGATG TCGATCGATCAGCTACGAT**AAGCTT"** τότε το πρόγραμμα θα εμφανίσει στην οθόνη τα παρακάτω μηνύματα:

Η ακολουθία GAATTC υπάρχει ως περιοριστικό ένζυμο.

Η ακολουθία GGATCC δεν εμφανίζεται.

Η ακολουθία AAGCTT υπάρχει ως περιοριστικό ένζυμο.

Βεβαίως στα παραπάνω προγράμματα ο χρήστης θα μπορούσε, με απλές τροποποιήσεις στα προγράμματα, να αποφύγει την ενδεχόμενη κοπιαστική εισαγωγή μιας μεγάλου μήκους ακολουθίας, προκειμένου να λαμβάνεται η ακολουθία από κάποιον φάκελο ήδη αποθηκευμένο στον υπολογιστή. Η παραπάνω διαδικασία γίνεται π.χ στο πρόγραμμα 5.3.2 εάν η εντολή sequence = raw_input("Δώσε ακολουθία DNA προς μελέτη: "); αντικατασταθεί από την εντολή sequence = open("C:/…/Python25/input.seq");. Σε αυτήν την περίπτωση θα ήταν χρήσιμο να αναφέρουμε μία μέθοδο αφαίρεσης από μια ακολουθία, whitespace χαρακτήρων. Οι χαρακτήρες αυτοί μπορεί να είναι οι tab (\t), newline (\n) κλπ και μπορούν με τη μέθοδο 5.3.3 να αφαιρεθούν (αντικατασταθούν με την κενή συμβολοσειρά ""). Με τον ίδιο ακριβώς τρόπο αφαιρούνται και οι αριθμοί που συνήθως είναι δείκτες του μήκους ακολουθιών.

```
>>> from string import *
>>> whitespace;  #Εμφάνιση στην οθόνη των χαρακτήρων αυτών
'\t\n\x0b\x0c\r '
>>> dna = open("C:/…input.seq");
```

```
>>> for s in whitespace:
        dna = replace(dna, s, "");
>>> digits;
'0123456789'
>>> for s in digits:
        dna=replace(dna,s,");
```

Μέθοδος 5.3.3

Ιδιαιτέρως χρήσιμη είναι επίσης η δομή του λεξικού, η οποία αναλύεται στα παραδείγματα 5.3.4 και 5.3.5. Στο παράδειγμα 5.3.4 ορίζεται το λεξικό complement_table ως το λεξικό των συμπληρωματικών βάσεων και είναι το complement_table = {"A": "T", "T": "A", "C": "G", "G": "C"}. Στη δομή αυτή ένα στοιχείο ορίζεται ως ένα ζεύγος συμβολοσειρών χωριζομένων από το σύμβολο (**:**), όπου η συμβολοσειρά-κλειδί (key) στα αριστερά αντιστοιχεί στην συμβολοσειρά-τιμή (value) στα δεξιά.

```
>>>complement_table = {"A": "T", "T": "A", "C": "G", "G": "C"}
>>> seq = raw_input("Δώσε την ακολουθία DNA για να βρεις την αντίστροφη
    συμπληρωματική της");
>>> new_seq = [ ] ;
>>>#Για κάθε σύμβολο της ακολουθίας εισόδου
>>> for letter in seq:
>>>#Αντικατέστησέ το με την τιμή του από τον πίνακα complement_table και
>>>#προσέθεσέ το στην κενή λίστα new_seq
        complement_letter = complement_table[letter] ;
        new_seq.append(complement_letter);
>>>print new_seq;
>>>#Αφού υπολογίστηκε η συμπληρωματική, αντέστρεψέ την
>>> new_seq.reverse( );
>>> print new_seq;
>>>#Εκτύπωσε τα στοιχεία της αφού τα συγχωνεύσεις σε μία συμβολοσειρά
>>> print "".join(new_seq);
```

Πρόγραμμα 5.3.4

Το πρόγραμμα 5.3.5 είναι σημαντικότατο εφόσον χρησιμεύει στη μετάφραση του DNA. Οι ακολουθίες DNA ουσιαστικά μεταφράζονται σε πρωτεΐνες, με μία τριάδα νουκλεοτιδίων να μεταφράζεται σε ένα αμινοξύ. Συνεπώς το λεξικό table χρησιμεύει στην αντιστοίχιση των τριάδων νουκλεοτιδίων σε αμινοξέα (εκτός από τις TAA, TGA, και TAG που είναι κωδικόνια τερματισμού στα οποία αντιστοιχίζουμε το σύμβολο *). Μάλιστα το πρόγραμμα αυτό μεταφράζει το πρώτο πλαίσιο ανάγνωσης (reading frame) της ακολουθίας (το DNA έχει έξι πιθανά πλαίσια ανάγνωσης).

```
>>>table = { 'TTT': 'F', 'TCT': 'S', 'TAT': 'Y', 'TGT': 'C', 'TTC': 'F',
            'TCC': 'S', 'TAC': 'Y', 'TGC': 'C', 'TTA': 'L', 'TCA': 'S',
            'TAA': '*' ,'TGA': '*', 'TTG': 'L', 'TCG': 'S', 'TAG': '*',
            'TGG': 'W','CTT': 'L', 'CCT': 'P', 'CAT': 'H', 'CGT': 'R',
            'CTC': 'L', 'CCC': 'P', 'CAC': 'H', 'CGC': 'R', 'CTA': 'L',
            'CCA': 'P', 'CAA': 'Q', 'CGA': 'R', 'CTG': 'L', 'CCG': 'P',
            'CAG': 'Q', 'CGG': 'R', 'ATT': 'I', 'ACT': 'T', 'AAT': 'N',
            'AGT': 'S', 'ATC': 'I', 'ACC': 'T', 'AAC': 'N', 'AGC': 'S',
            'ATA': 'I', 'ACA': 'T', 'AAA': 'K', 'AGA': 'R', 'ATG': 'M',
            'ACG': 'T', 'AAG': 'K', 'AGG': 'R', 'GTT': 'V', 'GCT': 'A',
```

```
                'GAT': 'D', 'GGT': 'G', 'GTC': 'V', 'GCC': 'A', 'GAC': 'D',
                'GGC': 'G', 'GTA': 'V', 'GCA': 'A', 'GAA': 'E', 'GGA': 'G',
                'GTG': 'V', 'GCG': 'A', 'GAG': 'E', 'GGG': 'G'}
>>>seq = raw_input("Δώσε την ακολουθία DNA : ");
>>>protein = [ ];
>>>#Από την αρχή ως το τέλος αφαιρώντας τα πλεονάζοντα νουκλεοτίδια
>>>for i in range(0, len(seq) - len(seq)%3, 3):
>>>#Αντικατέστησε τις βάσεις σύμφωνα με το λεξικό και προσάρτησέ τα αμινοξέα
>>>#σε μία αρχικά κενή λίστα
    protein.append( table[seq[i:i+3]] );
    protein = "".join(protein);
>>>print protein;
```

Πρόγραμμα 5.3.5

Σχολιάζοντας το πρόγραμμα πρέπει να αναφέρουμε πως η συνάρτηση range συντάσσεται με τη μορφή range(αρχή X, τέλος Y, βήμα P) και σημαίνει από την αρχή X έως το τέλος Y, με βήμα P. Επειδή τα κωδικόνια είναι τριπλέτες νουκλεοτιδίων χρησιμοποιούμε τη modulo διαίρεση με το 3, ώστε εάν το μήκος της ακολουθίας δεν είναι πολλαπλάσιο του 3 να αγνοούνται τα δύο ή το ένα νουκλεοτίδια που περισσεύουν (κάτι που γίνεται αφαιρώντας το υπόλοιπο της διαίρεσης από το μήκος της αρχικής ακολουθίας). Εάν το μήκος είναι πολλαπλάσιο του 3, τότε αφού len(seq)%3=0, ισχύει και ότι len(seq) – len(seq)%3 = len(seq).

Επίσης αξίζει να δειχθεί ο τρόπος ορισμού των συναρτήσεων (functions) που φαίνεται στο πρόγραμμα 5.3.6.

```
>>>from string import * #Εισαγωγή όλων (*) των έτοιμων συναρτήσεων για strings
>>>def complement(dna):
>>>#Εάν η πρώτη εντολή μιας συνάρτησης είναι συμβολοσειρά εγκλεισμένη σε
>>>#τριπλά quotes τότε είναι documentation αυτής
    """function to calculate the complement of a DNA sequence"""
>>>#Χρήση συνάρτησης maketrans που αντικαθιστά στοιχεία σε ένα string
>>>  tab = maketrans("AGCTagct", "TCGAtcga");
>>>Επιστροφή αποτελεσμάτων
    return translate(dna, tab);
```

Ορισμός Συνάρτησης 5.3.6

Ο ορισμός των συναρτήσεων αρχίζει με τη λέξη def ακολουθούμενη από το όνομα της συνάρτησης, το οποίο ακολουθείται από παρενθέσεις στις οποίες εσωκλείονται οι παράμετροι της συνάρτησης (όταν αυτές υπάρχουν). Στη συνέχεια, μετά την άνω-κάτω τελεία (colon) ακολουθούν οι εντολές ενώ στο τέλος επιστρέφεται το αποτέλεσμα με την εντολή return και γίνεται έξοδος από τη συνάρτηση. Αφού οριστεί μία συνάρτηση, χρησιμοποιείται με απλή κλήση του ονόματός της, ακολουθούμενο από τις παραμέτρους της μέσα σε παρενθέσεις.

5.3.3 *Κανονικές Εκφράσεις (Regular Expressions)*

Οι κανονικές εκφράσεις αποτελούν και στη γλώσσα Python χρήσιμο εργαλείο για την αναζήτηση μοτίβων. Έχοντας ήδη δείξει τις απλούστερες κανονικές εκφράσεις (συνεχή τμήματα συμβολοσειρών) και τις συναρτήσεις που χρησιμοποιούνται για την εύρεση, θα παραθέσουμε τις περισσότερο πολύπλοκες συντάξεις κανονικών εκφράσεων.

- Τα σύμβολα [ ] αναζητούν οποιοδήποτε εκ των συμβόλων που εσωκλείουν. Για παράδειγμα είναι συνήθης η αναζήτηση υδροφοβικών αμινοξέων σε μία πρωτεϊνική ακολουθία. Τα αμινοξέα αυτά είναι τα F, I, L, A, P, V, M και συνεπώς όταν θέλουμε να εντοπίσουμε αν υπάρχει κάποιο εκ των υδροφοβικών αμινοξέων συντάσσουμε την κανονική έκφραση [FILAPVM].
- Εάν θέλουμε να εντοπίσουμε την απουσία ενός συμβόλου (συνήθως αμινοξέως) χρησιμοποιούμε το σύμβολο ^ ανάμεσα στις αγκύλες [ ]. Η έκφραση λοιπόν [^P] ικανοποιείται για οποιοδήποτε άλλο σύμβολο εκτός από P.
- Η χρήση της τελείας (.) συνεπάγεται ταίριασμα (matching) με οποιοδήποτε σύμβολο.
- Τα σύμβολα {} σημαίνουν επανάληψη του προηγούμενου μοτίβου-έκφρασης και χρησιμοποιούνται με κάποιον αριθμό ανάμεσά τους, ο οποίος και δείχνει τον αριθμό των επαναλήψεων αυτών. Για παράδειγμα η έκφραση [FILAPVM]{5} ικανοποιείται αν εμφανιστεί οποιοδήποτε αμινοξύ εκ των F, I, L, A, P, V, M, σε 5 συνεχόμενες θέσεις.
- Εάν επιθυμούμε να καθορίσουμε το εύρος επαναλήψεων του προηγούμενου μοτίβου, τότε χρησιμοποιούμε την έκφραση {m, n} που ικανοποιείται εάν η προηγούμενη έκφραση επαναληφθεί τουλάχιστον m και έως n φορές. Μάλιστα κάποια εύρη επαναλήψεων που συναντούνται πολύ συχνά έχουν κωδικοποιηθεί με κάποια σύμβολα για λόγους ταχύτητας και ευκολίας:

  {0, 1} → ? = προαιρετική εμφάνιση

  {0, } → * = μηδέν ή περισσότερα

  {1, } → + = τουλάχιστον ένα

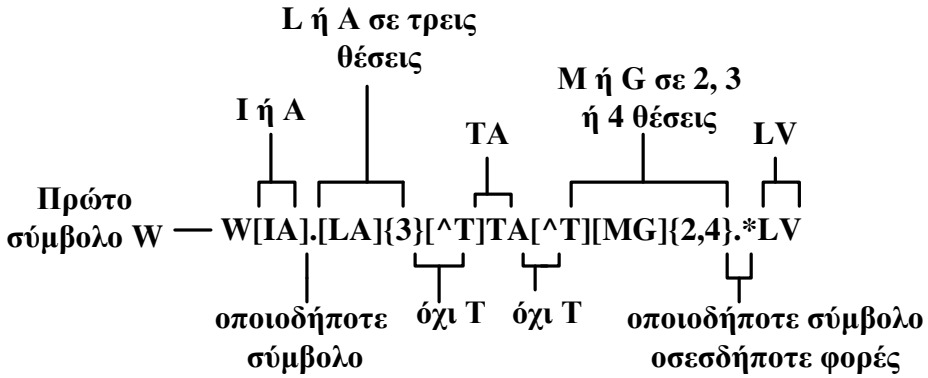
- Το σύμβολο ^ σημαίνει αρχή ακολουθίας, ενώ το σύμβολο του δολαρίου $, το τέλος.
- Επιπλέον συντομογραφίες για αριθμούς, γράμματα και κενά είναι οι:
  \d = [0123456789]
  \w = γράμματα, ψηφία και χαρακτήρας υπογράμμισης
  \s = whitespace χαρακτήρες (space, newline, tab κλπ)

Για να χρησιμοποιηθούν οι κανονικές εκφράσεις πρέπει πρώτα να μεταγλωττιστούν. Η διαδικασία καθώς και ένα παράδειγμα κανονικής έκφρασης φαίνεται στο πρόγραμμα 5.3.7.

```
>>>sequence="QAQITGRPEWIWLALGTALMGMGTLYFLVKGMGVSDPDA";
>>>import re # εισαγωγή των σχετικών συναρτήσεων
>>>expression="W[IA].[LA]{3}[^T]TA[^T][MG]{2,4}.*LV"; #κανονική έκφραση
>>>com=re.compile(expression); #Μεταγλώττιση
>>>result=com.search(sequence); #Αναζήτηση
>>>#Εμφάνιση του σημείου έναρξης, τέλους και της πλήρους ακολουθίας
>>>print result.start( ), result.end( ), sequence[result.start( ):result.end( )];
9 29 WIWLALGTALMGMGTLYFLV
```

Πρόγραμμα 5.3.7

Στο πρόγραμμα αυτό αναζητούμε το μοτίβο που αναλύεται στο Σχήμα 5.1.



Σχήμα 5.1: Παράδειγμα μοτίβου

Συνεπώς το πρόγραμμα εντοπίζει την έκφρασή μας και τυπώνει το σημείο αρχής, τέλους και το πλήρες τμήμα. Η ακολουθία εισόδου ήταν η QAQITGRPEWIWLALGTALMGMGTLYF LVKGMGVSDPDA και η έκφρασή μας αρχίζει από την $9^{η}$ θέση (ξεκινώντας από το 0) τελειώνει στην $29^{η}$ θέση και το πλήρες τμήμα που ταίριαξε με τις προδιαγραφές που θέσαμε ήταν το WIWLALGTALMGMGTLYFLV.

5.3.4 *Εκτέλεση του BLAST μέσω της BioPython*

Η ευκολία που τα Bio μέρη των γλωσσών Perl και Python προσφέρουν είναι μεγάλη. Όπως παρουσιάστηκε και στην Perl, έτσι και στην Python είναι έυκολη η εκτέλεση του BLAST μέσω ενός σεναρίου-προγράμματός της. Η εκτέλεση του blastp στο πρόγραμμα 5.3.8 γίνεται μέσω της ιστοσελίδας της NCBI και η έρευνά μας περιορίζεται στην πρωτεϊνική βάση δεδομέων swissprot.

```
>>>from Bio.Blast import NCBIWWW
>>>from Bio import Fasta
>>>from sys import *

>>>query_file = open(argv[1]);
>>>result_file = "blast.xml";

>>>fasta = Fasta.Iterator(query_file);
>>>query = fasta.next( );
>>>query_file.close( );

>>>results_handle = NCBIWWW.qblast("blastp", "swissprot", query);
>>>blast_results = results_handle.read( );

>>>save_file = open(result_file, 'w');
>>>save_file.write(blast_results);
>>>save_file.close( );

>>>print "Results saved in : ", result_file;
```

Πρόγραμμα 5.3.8

5.4 *Σύνοψη και Συμπεράσματα*

Συνοψίζοντας τη μελέτη μας πάνω στις σεναριακές γλώσσες προγραμματισμού Perl και Python, πρέπει να αναφέρουμε πως είναι μεγάλης αξίας η βοήθεια που προσφέρουν στο χώρο των Βιοεπιστημών. Συντελούν στην αυτοματοποίηση, διευκόλυνση και επιτάχυνση ενεργειών και ερευνών, απαλλάσσοντας έτσι τον ερευνητή-χρήστη, από επίπονες χειρωνακτικές προσπάθειες. Ως σεναριακές γλώσσες, είναι εύχρηστες, ιδιαιτέρως από νέους προγραμματιστές, που με αυτές αποφεύγουν την πολυπλοκότητα άλλων γλωσσών προγραμματισμού.

Η γλώσσα Perl που πρωτοεμφανίσθηκε το 1987 είναι η δημοφιλέστερη σεναριακή γλώσσα στο χώρο των Βιοεπιστημών (σαφώς πιο δημοφιλής από την Python), κάτι που οφείλεται εν μέρει και στην χρονολογία εμφάνισής της. Η βιβλιογραφία που είναι διαθέσιμη για την Perl, μπορεί να χαρακτηριστεί ως εξαιρετική και οι χρήστες της στο χώρο των Βιοεπιστημών είναι αναρίθμητοι. Από την άλλη η γλώσσα Python εμφανίστηκε το 1991 και κερδίζει σταδιακά πολλούς φίλους-χρήστες στον χώρο αυτόν. Παρά τη δημοφιλία της Perl, η γλώσσα Python είναι πιο φιλική, καθώς οι κώδικες της Perl είναι συχνά δυσανάγνωστοι ή δυσνόητοι.

Οι δύο αυτές γλώσσες είναι συμβατές με όλα σχεδόν τα λειτουργικά συστήματα, καθώς και με άλλες γλώσσες προγραμματισμού, κάνοντας έτσι δυνατή τη μεταφορά και το συμμερισμό προγραμμάτων, με στόχο την καλύτερη έρευνα. Τα προγράμματά τους έχουν το μεγάλο πλεονέκτημα της εύκολης συντήρησης και τροποποίησης προγραμμάτων. Τα χαρακτηριστικά των γλωσσών αυτών απεικονίζονται στον Πίνακα 5.2.

Χαρακτηριστικά	Perl	Python
Φιλικότητα	√√	√√√
Δημοφιλία	√√√√	√√√
Διαθέσιμη Βιβλιογραφία	√√√√	√√
Φορητότητα – Συμβατότητα	√√√√	√√√√
Ευκολία Συντήρησης Προγραμμάτων	√√√	√√√

Πίνακας 5.2: Χαρακτηριστικά σεναριακών γλωσσών

6

Επίλογος

6.1 *Σύνοψη της Διπλωματικής εργασίας και Συμπεράσματα*

Η παρούσα διπλωματική εργασία αποτελεί μία τεχνική προσέγγιση του χώρου των Βιοεπιστημών, που προσπαθεί να περιγράψει όσο το δυνατόν καλύτερα στον αναγνώστη, ορισμένες εκ των σημαντικοτέρων τεχνικών διαχείρισης δεδομένων Βιοεπιστημών.

Αρχικά υποθέτοντας ότι ο αναγνώστης δεν κατέχει γνώσεις βιολογίας, περιγράφθησαν βασικές βιολογικές έννοιες και διαδικασίες, οι οποίες όχι μόνον είναι απαραίτητες για την κατανόηση των τεχνικών διαχείρισης βιολογικών δεδομένων, αλλά είναι και γενικότερου ενδιαφέροντος. Μέσα από αυτήν την περιγραφή φθάσαμε στην αναμενόμενη διαπίστωση της σημασίας και του ρόλου του DNA και των πρωτεϊνών στη ζωή. Εφόσον τα μόρια του DNA και των πρωτεϊνών μπορούν να αναπαρασταθούν σε μορφή ακολουθιών, αρχικά περιγράφθησαν τρόποι αποθήκευσης και αναπαράστασης τέτοιων ακολουθιών. Οι τρόποι αυτοί είναι πρότυπα XML που έχουν αναπτυχθεί από διάφορες επιστημονικές ομάδες, με στόχο την αναπαράσταση, αποθήκευση, ή τον συμμερισμό δεδομένων βιολογικών ακολουθιών.

Η πρώτη σημαντική τεχνική που περιγράφθηκε είναι η τεχνική της ευθυγράμμισης ακολουθιών. Η τεχνική αυτή συνίσταται στην κατακόρυφη στοίχιση δύο ή περισσότερων ακολουθιών, με στόχο τον εντοπισμό ομοιοτήτων ανάμεσα σε αυτές. Η ευθυγράμμιση μπορεί να γίνει επιτυχώς, είτε με τη βοήθεια του δυναμικού προγραμματισμού, είτε με τη βοήθεια ευριστικών μεθόδων-προγραμμάτων. Για αυτό το λόγο μελετήθηκαν και συγκρίθησαν τα δημοφιλή ευριστικά προγράμματα BLAST, FASTA και CLUSTAL.

Η επόμενη τεχνική που μελετήθηκε ήταν η τεχνικής της κατασκευής φυλογενετικών δενδρών. Τα δένδρα αυτά απεικονίζουν εξελικτικές σχέσεις-αποστάσεις, οι οποίες μπορεί να έχουν έρθει σε γνώση μας από την τεχνική ευθυγραμμίσεως ακολουθιών. Με βάση κάποιον πίνακα αποστάσεων οι δύο δημοφιλέστεροι και απλούστεροι αλγόριθμοι κατασκευής φυλογενετικών δενδρών, UPGMA και Neighbor-Joining, περιγράφθησαν πρακτικά και θεωρητικά. Επίσης έγινε ανάλυση του δημοφιλέστερου λογισμικού πακέτου φυλογενετικής ανάλυσης, PHYLIP.

Τέλος, παρουσιάστηκαν οι δύο, πλέον εύχρηστες στο χώρο των Βιοεπιστημών, σεναριακές γλώσσες προγραμματισμού, Perl και Python, οι οποίες στοχεύουν στη διευκόλυνση, επιτάχυνση και αυτοματοποίηση ενεργειών. Ιδιαίτερη έμφαση δόθηκε στην αναζήτηση μοτίβων συγκεκριμένου βιολογικού ενδιαφέροντος, με τη βοήθεια των κανονικών εκφράσεων.

6.2 Μελλοντικές Επεκτάσεις

Ο συγγραφέας πιστεύει πως τα κεφάλαια της σύγκρισης ακολουθιών (κεφάλαιο 3), των εξελικτικών δενδρών (κεφάλαιο 4) και της χρήσης σεναριακών γλωσσών (κεφάλαιο 5) καλύθηκαν επαρκώς, χωρίς να αποτρέπει το μελλοντικό ερευνητή από το να εμβαθύνει περαιτέρω (μελέτη άλλων προγραμμάτων, αλγορίθμων ή σεναριακών γλωσσών). Η πρόταση του συγγραφέα είναι ο μελλοντικός ερευνητής να ασχοληθεί με θέματα όπως:

- Εντοπισμός μοτίβων σε βιολογικές ακολουθίες στηριζόμενος σε μάθηση μηχανής, νευρωνικά δίκτυα, πιθανοτικές γραμματικές ή Μαρκοβιανά μοντέλα. Τα μοτίβα αυτά είναι μεγάλης βιολογικής σημασίας όπως έχει αναφερθεί και προηγουμένως, καθώς μπορεί να αποτελούν θέσεις στις οποίες είναι δυνατή η προσκόλληση μορίων φαρμάκων για την αντιμετώπιση ασθενειών, ή μπορεί να είναι τα ίδια τα μοτίβα η αιτία εμφάνισης ασθενειών.
- Καθορισμός τρισδιάστατων δομών με βάση τις ακολουθίες. Όσον αφορά το RNA η προσέγγιση μπορεί να γίνει σε δύο διαστάσεις, ενώ όσον αφορά τις πρωτεΐνες πρέπει να εξεταστεί η τρισδιάστατη δομή. Η μεγαλύτερη αποθήκη δομών πρωτεϊνών PDB καταγράφει για κάθε πρωτεΐνη τις καρτεσιανές συντεταγμένες καθενός εκ των ατόμων που την αποτελούν. Το γραφικό πακέτο Rasmol μπορεί να παρουσιάσει την τρισδιάστατη δομή μιας πρωτεΐνης σε μορφή φιλική προς το χρήστη και εύκολη στην παρατήρηση. Η EBI έχει επίσης αναπτύξει μία βάση δεδομένων με σκίτσα (cartoons) για κάθε πρωτεΐνη.
- Κατανόηση κανονισμών λειτουργίας του κυττάρου. Προκειμένου να κατανοήσουμε τη λειτουργία του κυττάρου θέλουμε να προσομοιώσουμε τη λειτουργία των γονιδίων.

Αυτό γίνεται είτε με ένα διακριτό μοντέλο όπου έχουμε μια διαδικασία διεργασίας σε μορφή προγράμματος, είτε με συνεχές μοντέλο όπου εάν έχω αλληλεπίδραση n γονιδίων θα πρέπει να επιλύσω n διαφορικές εξισώσεις. Ο Somoguyi έχει προτείνει μια ενδιαφέρουσα μέθοδο για προσομοίωση και μοντελοποίηση της λειτουργίας του γονιδίου. Επίσης το E-CELL είναι ένα φιλόδοξο ιαπωνικό σχέδιο που στοχεύει στην προσομοίωση κυττάρων χρησιμοποιώντας στοχαστικά συστήματα μη γραμμικών διαφορικών εξισώσεων.

- Προσδιορισμός λειτουργίας πρωτεϊνών και καθορισμός μεταβολικών μονοπατιών. Στόχος σε αυτήν την περίπτωση είναι η κατανόηση των σχολίων των ερευνητών και η κατασκευή βάσεων δεδομένων που αντιπροσωπεύουν γράφους. Απαραίτητα είναι τα προγράμματα που θα απεικονίσουν γραφικώς τις πρωτεϊνες όπως το Ecocyc του Karp. Πακέτα που καθορίζουν μεταβολικά μονοπάτια είναι το Metacyc του Karp, καθώς και το ιαπωνικό KEGG.
- Συναρμολόγηση τμημάτων DNA. Η διαδικασία αυτή γίνεται μέσω ηλεκτρονικών υπολογιστών και εμπεριέχει κινδυνο λαθών καθώς το DNA έχει επαναλαμβανόμενες περιοχές και άρα κάποιο κομμάτι μποεί να ανήκει σε πολλές περιοχές. Εταιρείες όπως η Celera λαμβάνουν πολλά μικρά τμήματα DNA με τη μέθοδο του πυροβόλου όπλου και στη συνέχεια προσπαθούν να τα συναρμολογήσουν.

Τέλος πρέπει να σημειωθεί πως για οποιοδήποτε λάθος ή ασάφεια που πιθανόν να ταλαιπώρησαν τον αναγνώστη, αποκλειστική ευθύνη φέρει ο συγγραφέας.

7

Βιβλιογραφία

[ΚΜΠ+98] Α. Καψάλης, Ι.Ε. Μπουρμπουχάκης, Β. Περάκη, Σ. Σαλαμαστράκης-Βιολογία Γενικής Παιδείας Β τάξης Ενιαίου Λυκείου, Οργανισμός Εκδόσεων Διδακτικών Βιβλίων, 1998.

[Περ06] Δέσποινα Α. Περούλη-Θέματα Διαχείρισης Δεδομένων για Εφαρμογές Βιοεπιστημών, Διπλωματική Εργασία Εθνικού Μετσόβιου Πολυτεχνείου, Ιούλιος 2006.

[ΠΚΚ96] Θ.Α Παταργιάς, Κ. Κομητοπούλου, Σ. Κουγιανού–Εισαγωγή στη Βιολογία Φαρμακευτικού τμήματος Πανεπιστημίου Αθηνών, 1996.

[Ace07] http://www.accessexcellence.org.

[Adb07] Advanced Bioperl, available at http://stein.cshl.org/genome_informatics/ bioperl_03/ Bioperl-2.pdf.

[AJS+04] Zayed I. Albertyn, Tan Ka Ju, Wong Chee San, M. Ramachandran and John Iskandar-Reconstructing Alignments Based on Patterns Inherent in Biological Data: A Qualitative Comparison, 2004.

[Alt90] Stephen F.Altschul-Basic Local Alignment Search Tool, Feb 1990.

[AMS97] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman–Gapped BLAST and PSI-BLAST: a new generation of protein database search programs-Nucleic Acids Research 1997, Vol. 25. Oxford

University Press 1997.

[Bif07] http://www.bioinformatica.crs4.org/help/analysis/fasta.

[Bis07] http://www.bioinfo.se/kurser/swell/blasta-fasta.shtml.

[Bld07] THE NEW BIOLOGY, LIKE FIRE, CAN LIGHT UP THE NIGHT OR BURN DOWN THE HOUSE, BILD 10 2007, available at www.biology. ucsd.edu/classes/bild10.SP07/BILD10%20week%201.pdf.

[BM06] Rodolfo Bezerra Batista and Alba Cristina Magalhaes Alves de Melo- A Parallel Strategy for Local Biological Sequence Alignment in User - Restricted Memory Space, 2006.

[Bml07] http://www.bioml.com/BIOML.

[Bph07] Building phylogenetic trees, available at www.cs.pu.edu.tw/~yawlin/docs/ 2004- 0514-Phylo-Trees.ppt.

[Bpl07] Bioinformatics explained: Phylogenetics, August 2005, available at www.clcbio.com/sciencearticles/BE-phylogenetics.pdf.

[Bpr07] Running Blast Through Perl, available at www.bios.niu.edu/johns/ bioinform/perl_blast.ppt.

[Bps07] http://bioweb.pasteur.fr/seqanal/ interfaces/.

[Bsm07] www.labbook.com.

[Cas06] M. Caselle-Theoretical Physics Methods for Computational Biology, April 2006.

[Cbi07] http://www.cbi.pku.edu.cn/Doc/search/seqsearch.

[Cha07] Alexander Chan-An Analysis of Pairwise Sequence Alignment Algorithm Complexities: Needleman-Wunsch, Smith-Waterman, FASTA, BLAST and Gapped BLAST.

[Che06] www.ch.embnet.org.

[Cml07] http://www.cellml.org.

[Coh04] J. Cohen-Bioinformatics, An Introduction for Computer Scientists. Brandeis University 2004.

[CP05] Mark Craven and David Page–Pairwise Sequence Alignment - lecture3 September 2005, available at www.biostat.wisc.edu/bmi576.html.

[CRZ03] A.B. Chaudhri, A. Rashid, R. Zicari-XML Data Management: Native XML and XML-Enabled Database Systems, March 2003.

[CS03] Sophie Coon and Michel Sanner-Python For Structural Bioinformatics, February 2003.

[DAW04] J. P. Davis, S. Akella1, P. H. Waddell-Accelerating Phylogenetics Computing on the Desktop: Experiments with Executing UPGMA in Programmable Logic, September 2004.

[DBL+00] Shawn Delaney, Greg Butler, Clement Lam, Larry Thiel-Three Improvements to the BLASTP Search of Genome Databases, 2000.

[Dik06] www.diku.dk/~pawel/comp-bio/ev_trees/intro/upgma_ex.html.

[Ebc06] www.ebi.ac.uk./2can/home.html.

[Ebi07] www.ebi.ac.uk.

[Emb06] http://www.ebi.ac.uk/embl/.

[Gam07] www.bioxml.org/Projects/game/game0.1.html.

[Gau05] Daniel Gautheret-Notions de Perl, 2005, available at rna.igmors.u-psud.fr/gautheret/cours/perl.pdf.

[Gen06] http://www.ncbi.nlm.nih.gov/Genbank/index.html.

[Han04] Theodor Hanekamp-Protein sequence alignments, University of Wyoming, Spring 2002, updated 2004, available at www-ibmc.u-strasbg.fr/upr9002/westhof/presentation/Prot_seq_ali_Cours2_2004.pdf.

[Hob03] Asger Hobolth-Computational gene prediction, Bioinformatics Research Centre, University of Aarhus, March 2003, available at www.daimi.au.dk/~schauser/bioinformatik_E03/lectures_E03/genepredict.pdf.

[Hsi07] Will Hsiao-An Introduction to Perl for Bioinformatics – Part 2 Lecture 6.1, available at bioinformatics.ca/workshop_pages/bioinformatics/day6/6_1.pdf

[Liu07] Jun Liu-Biological Sequence Analysis and Motif Discovery, Introductory Overview Lecture Joint Statistical Meetings 2001, Atlanta, available at www.people.fas.harvard.edu/~junliu/sequence_analysis.pdf.

[Leh07] Heikki Lehvaslaiho-Introduction to Perl and BioPerl, Institut Pasteur Tunis, SANBI, March 2007.

[Lin07] http://www.linuxjournal.com/article/3882.

[Mag07] http://www.mged.org/Workgroups/MAGE/introduction.html.

[Mus06] Gabe Musso-Alignment (Needleman-Wunsch, Smith-Waterman) CSC2427 Course Notes, February 2006, available at www.cs.toronto.edu/~brudno/csc2427/Lec7Notes.pdf.

[Ncb06] www.ncbi.nlm.nih.gov.

[Neoc01] What is NeoCore XML Management System?, Neocore Inc. Release 1.0, June 2001.

[Pab07] http://bioweb.pasteur.fr/seqanal/interfaces/protdist-simple.html.

[Pas07] www.pasteur.fr.

[Per07] www.perl.com.

[Pev05] Jonathan Pevsner, BLAST:Basic local alignment search tool, September 2005.

[Pev06] pevsnerlab.kennedykrieger.org/bioinfo_course.htm.

[Phv07] Overview of Phylogeny, available at robotics.stanford.edu/~serafim/cs262/Spring2003/Slides/Lecture18.pdf.

[Phy07] http://evolution.genetics.washington.edu/phylip.html.

[Pir06] http://pir.georgetown.edu/.

[Pyt07] www.python.org.

[Rmg07] http://rss.acs.unt.edu/Rdoc/library/RMAGEML/doc.

[Rob86] M.B.V Roberts-Biology, A Functional Approach, Fourth Edition, 1986.

[Ros02] Guido Van Rossum-Introduction to Python, Zope Corporation, January 2002.

[SBB+02] Jason E. Stajich, David Block, Kris Boulez, Steven E. Brenner, Stephen A. Chervitz, Chris Dagdigian, Georg Fuellen, James G.R. Gilbert, Ian Korf, Hilmar Lapp, Heikki Lehvaslaiho, Chad Matsalla, Chris J. Mungall, Brian I. Osborne, Matthew R. Pocock, Peter Schattner, Martin Senger, Lincoln D. Stein, Elia Stupka, Mark D. Wilkinson, and Ewan Birney-The Bioperl Toolkit: Perl Modules for the Life Sciences, 2002.

[Sbm07] http://www.cds.caltech.edu/erato/index.html.

[Sch07] M.J. Schilstra-SBML and CellML, Standard XML-based markup languages for models of biochemical systems, available at http://sbml.org/ documents/presentations/isb-2005/sbml-and-cellml.ppt.

[Sci07] http://www.the-scientist.com/2005/8/29/21/1/.

[Seq07] Sequence Analysis Model and Implementation, available at http://oldsite. capital.edu/acad/as/csac/Comp_Bio/Analysis_module/model.pdf.

[SL07] Katja Schuerer and Catherine Letondal-Python Course in Bioinformatics, 2007.

[Spi02] Joseph Spitzner-BSML Overview, I3C Technical Committee Meeting, VP Technology, LabBook, July 2002, available at http://www.bsml.org/i3c/ docs/I3C_BSML_July2002.ppt.

[Spr06] http://www.expasy.ch/sprot/.

[Sta01] Jason Stajich-Using Perl for Bioinformatics, Module 1: Learning Perl, July 2001.

[Str07] Lena Strömbäck-XML representations of pathway data: a comparison.

[Tis01] James Tisdall-Beginning Perl for Bioinformatics, First Edition, October 2001.

[Tis03] James Tisdall-Mastering Perl for Bioinformatics, First Edition, September 2003.

[Top07] http://www.topogen.com/sbir/rfc.html.

[Tui06] Jarno Tuimala-A primer to phylogenetic analysis using the PHYLIP package Fifth Edition, July 2006.

[Usp07] Using Perl for Bioinfromatics, available at www.osc.edu/hpc/training/bioperl/perl_bioinf_0411_pdf.pdf.

[Wer07] Rasmus Wernersson-Virtual Ribosome - a comprehensive DNA translation tool with support for integration of sequence feature annotation.

[WFR96] P. S. Walsh, N. J. Fildes and Rebecca Reynolds-Sequence analysis and characterization of stutter products at the tetranucleotide repeat locus vWA, 1996.

[Wik07] http://el.wikipedia.org.

[Xin07] Yun Xing-Sequence Alignment Algorithm, School of Electronics Engineering and Computer Science, Peking University, Beijing, available at icl.pku.edu.cn/yujs/papers/pdf/SeqAli.pdf.

[Xms07] http://xml.sys-con.com.

[YSB+05] Joe W. Yeol, W. Samarrai, I. Barjis, and Y. S. Ryu-Data-flow Diagram Representation of Biological Process: DNA, RNA, and Protein, 2005.