ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2011/2012

Εισαγωγή

Η ανακάλυψη πως τα νουκλεϊκά οξέα αποτελούνται από ακολουθίες συγκεκριμένων νουκλεοτιδίων, καθώς επίσης πως οι πρωτεΐνες αποτελούνται από ακολουθίες συγκεκριμένων αμινοξέων, αποτέλεσε τεράστια επιτυχία στον τομέα της βιολογίας. Γρήγορα έγινε σαφές πως συγκρίνοντας τις ακολουθίες αυτές μεταξύ διαφορετικών όντων μπορούσαμε να εξάγουμε πληροφορίες για το κατά πόσο τα όντα αυτά συγγενεύουν μεταξύ τους. Όσο περισσότερο μοιάζουν οι συγκεκριμένες ακολουθίες, τόσο πιο πιθανό είναι τα όντα αυτά να έχουν εξελιχθεί από κάποιον κοινό πρόγονο.

Τα 4 βασικά συστατικά στοιχεία ενός νουκλεϊκού οξέος συμβολίζονται με τα γράμματα A, C, G και T (για Adenine, Cytosine, Guanine και Thymine αντίστοιχα). Ένα νουκλεϊκό οξύ αποτελείται λοιπόν από μία σειρά από αυτές τις βάσεις. Ας υποθέσουμε πως έχουμε τις παρακάτω 2 ακολουθίες:

```
X: C G G G T A T C C A A
Y: C C C T A G G T C C C A
```

Το ερώτημα είναι αν θα μπορούσε η μια από αυτές να προκύψει από την άλλη. Για παράδειγμα, αν (για τον οποιοδήποτε λόγο) η βάση G στην ακολουθία X είχε αντικατασταθεί από την βάση C στην ακολουθία Y, τότε οι δύο αυτές ακολουθίες θα έμοιαζαν πολύ περισσότερο. Επιπλέον όμως, η ακολουθία Y είναι πιο μεγάλη. Μήπως λοιπόν προστέθηκε και μια βάση ακόμα κατά την διαδικασία της εξέλιξης;

Σκοπός της ευθυγράμμισης τέτοιων ακολουθιών (sequence alignment) είναι να πετύχουμε το καλύτερο δυνατό ταίριασμα της μιας ακολουθίας με την άλλη. Η ευθυγράμμιση πρέπει να ακολουθεί τους παρακάτω κανόνες:

- Όλα τα σύμβολα των δύο ακολουθιών πρέπει να υπάρχουν στην ευθυγράμμιση και μάλιστα με την ίδια σειρά με την οποία εμφανίζονται στις ακολουθίες.
- Κάθε σύμβολο μιας ακολουθίας πρέπει να ευθυγραμμιστεί με ένα μόνο σύμβολο της άλλης ακολουθίας.
- Ένα σύμβολο μπορεί να ευθυγραμμιστεί με ένα κενό (‘-‘).
- Δύο κενά δεν επιτρέπεται να είναι ευθυγραμμισμένα.

Ένα παράδειγμα ευθυγράμμισης των παραπάνω ακολουθιών θα ήταν το εξής:

```
X: C G G G T A - - T - C C A A
Y: C C C - T A G G T C C C - A
```

Παρατηρήστε ότι οι παραπάνω κανόνες δεν απαιτούν ένα σύμβολο να είναι οπωσδήποτε ευθυγραμμισμένο με το κενό ή με το ίδιο ακριβώς σύμβολο. Στο παραπάνω παράδειγμα έχουμε χρωματίσει με πράσινο τις ευθυγραμμίσεις όπου αυτό ισχύει. Με κόκκινο έχουμε χρωματίσει τις ευθυγραμμίσεις όπου ένα σύμβολο έχει ευθυγραμμιστεί με άλλο σύμβολο.

Όπως γίνεται κατανοητό, υπάρχουν πολλές ευθυγραμμίσεις που θα μπορούσαμε να έχουμε για τις συγκεκριμένες ακολουθίες. Εκτός από την παραπάνω ευθυγράμμιση, 2 επιπλέον παραδείγματα θα μπορούσαν να είναι τα εξής:

```
X: C G G G T A - - - T C C A A
Y: C C - - C T A G G T C C C A

X: C - G G G T A - - T C C A A
Y: C C - - C T A G G T C C C A
```

Εύλογα λοιπόν γεννιέται το ερώτημα ποια είναι η καλύτερη δυνατή ευθυγράμμιση δύο ακολουθιών. Η απάντηση δεν είναι εύκολη, καθώς συνήθως οι ακολουθίες αυτές για πραγματικούς ζωντανούς οργανισμούς αποτελούνται από τουλάχιστον δεκάδες χιλιάδες σύμβολα η καθεμία. Η εύρεση της βέλτιστης ακολουθίας «στο χαρτί» είναι κατά συνέπεια αδύνατη. Όπως σωστά φανταστήκατε, η αξιοποίηση της υπολογιστικής ισχύος των σύγχρονων υπολογιστών δίνει την λύση.

<u>Ο Αλγόριθμος Smith-Waterman</u>

Για να λύσουμε το παραπάνω πρόβλημα χρειαζόμαστε φυσικά τον κατάλληλο αλγόριθμο. Ο πλέον χρησιμοποιούμενος αλγόριθμος προτάθηκε το 1981 από τους Temple F. Smith και Michael S. Waterman και είναι γνωστός με το όνομα Smith-Waterman [1].

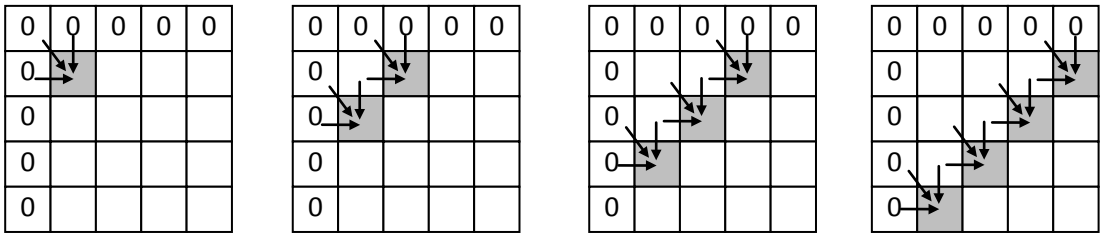
Βασικό στοιχείο στον συγκεκριμένο αλγόριθμο είναι η δημιουργία ενός πίνακα *H*, γνωστού ως «πίνακα βαθμολόγησης» (scoring matrix). Ουσιαστικά, στον πίνακα αυτό αποθηκεύουμε βαθμολογία για κάθε πιθανή ευθυγράμμιση δύο ακολουθιών. Αν υποθέσουμε πως οι δύο ακολουθίες X, Y έχουν μήκος *m* και *n* αντίστοιχα, τότε ο πίνακας *H* έχει διαστάσεις (*m*+1)×(*n*+1) και κατασκευάζεται βάσει των παρακάτω κανόνων:

- $H(i, 0) = 0,\ 0 \le i \le m$ (Στοιχεία πρώτης στήλης είναι 0)
- $H(0, j) = 0,\ 0 \le j \le n$ (Στοιχεία πρώτης γραμμής είναι 0)
- if $X_i = Y_j$ then $w(X_i, Y_j) = w(Match)$
- if $X_i \neq Y_j$ then $w(X_i, Y_j) = w(Mismatch)$
- $$H(i,j) = max \begin{Bmatrix} 0 & & \\ H(i-1,j-1) + w(X_i, Y_j) & Match/Mismatch & \\ H(i-1,j) + w(X_i, -) & Deletion & \\ H(i,j-1) + w(-, Y_j) & Insertion & \end{Bmatrix} \begin{matrix} \\ 1 \le i \le m \\ 1 \le j \le n \end{matrix}$$

Κατά συνέπεια, ο πίνακας *H* κατασκευάζεται αναδρομικά. Κάθε κελί χρειάζεται τους 3 γείτονες που «προηγούνται» από αυτό για να υπολογιστεί. Η συνάρτηση *w*() επιστρέφει έναν αριθμό, ο οποίος βαθμολογεί το ταίριασμα δύο συμβόλων στις δύο ακολουθίες. Ας πάρουμε για παράδειγμα τον κλάδο $H(i-1, j-1) + w(X_i, Y_j)$. Ο όρος $H(i-1, j-1)$ αντιστοιχεί στην βαθμολογία της ακολουθίας που έχει ήδη υπολογιστεί. Η συνάρτηση *w*() θα επιστρέψει έναν αριθμό, ανάλογα με το πως το σύμβολο X_i ταιριάζει με το σύμβολο Y_j. Αν τα σύμβολα είναι ίδια συνήθως επιστρέφεται ένας θετικός αριθμός. Αυτό όμως δεν σημαίνει απαραίτητα ότι ο αριθμός αυτός είναι ίδιος για τα 4 βασικά συστατικά στοιχεία ενός νουκλεϊκού οξέος. Έτσι, αν και οι δύο ακολουθίες έχουν το σύμβολο C στις συγκεκριμένες θέσεις, τότε μπορεί να επιστρέφεται π.χ. ο αριθμός 12, ενώ αν έχουν το G τότε μπορεί να επιστρέφεται ο αριθμός 5. Αυτό συμβαίνει γιατί η ταύτιση για συγκεκριμένα σύμβολα μπορεί να θεωρηθεί πιο σημαντική από την ταύτιση για άλλα σύμβολα. Αν οι δύο ακολουθίες δεν ταιριάζουν στην

συγκεκριμένη θέση, τότε συνήθως επιστρέφεται ένας αρνητικός αριθμός, μειώνοντας έτσι την συνολική βαθμολογία. Αντίστοιχα, για τους δύο τελευταίους κλάδους βαθμολογείται η ταύτιση των συμβολοσειρών όταν προσθέσουμε το κενό σύμβολο σε μία από τις δύο ακολουθίες. Το ποιούς ακριβώς αριθμούς θα επιστρέφει η συνάρτηση *w*() δεν το καθορίζει ο αλγόριθμος Smith-Waterman. Είναι στην ευχέρεια του καθένα να τους καθορίσει ανάλογα με τις ανάγκες της εφαρμογής.

Ο αλγόριθμος αυτός ακολουθεί το μοντέλο του «κυματομέτωπου» (wave front). Αυτό αποτελεί πολύ σημαντικό στοιχείο για την παραλληλοποίηση του αλγόριθμου. Θα πρέπει να βεβαιώνεστε πως κατά τους υπολογισμούς κάθε κελιού έχουν υπολογιστεί ήδη τα κελιά που χρειάζονται. Με λίγα λόγια, στον συγκεκριμένο αλγόριθμο υπάρχουν «εξαρτήσεις δεδομένων». Σχηματικά, ο αλγόριθμος θα ακολουθούσε τα εξής βήματα (για έναν μικρό πίνακα):



Σε κάθε βήμα εκτέλεσης μπορούν να υπολογιστούν τα στοιχεία του πίνακα Η που έχουν χρωματιστεί γκρι. Βεβαίως οι υπολογισμοί θα συνεχίζονταν για 3 ακόμα βήματα, μέχρι και το τελευταίο στοιχείο του πίνακα.

Το τελευταίο βήμα του αλγόριθμου είναι να βρεθεί η καλύτερη ακολουθία. Προφανώς, αυτή θα έχει την μεγαλύτερη βαθμολογία. Κατά συνέπεια, θα πρέπει να αναζητηθεί η μέγιστη βαμολογία στον πίνακα *H* και στην συνέχεια να ακολουθήσουμε ανάποδα την διαδρομή από την οποία προέκυψε η μέγιστη βαθμολογία (back tracking). Με αυτόν τον τρόπο ολοκληρώνεται η εύρεση της βέλτιστης ευθυγράμμισης.

Αναφορές

[1] Temple F. Smith and Michael S. Waterman. "Identification of Common Molecular Subsequences". Journal of Molecular Biology 147: 195–197, 1981.

Ζητούμενα της άσκησης

Στα πλαίσια της άσκησης θα σας δοθεί μια απλοποιημένη έκδοση του αλγόριθμου Smith-Waterman γραμμένη σε C++. Σκοπός σας θα είναι να παραλληλοποιήσετε το πρόγραμμα και να δημιουργήσετε δύο διαφορετικές παράλληλες εκδόσεις:

1) Μια με χρήση POSIX Threads
2) Μια με χρήση OpenMP

Είσαστε ελεύθεροι να παραλληλοποιήσετε τον κώδικα με καθένα από τα παραπάνω πρότυπα με όποιον τρόπο θέλετε. Ωστόσο, θεωρείται αυτονόητο πως η υλοποίηση θα πρέπει να είναι σωστή (να δίνει **πάντα** τα ίδια αποτελέσματα με την ακολουθιακή έκδοση του προγράμματος) και πως στην αναφορά που θα παραδώσετε θα πρέπει να αιτιολογήσετε γιατί παραλληλοποιήσατε με τον συγκεκριμένο τρόπο την εφαρμογή.

Όπου χρειάζεται, οι υλοποιήσεις σας θα πρέπει να είναι παραμετροποιημένες ως προς το πλήθος των νημάτων. Μπορείτε να προσθέσετε μια επιπλέον παράμετρο γραμμής εντολών στο πρόγραμμα που να δηλώνει με πόσα νήματα θα εκτελεστεί κάθε φορά το πρόγραμμα.

Για να μεταγλωττίσετε το πρόγραμμα που σας δίνεται θα πρέπει να εκτελέσετε την εντολή:

g++ –O0 –o SmithWaterman SmithWaterman.cpp

Η εντολή αυτή θα μεταγλωττίσει το πρόγραμμα χωρίς βελτιστοποιήσεις (-O0) και θα παράξει ένα εκτελέσιμο αρχείο με το όνομα SmithWaterman.

Για να το τρέξετε, χρησιμοποιήστε την εντολή:

./SmithWaterman 0.33 1.33 SequenceAFile SequenceBFile 1000

Τα αρχεία SequenceAFile και SequenceBFile είναι απλά αρχεία κειμένου που περιέχουν τις ακολουθίες των συμβόλων που θέλουμε να ευθυγραμμίσουμε. Στον παρακάτω πίνακα αναφέρονται 6 αρχεία με ακολουθίες συμβόλων που θα σας δωθούν. Όταν θα δοκιμάζετε το πρόγραμμα σας να συγκρίνετε τις ακολουθίες που βρίσκονται στην ίδια σειρά του πίνακα.

Ακολουθία Α	**Ακολουθία Β**
LITMUS28i.gbk.txt	LITMUS28i-mal.gbk.txt
M13KE.gbk.txt	M13KO7.gbk.txt
Adenovirus-C.txt	Enterobacteria-phage-T7.txt

Τα τελικά αποτελέσματα που θα συμπεριλάβετε στην αναφορά θα πρέπει να αφορούν τις ακολουθίες της τελευταίας σειράς του πίνακα. Οι πρώτες δύο σειρές περιλαμβάνουν μικρότερες ακολουθίες που μπορείτε να χρησιμοποιήσετε όσο αναπτύσσετε τα προγράμματα σας για να ελέγχετε ότι λειτουργούν σωστά.

Αφού παραλληλοποιήσετε το πρόγραμμα θα προβλέψετε ώστε να δίνετε άλλη μια παράμετρο κατά την εκτέλεση του προγράμματος, π.χ., η εντολή:

./SmithWaterman 0.33 1.33 SequenceAFile SequenceBFile 1000 4

θα σημαίνει πως το πρόγραμμα θα εκτελεστεί με 4 νήματα, ενώ η εντολή:

./SmithWaterman 0.33 1.33 SequenceAFile SequenceBFile 1000 8

πως θα εκτελεστεί με 8 νήματα.

Ένας ακόμη σκοπός της εργασίας είναι να σας φέρει σε επαφή και με άλλα εργαλεία ανάπτυξης. Για τον λόγο αυτό, θα χρησιμοποιήσετε και έναν δεύτερο compiler. Συγκεκριμένα, θα χρησιμοποιήσετε τους compilers της Intel, οι οποίοι διανέμονται δωρεάν για ακαδημαϊκή χρήση. Θα πρέπει επομένως να μεταβείτε στην διεύθυνση:

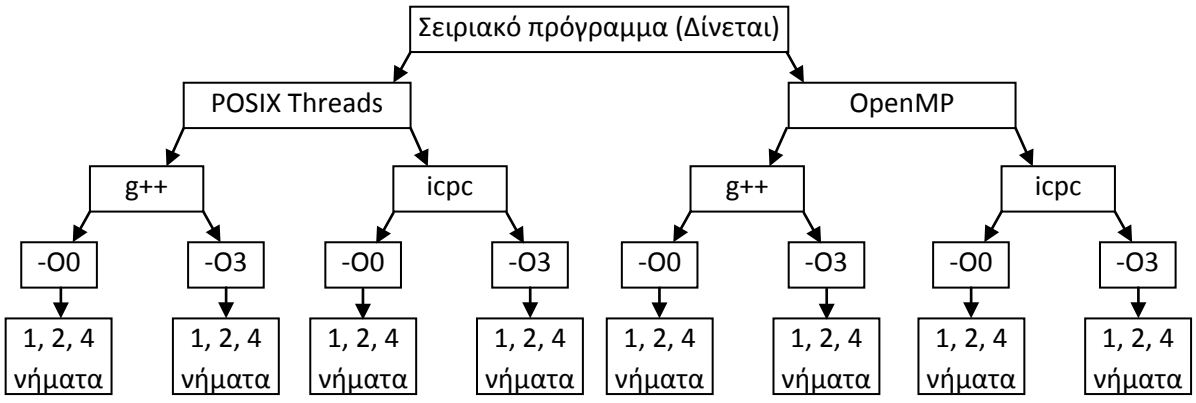
http://software.intel.com/en-us/articles/non-commercial-software-development

και να επιλέξετε το "Accept" (αφορά αποδοχή ότι θα χρησιμοποιήσετε τα συγκεκριμένα προϊόντα μόνο για μη εμπορικούς σκοπούς). Στην επόμενη σελίδα που θα εμφανιστεί επιλέγετε το "Intel® C++ Composer XE 2011 for Linux". Συμπληρώνετε τα απαραίτητα στοιχεία και στην συνέχεια κατεβάζετε το πρόγραμμα και το εγκαθιστάτε. Η αντίστοιχη εντολή για να μεταγλωττίσετε το πρόγραμμα με τον compiler της Intel είναι η "icpc". Οι υπόλοιπες παράμετροι είναι ίδιες με αυτές του "g++".

Συγκεντρωτικά, θα πρέπει στην αναφορά σας να παραθέσετε μετρήσεις και να σχολιάσετε τις παρακάτω περιπτώσεις:

1) Πρόγραμμα με POSIX Threads και με OpenMP.
2) Μεταγλώττιση και εκτέλεση προγράμματος με χρήση gcc και icpc.
3) Μεταγλώττιση και εκτέλεση προγράμματος χωρίς βελτιστοποιήσεις (-O0) και με μέγιστες βελτιστοποιήσεις (-O3).
4) Εκτέλεση κάθε περίπτωσης με 1, 2 και 4 νήματα τουλάχιστον. Αν το σύστημα σας διαθέτει περισσότερα, ακόμα καλύτερα!

Για κάθε έναν από τους παραπάνω συνδυασμούς συμπεριλάβετε στην αναφορά σας πίνακες με τους χρόνους εκτέλεσης του βασικού αλγόριθμου (υπάρχει έτοιμο στον κώδικα που σας δίνετε) και διαγράμματα της χρονοβελτίωσης. Συγκεντρωτικά, όλες οι περιπτώσεις που θα πρέπει να συμπεριλάβετε φαίνονται στο παρακάτω σχήμα:



Διαδικαστικά

Η εργασία θα πρέπει να γίνει σε ομάδες των 3 ή 4 ατόμων **ακριβώς**. Δεν θα γίνει δεκτή ομάδα με λιγότερα ή περισσότερα άτομα **για κανέναν λόγο**. Αν δεν συμπληρώσετε ομάδα 3 ή 4 ατόμων, τότε θα προστεθούν άτομα στην ομάδα σας από εμάς.

Ένα άτομο από κάθε ομάδα θα αναλάβει να δηλώσει την ομάδα του μέχρι την **Τετάρτη, 11/04/2012 και ώρα 23:59:59**. Το άτομο αυτό θα είναι επίσης υπεύθυνο για όλη την επικοινωνία της ομάδας μαζί μας, καθ' όλη την διάρκεια του εξαμήνου και μέχρι την παράδοση της άσκησης. Η ομάδα θα δηλωθεί μέσω e-mail στην διεύθυνση iev@hpclab.ceid.upatras.gr. Για την ευκολότερη ταξινόμηση από την μεριά μας και την δυνατότητα αυτόματης προώθησης, το e-mail θα πρέπει να έχει τον εξής τίτλο (χωρίς κενά στο τμήμα [ParPro11-12]):

[ParPro11-12] Δήλωση ομάδας

Το περιεχόμενο του e-mail θα πρέπει να είναι ο Α.Μ. και το ονοματεπώνυμο των τριών μελών της ομάδας. **Κάθε επικοινωνία μαζί μας θα πρέπει να απευθύνεται στην προαναφερθείσα διεύθυνση e-mail και ο τίτλος να ξεκινάει με [ParPro11-12]**.

Παραδοτέα

Η άσκηση θα πρέπει να παραδοθεί μέχρι την **Παρασκευή, 01/06/2012 και ώρα 23:59:59**. Τα παραδοτέα σας θα αποτελούνται από δύο τμήματα:

1) Μια γραπτή αναφορά, την οποία θα παραδώσετε στις κυρίες Ελένη Μιχαλά και Φωτεινή Γαριδάκη στο Εργαστήριο Πληροφοριακών Συστημάτων Υψηλών Επιδόσεων, το οποίο βρίσκεται στο Κτίριο Β', στον δεύτερο όροφο (στο γραφείο του κ. Παπαθεοδώρου).
   Στην αναφορά **δεν** θα πρέπει να περιλαμβάνεται επεξήγηση του ακολουθιακού αλγόριθμου Smith-Waterman (ο οποίος σας δώθηκε εδώ). Επικεντρωθείτε στην επεξήγηση της παραλληλοποίησης που κάνατε με κάθε πρότυπο, στις μετρήσεις σας και στα διαγράμματα που θα προσθέσετε. Σχολιάστε τις διαφορές που βλέπετε για την χρήση διαφορετικών compilers και τις επιπτώσεις της χρήσης βελτιστοποιήσεων στον χρόνο εκτέλεσης της εφαρμογής και (κυρίως) στην χρονοβελτίωση.
2) Η αναφορά σας σε ηλεκτρονική μορφή και ο κώδικας της άσκησης, τα οποία θα παραδωθούν μέσω e-mail στην διεύθυνση iev@hpclab.ceid.upatras.gr. Ο τίτλος θα πρέπει να έχει την μορφή:

   [ParPro11-12] Παράδοση Άσκησης-Ομάδα XX

   Ο αριθμός της ομάδας σας θα σας γνωστοποιηθεί μετά την ολοκλήρωση των δηλώσεων των ομάδων και θα πρέπει να αντικαταστήσετε τα "XX" στον παραπάνω τίτλο με τον αριθμό της ομάδας.

Η άσκηση θα μετρήσει στην τελική βαθμολογία ως εξής:

1) 30% για όσους την παραδώσουν τον Ιούνιο. Το υπόλοιπο 70% θα είναι η τελική εξέταση.
2) 20% για όσους την παραδώσουν τον Σεπτέμβριο (ακριβής ημερομηνία θα ανακοινωθεί στο μέλλον). Το υπόλοιπο 80% θα είναι η τελική εξέταση.