Ατομική Διπλωματική Εργασία

ΠΑΡΑΛΛΗΛΟΠΟΙΗΣΗ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΧΑΡΤΩΝ ΑΥΤΟ-ΟΡΓΑΝΩΣΗΣ ΚΑΙ ΧΡΗΣΗ ΤΟΥΣ ΓΙΑ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΣΗΜΕΙΑΚΩΝ ΝΟΥΚΛΕΟΤΙΔΙΚΩΝ ΠΟΛΥΜΟΡΦΙΣΜΩΝ

Άριστος Αριστοδήμου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Παραλληλοποίηση και βελτιστοποίηση χαρτών αυτό-οργάνωσης και χρήση τους για κατηγοριοποίηση σημειακών νουκλεοτιδικών πολυμορφισμών

Άριστος Αριστοδήμου

Επιβλέπων Καθηγητής
Δρ. Κωνσταντίνος Παττίχη

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

Μάιος 2010

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Δρ. Παττίχη Κωνσταντίνο που με επέλεξε για την διπλωματική εργασία και για την επίβλεψη που μου έκανε καθ' όλη τη διάρκεια του έτους. Επίσης θα ήθελα να ευχαριστήσω τον κύριο Άθω Αντωνιάδη χωρίς τον οποίο αυτή η διπλωματική εργασία δεν θα μπορούσε να γίνει. Ο κύριος Άθως Αντωνιάδης πρότεινε το θέμα αυτής της διπλωματικής εργασίας και ήταν ο γενικός διαχειριστής του όλου έργου. Χάρη στη συνεργασία του με την Glaxo Smith Kline, η εταιρεία μας παραχώρησε τα δεδομένα για τη συγκεκριμένη εργασία, χωρίς τα οποία και πάλι δεν θα μπορούσαμε να υλοποιήσουμε αυτό το έργο. Τέλος θα ήθελα να ευχαριστήσω το συμφοιτητή και πολύ καλό μου φίλο, κύριο Λοΐζο Λοΐζου, για την άψογη συνεργασία που είχαμε, γιατί χωρίς τη συνεργασία που υπήρξε, δεν θα μπορούσε να ολοκληρωθεί το όλο έργο εντός των χρονικών πλαισίων που είχαμε.

Περίληψη

Η χαρτογράφηση του ανθρώπινου δεσοξυριβοζονουκλεϊνικού οξέος, προκάλεσε το ενδιαφέρον για την μελέτη του για εύρεση πιθανών αιτιών που προκαλούν διάφορες ασθένειες. Στη συγκεκριμένη εργασία, έγινε μια προσπάθεια κατηγοριοποίησης των σημειακών νουκλεοτιδικών πολυμορφισμών ατόμων που πάσχουν από κατά πλάκα σκλήρυνση και ατόμων που δεν έχουν αυτή την ασθένεια.

Λόγο της ανάγκης για κατηγοριοποίηση με μη επιβλεπόμενη μάθηση, επιλέχτηκε η υλοποίηση χαρτών αυτό-οργάνωσης. Η εφαρμογή αυτή όμως χρειάζεται αρκετό χρόνο για την κατηγοριοποίηση των δεδομένων και λόγο του ότι θα έπρεπε να τρέξει με διάφορα σενάρια θα χρειαζόταν πολύ χρόνο για την εξαγωγή αποτελεσμάτων. Αυτό ώθησε στην παραλληλοποίηση του αλγορίθμου και στην εύρεση κάποιων βελτιστοποιήσεων για την μείωση του χρόνου εκτέλεσής του. Συγκεκριμένα έγινε μια βελτιστοποίηση για τον προϋπολογισμό των αποστάσεων μεταξύ των νευρώνων του δισδιάστατου πλέγματος και μια για πιο αποδοτικό συγχρονισμό μεταξύ των επεξεργαστών. Τα αποτελέσματα ήταν αρκετά καλά και βοήθησαν στην μείωση του χρόνου εκτέλεσης σε σχέση με τον αντίστοιχο σειριακό χάρτη αυτό-οργάνωσης.

Αφού έτρεξε η εφαρμογή σε τρία διαφορετικά σενάρια με διαφορά στο μέγεθος των διαστάσεων του δισδιάστατου πλέγματος του χάρτη αυτό-οργάνωσης για δέκα φορές στο κάθε σενάριο, επιλέγηκαν οι σημαντικότερες κατηγορίες που προέκυψαν και συγκρίθηκαν για να ελεγχθεί η συνέπεια των αποτελεσμάτων. Στο δίκτυο με διαστάσεις 10x10 υπήρχε η μικρότερη συνέπεια των αποτελεσμάτων, ενώ με την αύξηση των διαστάσεων σε 20x20 και 50x50 αυξήθηκε και η συνέπεια των αποτελεσμάτων. Αυξάνοντας το δίκτυο όμως, μίκραινε και αριθμός των ατόμων που αποτελούσαν την κάθε κατηγορία, αλλά σε κάθε κατηγορία τα άτομα που την αποτελούσαν είχαν περισσότερα κοινά χαρακτηριστικά μεταξύ τους.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Παρακίνηση για αυτή την διπλωματική εργασία

Το μεγάλο ενδιαφέρον για την κατανόηση του τεράστιου όγκου πληροφοριών που προέρχονται από την μελέτη του γονιδιώματος των οργανισμών, οδήγησε στην ανάγκη για μελέτη των περιοχών του DNA για την εύρεση πιθανών αιτιών που προκαλούν διάφορες ασθένειες. Η μελέτη σημειακών νουκλεοτιδικών πολυμορφισμών (SNP), είναι μια μέθοδος για τη μελέτη της επίδρασης που μπορεί να έχει μια αλλαγή στην τιμή ενός νουκλεοτιδίου στο DNA στο φαινότυπο ενός ατόμου.

Οι αλληλεπιδράσεις των SNPs πιστεύεται ότι είναι υπεύθυνες για περίπλοκες ασθένειες, όπως η κατά πλάκα σκλήρυνση. Αρκετές μελέτες έχουν γίνει για την εύρεση του πώς αυτές οι αλληλεπιδράσεις μπορούν να επηρεάσουν σε διάφορες ασθένειες. Παρά το γεγονός ότι οι άνθρωποι μοιραζόμαστε πάνω από το 99% του DNA μας, εξακολουθούν να υπάρχουν εκατομμύρια αλλαγές στο DNA μεταξύ δύο ατόμων. Πιστεύεται ότι τα SNPs μπορούν να προκαλέσουν μια συγκεκριμένη ασθένεια, αλλά θεωρείται πολύ απίθανο από μόνο του ένα SNP να παίζει σημαντικό ρόλο στην πρόκληση μιας περίπλοκης ασθένειας όπως η κατά πλάκα σκλήρυνση.[1]

Η ανάπτυξη της τεχνολογίας σε υλικό και λογισμικό, μας επιτρέπει πλέον να τρέχουμε πιο απαιτητικές εφαρμογές στους υπολογιστές μας, για την επίλυση πιο περίπλοκων προβλημάτων. Η ύπαρξη αλγορίθμων για κατηγοριοποίηση δεδομένων, μας ώθησε στην ιδέα κατηγοριοποίησης των SNPs διάφορων ατόμων που είτε πάσχουν από την ασθένεια και

ατόμων που δεν έχουν αυτή την ασθένεια, ώστε να δούμε αν υπάρχουν κατηγορίες SNPs που μπορεί να ευθύνονται για την ανάπτυξη της ασθένειας αυτής.

1.2 Στόχος διπλωματικής εργασίας

Γενετικοί πολυμορφισμοί έχουν βρεθεί να επηρεάζουν την έκφραση γονιδίων σε όλα τα είδη κυττάρων. Έχοντας στη διάθεση μας πειραματικά δεδομένα από 1853 δείγματα (subjects) με 327 SNPs το κάθε ένα, όπου τα 881 δείγματα ήταν από υγιή άτομα (controls) και τα 972 από άτομα που είναι ασθενής της κατά πλάκας σκλήρυνσης (cases), θέλουμε να δούμε κατά πόσο μπορούμε να βρούμε κοινά πρότυπα μεταξύ των SNPs τα οποία ίσως προκαλούν ή αποτρέπουν την εμφάνιση της ασθένειας της κατά πλάκας σκλήρυνσης.

Για την εύρεση των κοινών προτύπων, θα χρησιμοποιηθεί ο αλγόριθμος των χαρτών αυτό-οργάνωσης, ο οποίος έχει την ικανότητα να κατηγοριοποιεί δεδομένα με κοινά χαρακτηριστικά σε ομάδες. Επειδή ο όγκος δεδομένων που θα δεχτεί ο αλγόριθμος είναι μεγάλος και απαιτεί αρκετό χρόνο για την εκτέλεσή του, είναι απαραίτητη η παραλληλοποίησή του και η προσπάθεια βελτιστοποίησής του. Επίσης, λόγο του ότι τα δεδομένα εισόδου έχουν γενετική σημασία, θα πρέπει να βρεθεί και ο τρόπος που θα κωδικοποιηθούν πριν δοθούν στον αλγόριθμο, ώστε να έχουμε καλύτερα αποτελέσματα. Στο τέλος θα γίνει μια σύγκριση της επιτάχυνσης του παράλληλου αλγορίθμου σε σύγκριση με το σειριακό και θα γίνει μια μεταεπεξεργασία στα αποτελέσματα του αλγόριθμου, ώστε να είναι πιο κατανοητά.

1.3 Διαμοιρασμός εργασίας

Λόγο του ότι ο όγκος εργασίας ήταν αρκετά μεγάλος και έπρεπε να ελεγχθούν δύο διαφορετικά σενάρια για την κωδικοποίηση των δεδομένων πριν περάσουν ως είσοδος στον αλγόριθμο, αποφασίστηκε να διαχωριστεί η εργασία σε υποεργασίες και να διαμοιραστούν σε μένα και το συμφοιτητή μου κύριο Λοΐζο Λοΐζου. Για κάθε υποεργασία ορίστηκε και ένας από εμάς ως υπεύθυνος για την ολοκλήρωση της. Ο διαχωρισμός που έγινε παρουσιάζεται στον πίνακα 1.1. Στις εργασίες που παρουσιάζονται περισσότερο από ένα άτομα υπεύθυνα, αφορά εργασίες οι οποίες έπρεπε να αναπτυχθούν ξεχωριστά ή περισσότερες από μια φορές είτε λόγω διαφορετικών δεδομένων εισόδου είτε επειδή μπορούσαν να ξαναμοιραστούν σε μικρότερες υπο-εργασίες που δεν αναφέρονται στον πίνακα.

Ο κύριος Άθως Αντωνιάδης, ήταν ο γενικός διαχειριστής ολόκληρου του έργου και η συνεισφορά του και η εμπειρία του ήταν καθοριστικοί παράγοντες για την ολοκλήρωση του έργου. Στον πίνακα 1.1, ο κύριος Άθως Αντωνιάδης παρουσιάζεται σε όλες τις υποεργασίες λόγω του ρόλου του. Οι υποεργασίες που διαμοιράστηκαν ανάμεσα σε μένα και τον κύριο Λοΐζο Λοΐζου, είχαν την ίδια βαρύτητα και φόρτο εργασίας, ώστε να υπάρχει ισότητα στην συνεισφορά του έργου. Χωρίς τη συνεργασία που υπήρξε μεταξύ μας, θα ήταν αδύνατη η εκπόνηση του συνολικού έργου, λόγο του ότι κάθε υποεργασία απαιτούσε σημαντικό χρόνο για την υλοποίησή της.

	Άθως Αντωνιάδης	Άριστος Αριστοδήμου	Λοΐζος Λοΐζου
Προεπεξεργασία – Αλγόριθμος Συμπίεσης – Μετατροπή .ped αρχείων σε δυαδική αποθήκευση .bin	X		X
Προεπεξεργασία – Αλγόριθμος Συμπίεσης – Μετατροπή .bin αρχείων σε .ped αρχείων	X	X	
Προεπεξεργασία – Φιλτράρισμα Δεδομένων	X	X	X
Χάρτες Αυτό-οργάνωσης – Σειριακός Αλγόριθμος με Categorical Data	X		X
Χάρτες Αυτό-οργάνωσης – Σειριακός Αλγόριθμος με Linear Data	X	X	
Χάρτες Αυτό-οργάνωσης – Παραλληλοποίηση και Βελτιστοποιήσεις	X	X	
Χάρτες Αυτό-οργάνωσης – Προσαρμογή σε γενετικά δεδομένα – Missing Data	X		X
Σφάλμα Κβαντισμού (Quantization Error)	X		X
Συνέπεια Αλγορίθμου (Consistency Checking)	X	X	X
Γραφικές Παραστάσεις Δισδιάστατου Χώρου	X	X	X
Γραφικές Παραστάσεις Βαρών	X		X

Πίνακας 1.1 Διαμοιρασμός υποεργασιών

Κεφάλαιο 2

Βασικές Έννοιες Μοριακής Βιολογίας

2.1 Ανθρώπινο Κύτταρο

Εικόνα 2.1 Ανθρώπινο Κύτταρο παρθέν από 'Life Form'

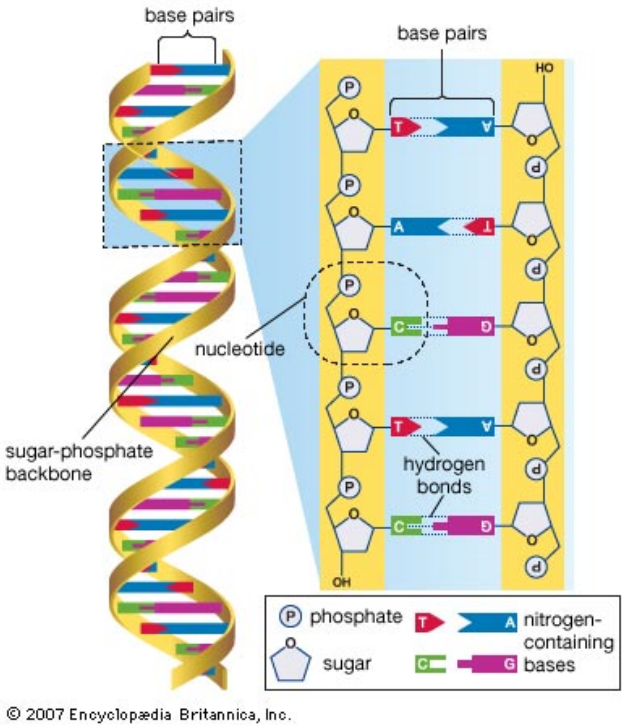
(http://www.life-form.eu/Natural%20History/Cell%20Model%2001med.jpg)

Το ανθρώπινο σώμα αποτελείται από τρισεκατομμύρια κύτταρα, τα οποία διαφέρουν ανάλογα με τη λειτουργία τους. Σε όλα τα κύτταρα του ανθρώπινου οργανισμού, υπάρχει μια κεντρική περιοχή, ο πυρήνας, ο οποίος περιέχει όλες τις γενετικές πληροφορίες ενός ανθρώπου. Συγκεκριμένα, οι πληροφορίες αυτές βρίσκονται στο δεσοξυριβοζονουκλεϊνικό οξύ (DNA) το οποίο καθορίζει τα χαρακτηριστικά ενός ανθρώπου, όπως είναι το χρώμα των

ματιών ή ακόμη και την ευπάθεια σε μια ασθένεια. Έχει τη μορφή διπλής έλικας και μορφοποιείται σε χρωμοσώματα μόνο στη φάση που το κύτταρο πολλαπλασιάζεται. Στον πυρήνα του ανθρώπινου κυττάρου, υπάρχουν 23 ζεύγη χρωμοσωμάτων. Κάθε περιοχή του DNA που αναφέρεται σε ένα συγκεκριμένο χαρακτηριστικό, ονομάζεται γονίδιο και είναι υπεύθυνη για την παραγωγή πρωτεϊνών, μέσω των οποίων τα κύτταρα εκτελούν τις απαραίτητές τους λειτουργίες. [2]

2.2 Δεσοξυριβοζονουκλεϊνικό οξύ

Το DNA περιέχει όλες τις γενετικές πληροφορίες των περισσότερων οργανισμών και είναι διαφορετικό για κάθε άνθρωπο. Βρίσκεται στον πυρήνα των κυττάρων και ο κύριος ρόλος του είναι η μακροχρόνια αποθήκευση πληροφοριών και οδηγιών για την παραγωγή άλλων συστατικών των κυττάρων, όπως είναι τα μόρια RNA και οι πρωτεΐνες. Η ανακάλυψη της δομής του έγινε το 1953 από τους Watson & Crick, οι οποίοι παρουσίασαν ένα μοντέλο της δομής του DNA, που ονομάσθηκε "μοντέλο της διπλής έλικας". Σύμφωνα με το μοντέλο αυτό, το DNA αποτελείται από δύο πολυνουκλεοτιδικές αλυσίδες σε μορφή δύο αντιτακτών κλώνων που σχηματίζουν δεξιόστροφη διπλή έλικα. Οι αλυσίδες αυτές συγκροτούνται από αζωτούχες βάσεις, φωσφορικές ρίζες και ένα σάκχαρο με πέντε άτομα άνθρακα (πεντόζη). Οι αζωτούχες βάσεις είναι η αδενίνη (A), η θυμίνη (T), η γουανίνη (G) και η κυτοσίνη (C). Οι έλικες του DNA είναι αντιπαράλληλες και συνδέονται πάντοτε σύμφωνα με τον κανόνα συμπληρωματικότητας των Watson & Crick, με την αδενίνη να ενώνεται πάντοτε με τη θυμίνη με δύο δεσμούς υδρογόνου και την κυτοσίνη να ενώνεται πάντοτε με τη γουανίνη με τρεις δεσμούς υδρογόνου. Η αλληλουχία των νουκλεοτιδίων είναι υπεύθυνη για την κωδικοποίηση και την αποθήκευση της πληροφορίας για το πως εκφράζεται μία πρωτεΐνη. Η διαδικασία με την οποία αναπαράγεται το DNA ονομάζεται αντιγραφή [2].

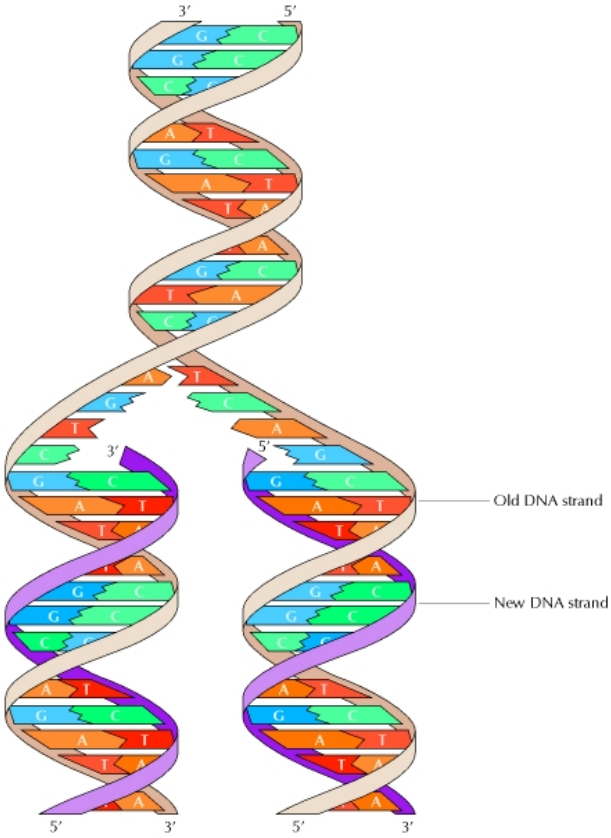


Εικόνα 2.2 Η δομή του DNA παρθέν από 'Encyclopedia Britannica'

(http://media-2.web.britannica.com/eb-media/64/47664-004-7088EE3D.jpg)

2.3 Αντιγραφή

Κατά τη διαδικασία της αντιγραφής, οι δύο έλικες του DNA διαχωρίζονται, με αποτέλεσμα να μένουν ελεύθερες οι βάσεις των νουκλεοτιδίων. Τότε ένζυμα, φέρνουν απέναντι από κάθε βάση την συμπληρωματική της. Με αυτό τον τρόπο, κάθε αλυσίδα, λειτουργεί ως καλούπι για την δημιουργία της συμπληρωματικής της, και το DNA αυτοδιπλασιάζεται. Η διαδικασία αυτή οδηγεί στην δημιουργία δύο μορίων οι DNA οι οποίοι είναι όμοιοι μεταξύ τους, αλλά και με το αρχικό μόριο DNA.



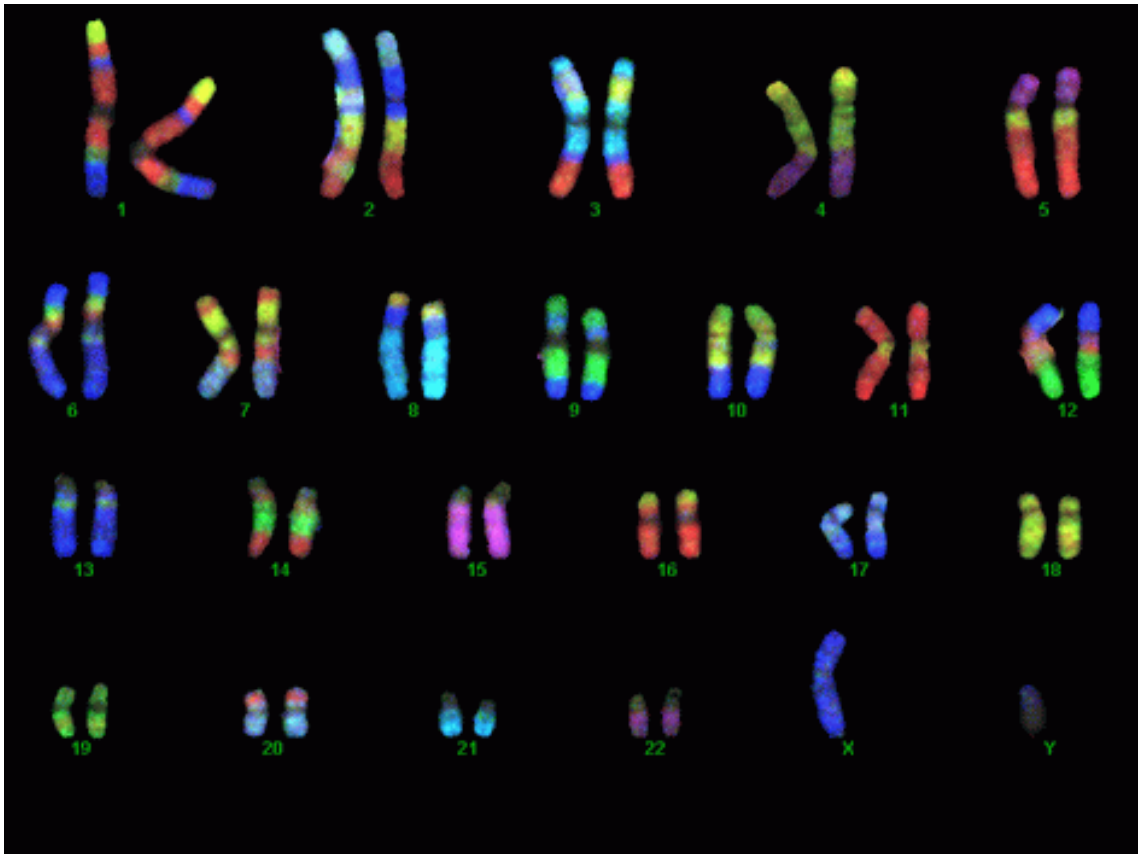
Εικόνα 2.3 Η αντιγραφή του DNA παρθέν από 'NCBI'

(http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=cooper&part=A417&rendertype=figure&id=A430)

2.4 Χρωμοσώματα

Τα χρωμοσώματα αποτελούν οργανωμένες δομές DNA ή διαφορετικά, δομές που σχηματίζουν πακέτα DNA. Κάθε ένα από αυτά περιέχει ένα πάρα πολύ μακρύ μόριο DNA, που είναι πακεταρισμένο με τέτοιο τρόπο ώστε να έχει 50000 φορές μικρότερο μήκος. Στη διαδικασία δημιουργίας των χρωμοσωμάτων λαμβάνουν μέρος κάποιες ειδικές πρωτεΐνες που συνδέονται μαζί με το μακρομόριο του DNA σχηματίζοντας ένα σύμπλοκο που ονομάζεται χρωματίνη. Το βασικό στοιχείο από το οποίο αποτελείται η χρωματίνη είναι το νουκλεόσωμα, ένα πρωτεϊνικό οκταμερές που αποτελείται από τέσσερις διαφορετικούς τύπους πρωτεϊνών που ονομάζονται ιστόνες. Στον άνθρωπο κάθε σωματικό κύτταρο περιέχει δύο αντίγραφα από κάθε χρωμόσωμα από τα οποία το ένα προέρχεται από τον πατέρα και το άλλο από τη μητέρα. Τα δύο αυτά αντίγραφα σχηματίζουν ζεύγη χρωμοσωμάτων που ονομάζονται ομόλογα λόγω της ίδιας δομής και οργάνωσης που παρουσιάζουν. Υπάρχουν 23 τέτοια ζεύγη ομόλογων χρωμοσωμάτων στον άνθρωπο, κάθε ένα από τα οποία είναι υπεύθυνο για την έκφραση ενός διαφορετικού χαρακτηριστικού του ανθρώπινου οργανισμού και ένα από αυτά είναι υπεύθυνο για το φύλο του ανθρώπου. Το 23^{ο} ζεύγος, δεν αποτελεί ομόλογο ζεύγος χρωμοσωμάτων, γιατί στην περίπτωση άρρεν ατόμου, το ένα αποτελείται

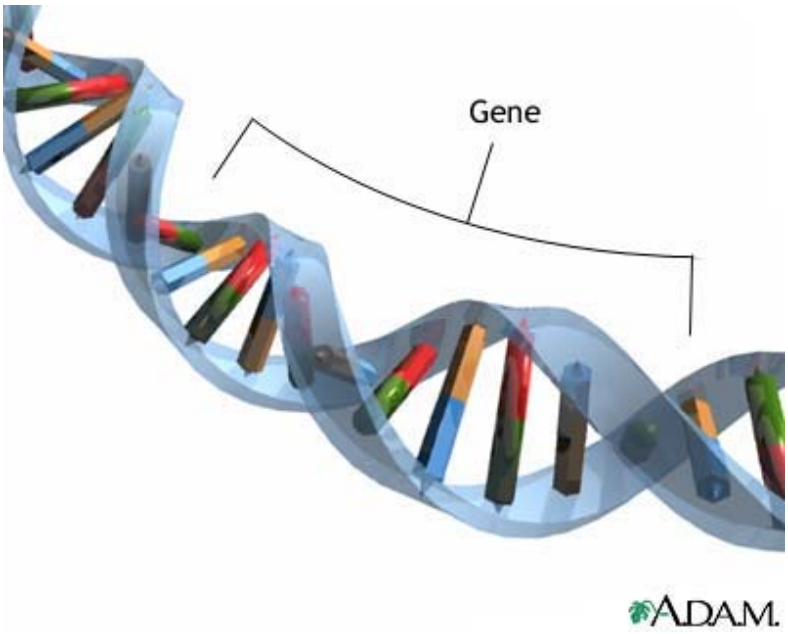
από το X και το άλλο από το Y. Σε περίπτωση που τα δύο αυτά χρωμοσώματα είναι ομόλογα, δηλαδή XX τότε το άτομο αυτό είναι θηλυκό. [2].



Εικόνα 2.4 Τα 23 ζεύγη ομόλογων χρωμοσωμάτων παρθέν από 'EMERGENCE' (http://www.homodiscens.com/home/core_content/who_knows/astonishing_predicament/contingency/humilis/dangerous_algorithm/nature_self_aware/karyotype.gif)

2.5 Γονίδια

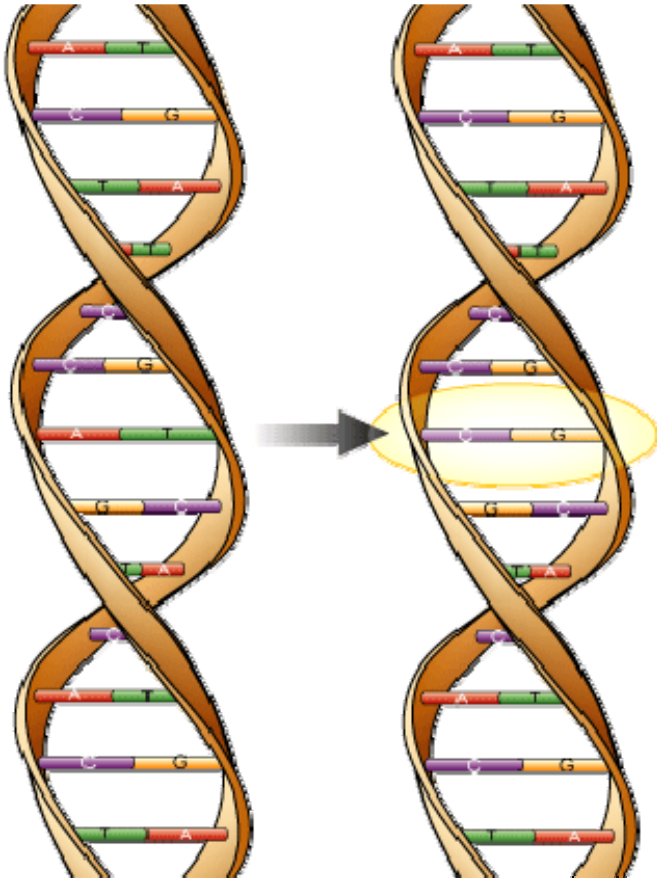
Τα γονίδια αποτελούν τμήματα DNA που μπορούν να μεταφραστούν σε ένα προϊόν πρωτεΐνης. Τα τμήματα αυτά του DNA που δεν μεταφράζονται δηλαδή που δεν παράγουν κάποιο προϊόν ονομάζονται ιντρόνια ενώ τα τμήματα που μεταφράζονται παράγοντας κάποιο προϊόν ονομάζονται εξώνια. Τα γονίδια αποτελούν εξώνια γιατί κάθε μετάφραση γονιδίου παράγει κάποιο προϊόν πρωτεΐνης [2].



Εικόνα 2.5 Το γονίδιο παρθέν από ‘Walgreens’
(http://www.walgreens.com/library/contents.html?docid=000437&doctype=10)

2.6 Σημειακοί Νουκλεοτιδικοί Πολυμορφισμοί (SNP

Σημειακοί νουκλεοτιδικοί πολυμορφισμοί (single nucleotide polymorphism SNP), είναι παραλλαγές που παρατηρούνται στις ακολουθίες του DNA και οι οποίες εμφανίζονται όταν ένα νουκλεοτίδιο (A,T,C ή G) στην ακολουθία του γονιδιώματος αλλαχθεί. Για παράδειγμα ένα SNP θα μπορούσε να αλλάξει μια ακολουθία DNA από AAGGCTAA σε ATGGCTAA. Τα SNPs καλύπτουν 90% των γενετικών παραλλαγών που παρατηρούνται στον άνθρωπο. Ένας τέτοιος πολυμορφισμός παρατηρείται κάθε 100 μέχρι και 300 βάσεις κατά μήκος των 3 δισεκατομμυρίων βάσεων του ανθρώπινου γονιδιώματος. Στο παράδειγμα που προαναφέρθηκε, τα δύο διαφορετικά νουκλεοτίδια που παρουσιάζονται στην ίδια θέση στις δύο νουκλεοτιδικές ακολουθίες, θεωρούνται αλληλόμορφα. Τα περισσότερα κοινά SNPs αποτελούνται από 2 μόνο αλληλόμορφα. Οι τέσσερις διαφορετικές περιπτώσεις SNPs είναι AA, Aα, αα και η τέταρτη περίπτωση αυτή στην οποία παρατηρούνται missing data δηλαδή έλλειψη δεδομένων, διαφορετικά έλλειψη νουκλεοτιδίων σε κάποιες θέσεις των ακολουθιών DNA. Τα SNPs μπορούν να εμφανιστούν και στα εξώνια αλλά και στα ιντρόνια. Δηλαδή μπορούν να εμφανιστούν σε περιοχές του DNA που κωδικοποιούν κάποιο προϊόν αλλά και σε αυτές τις περιοχές που δεν κωδικοποιούν κάποια πρωτεΐνη ή κάποιο μόριο mRNA. Πολλά από τα SNPs δεν επηρεάζουν την λειτουργία του κυττάρου όμως πιστεύεται πως κάποια άλλα θα μπορούσαν να δημιουργήσουν την προδιάθεση στον άνθρωπο να ασθενήσει ή ακόμη να επηρεάσουν την αντίδραση του οργανισμού τους σε κάποιο φάρμακο.



Εικόνα 2.6 Σημεικός Νουκλεοτιδικός Πολυμορφισμός (SNP) παρθέν από 'The Science Creative Quarterly' (http://www.scq.ubc.ca/wp-content/uploads/2006/07/dna1.gif)

2.7 Linkage Disequilibrium (LD

Ανισορροπία συνδέσμων (linkage disequilibrium - LD) η κατάσταση στην οποία κάποιοι συνδυασμοί αλληλόμορφων γονιδίων ή γενετικών δεικτών παρατηρούνται σε μεγαλύτερο ή λιγότερο συχνά σε ένα πληθυσμό, απ' ότι θα αναμενόταν από τον τυχαίο σχηματισμό των απλότυπων (haplotypes) των αλληλόμορφων γονιδίων, που βασίζεται στη συχνότητα τους. Αποτελούν μη τυχαίες συσχετίσεις μεταξύ πολυμορφισμών που παρατηρούνται σε διαφορετικούς γεωμετρικούς τόπους. Αιτία αυτής της κατάστασης είναι η παρουσία γενετικών συνδέσμων, δηλαδή γενετικές περιοχές στις οποίες το ποσοστό ανασυνδυασμού που μπορεί να προκύψει παρατηρείται να είναι σταθερό σε ένα πληθυσμό και διαφορετικό από το τι θα αναμενόταν αν οι συνδυασμοί ήταν τυχαίοι [3].

2.8 Επίσταση

Ο όρος επίσταση, χρησιμοποιήθηκε για πρώτη φορά από τον Bateson το 1909, για να περιγράψει το φαινόμενο όπου μια παραλλαγή ή ένα αλληλόμορφο γονίδιο (allele) σε μια

περιοχή, αποτρέπει σε μια άλλη παραλλαγή σε κάποια άλλη περιοχή να επιδράσει στο φαινότυπο του ατόμου. Ας υποθέσουμε ότι έχουμε 2 περιοχές, την B και την G, οι οποίες καθορίζουν το χρώμα του τριχώματος ενός ποντικού. Η περιοχή B έχει δύο πιθανά alleles, B ή b, και η περιοχή G το G ή g. Τα πιθανά αποτελέσματα για τον φαινότυπο (άσπρο, γκρίζο, μαύρο) φαίνονται στον πίνακα 2.1. Βλέπουμε ότι ανεξάρτητα του γεγονότος ότι στην περιοχή G έχουμε g/g, όταν στην περιοχή B υπάρχει έστω και ένα allele με τιμή B, τότε το χρώμα είναι το μαύρο. Επομένως, το allele B είναι επικρατέστερο του b. Όταν υπάρχει έστω και ένα allele με την τιμή G, τότε ανεξαρτήτως των υπόλοιπων alleles, το χρώμα είναι γκρίζο. Αυτό σημαίνει ότι το G είναι επικρατέστερο του g, αλλά ότι είναι και επιστατικό ως προς την περιοχή B (πιο συγκεκριμένα, το allele G της περιοχής G, είναι επιστατικό ως προς το allele B της περιοχή B). [4]

	Genotype at locus G		
Genotype at locus B	*g/g*	*g/G*	*G/G*
b/b	White	Grey	Grey
b/B	Black	Grey	Grey
B/B	Black	Grey	Grey

Πίνακας 2.1 Παράδειγμα φαινοτύπου από διαφορετικούς γονότυπους σε δύο περιοχές που αλληλεπιδρούν επιστατικά, βάση του ορισμού της επίστασης που έδωσε ο Bateson παρθέν από 'Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans' http://hmg.oxfordjournals.org/cgi/reprint/11/20/2463

Κεφάλαιο 3

Μεθοδολογία

3.1 Βάση Δεδομένων

Η συλλογή των δεδομένων εισόδου έγινε από πολλά κέντρα και η προσπάθεια ξεκίνησε το 2003. Τρία κλινικά κέντρα της Κατά πλάκα Σκλήρυνσης (MS) συνεργάστηκαν για να πάρουν βιολογικά δείγματα από ασθενείς. Τα δύο ήταν στην Ευρώπη και το ένα στην Αμερική. Η μελέτη αυτή πήρε ασθενείς με ιστορική καταγωγή από την βόρεια Ευρώπη και κατά προτίμηση να έχουν μια αρχική υποτροπή MS, αν και τα άτομα με όλες τις κλινικές υποκατηγορίες της ασθένειας συμμετείχαν, συμπεριλαμβανομένου του κλινικά απομονωμένου συνδρόμου (CIS), της επιστρέφοντας υποτροπής MS (RRMS), τη δευτεροβάθμια προοδευτική MS (SPMS), την αρχική προοδευτική MS (PPMS) και την

προοδευτική υποτροπή MS (PRMS) [5]

Από τη βάση αυτή, μας δόθηκαν 1853 δείγματα με 327 SNPs το κάθε ένα, όπου τα 881 δείγματα ήταν από υγιή άτομα (controls) και τα 972 από ασθενείς (cases). Η μορφή που μας δόθηκαν ήταν σε QTDT Merlin. Τα δεδομένα είναι σπασμένα σε 3 αρχεία, το αρχείο Pedigree που περιέχει το φαινότυπο και το γονότυπο για ένα συγκεκριμένο αριθμό δειγμάτων (subjects), το αρχείο Map που περιέχει πληροφορίες για κάθε SNP και το Dat αρχείο που περιγράφει το αρχείο Pedigree.

Το pedigree αρχείο, συνήθως ξεχωρίζει τα δεδομένα με τη χρήση white-spaces. Οι πρώτες 6 στήλες, περιέχουν πληροφορίες για το δότη του δείγματος (Ταυτότητα οικογένειας, Ταυτότητα ατόμου, Ταυτότητα πατέρα, Ταυτότητα μητέρας, φύλο, φαινότυπος). Ο συνδυασμός των πληροφοριών κάθε ατόμου πρέπει να είναι μοναδικός. Οι επόμενες στήλες περιέχουν bi-allelic δείκτες που είναι τα SNPs. Οι δείκτες των γονότυπων είναι κωδικοποιημένοι με τα γράμματα "A", "C", "T" και "G", ένα για κάθε allele. Για να παρουσιάσουν ένα ελλιπών allele χρησιμοποιείται η τιμή 0. [6]

Τα αρχεία Map χρησιμοποιούνται για την ανάλυση των δεικτών στο αντίστοιχο αρχείο pedigree. Κάθε γραμμή είναι για ένα δείκτη και περιέχει 3 τιμές, το χρωμόσωμα, την ταυτότητα του SNP και τη τοποθεσία του σε αριθμό βάσεων στο strand. [6]

Το Dat αρχείο, έχει μια γραμμή για κάθε δεδομένο στο pedigree αρχείο, δείχνοντας τον τύπο του δεδομένου χρησιμοποιώντας ένα γράμμα ως αναγνωριστικό. [6]

3.3 Προεπεξεργασία

Για να μειώσουμε το μέγεθος των δεδομένων και να χρησιμοποιήσουμε δεδομένα που πιθανό να είναι πιο σημαντικά, χρειάστηκε να γίνει μια προεπεξεργασία στα αρχικά μας δεδομένα. Επίσης έγινε και μια κωδικοποίηση των SNP, ώστε να ακολουθούν ένα συγκεκριμένο πρότυπο, και να μπορεί το δίκτυο να κατηγοριοποιήσει τα δεδομένα βάση αυτών των χαρακτηριστικών που μας ενδιαφέρουν.

3.3.1 Συμπίεση δεδομένων [7]

Λόγο του ότι οι βάσεις δεδομένων που είχαμε να χειριστούμε καταλάμβαναν αρκετό χώρο, χρειάστηκε να υλοποιήσουμε ένα αλγόριθμο συμπίεσης και αποσυμπίεσης των δεδομένων. Με τον τρόπο αυτό θα γλιτώσουμε χώρο, αλλά θα μπορούμε και να τα τρέξουμε στην

συμπιεσμένη τους μορφή σε μια εφαρμογή του κυρίου Άθου Αντωνιάδη, η οποία θα μας έδινε κάποια αποτελέσματα που θα χρειαζόμασταν όπως η επίσταση, χωρίς να πάρουν μεγάλο χώρο στη RAM.

Λόγο του ότι οι αναλύσεις καταλογισμού (imputation analyses) χρειάζονται την πληροφορία του strand του allele του ετερόζυγου SNP, χρειάζεται να κωδικοποιήσουμε όλα τα allele σε κάθε strand ξεχωριστά για όλες τις περιπτώσεις. Υπάρχουν 3 καταστάσεις όπου ένα allele μπορεί να βρίσκεται. Αυτό που προτείναμε είναι να χρησιμοποιούνται 2 bit για την κωδικοποίηση κάθε allele ώστε να αντιπροσωπεύουν τα δύο πιθανά νουκλεοτίδια (Α ή α), την περίπτωση του ελλιπούς στοιχείου και την περίπτωση του διαγραμμένου στοιχείου.

Η προτεινόμενη μορφή αποτελείται από ένα δισδιάστατο πίνακα από alleles. Το μέγεθος της πρώτης διάστασης είναι ίσο με τον αριθμό των strands που έχει ο δότης. Στους ανθρώπους όλα τα χρωμοσώματα έχουν 2 strands με εξαίρεση το X και Y χρωμόσωμα στους άντρες που έχουν από ένα strand. Σε αυτή την περίπτωση μπορεί να υπάρχει ένα δεύτερο strand που να καταχωρεί τα alleles ως ελλιπή.

Κάθε στοιχείο του πίνακα για κάθε strand θα κωδικοποιεί ένα allele. Το allele θα κωδικοποιείται από 2 bits επιτρέποντας έτσι την κωδικοποίηση 4 καταστάσεων για αυτό. Στον πίνακα 3.1 φαίνεται η προτεινόμενη κωδικοποίηση για ένα allele και στον πίνακα 3.2 η προτεινόμενη κωδικοποίηση για alleles σε δύο strands.

Allele	Κωδικοποίηση
Άγνωστο	00
A	01
a	10
Διαγραμμένο	11

Πίνακας 3.1 Προτεινόμενη Κωδικοποίηση Allele

Allele 1	Allele 2	Κωδικοποίηση
Άγνωστο	Άγνωστο	0000
Άγνωστο	A	0001
Άγνωστο	a	0010
Άγνωστο	Διαγραμμένο	0011
A	Άγνωστο	0100
A	A	0101
A	a	0110
A	Διαγραμμένο	0111
a	Άγνωστο	1000
a	A	1001
a	a	1010

a	Διαγραμμένο	1011
Διαγραμμένο	Άγνωστο	1100
Διαγραμμένο	A	1101
Διαγραμμένο	a	1110
Διαγραμμένο	Διαγραμμένο	1111

Πίνακας 3.2 Προτεινόμενη κωδικοποίηση bi-allelic δείκτες

Τα 2 strands σε κάθε πίνακα του δότη, πρέπει να είναι απόλυτα ταυτισμένα, δηλ. το ι-οστό στοιχείο κάθε πίνακα θα δείχνει στο ίδιο δείκτη alleles στο κάθε strand. Αυτό ευκολύνει την πρόσβαση στην πληροφορία αφού έτσι γινόταν και στις προηγούμενες μεθόδους.

Βάση αυτών, υλοποιήσαμε τον αλγόριθμο για την συμπίεση και αποσυμπίεση των δεδομένων. Συγκεκριμένα, την εφαρμογή για την συμπίεση των δεδομένων ανέλαβε και υλοποίησε ο κύριος Λοΐζος Λοΐζου και εγώ ανέλαβα την υλοποίηση της εφαρμογής για την αποσυμπίεση των δεδομένων.

3.3.1 Επιλογή επιθυμητών δεδομένων

Η βάση δεδομένων περιέχει 327 SNP για κάθε ασθενή και από αυτά έπρεπε να επιλεγούν αυτά που είναι πιο σημαντικά. Τα SNP που επιλέγηκαν από τη βάση δεδομένων, είναι αυτά που έχουν επίσταση μεταξύ τους μεγαλύτερη του 45 ώστε να έχουμε p-value < 10^{-9} . Για να απομονώσουμε τα SNP αυτά, αρχικά επιλέξαμε τα SNP που είχαν αυτές τις τιμές και στη συνέχεια με τη βοήθεια του plink [8], αφαιρέσαμε τα υπόλοιπα SNP από τη βάση δεδομένων μας.

3.3.2 Κωδικοποίηση δεδομένων

Για την εύρεση του τρόπου κωδικοποίησης των δεδομένων υπεύθυνος ήταν ο κύριος Λοΐζος Λοΐζου. Μέσω έρευνας και συζήτησης μεταξύ μας, για την κωδικοποίηση των δεδομένων για το δικό μου μέρος που θα γίνονταν γραμμικά, αποφασίστηκε η κωδικοποίηση που φαίνεται στον πίνακα 3.3.

Allele	Κωδικοποίηση
AA	1
aA	2
Aa	2
aa	3
Άγνωστο	-1

Πίνακας 3.3 Κωδικοποίηση alleles σε γραμμική μορφή

Τα δεδομένα μετατρέπονται σε γραμμικά, αντί να χρησιμοποιείται δυαδική απεικόνιση. Για κάθε SNP, αναπαριστώνται 3 καταστάσεις ανάλογα με τις τιμές των alleles τους. Συγκεκριμένα, το AA μετατρέπεται σε 1, τα aA και Aa σε 2 και το aa σε 3. Η βασική διαφορά μεταξύ των δύο αναπαραστάσεων αφορά τον διαχωρισμό μεταξύ aA και Aa, αφού στην υλοποίηση με τα γραμμικά δεδομένα δεν διαχωρίζεται – θεωρείται σαν ίδια περίπτωση – ενώ στην δυαδική αναπαράσταση που υλοποιεί ο κύριος Λοΐζος Λοΐζου, το δίκτυο μπορεί να τα αναγνωρίσει έως ξεχωριστές παραστάσεις.

3.3 Self Organizing Map (SOM)

Τα SOM, ή αλλιώς δίκτυα Kohonen, χρησιμοποιούνται για την κατηγοριοποίηση δεδομένων. Ουσιαστικά βάζουν σε κοινές κατηγορίες τα δεδομένα που έχουν μεταξύ τους κοινές περιοχές. Αυτός είναι ένας τρόπος συμπίεσης των δεδομένων, γιατί από τις πολλές αρχικές κατηγορίες που έχουμε στην αρχή (ουσιαστικά κάθε δεδομένο εισόδου είναι και μια ξεχωριστή κατηγορία), καταλήγουμε σε ένα μικρότερο αριθμό, όπου κάθε κατηγορία περιέχει δεδομένα με κοινά χαρακτηριστικά. Αυτά τα δίκτυα τα χρησιμοποιούμε όταν υποθέτουμε ότι στα δεδομένα μας υπάρχουν κοινά χαρακτηριστικά, τα οποία το δίκτυο θα αναγνωρίσει και έτσι θα τα κατατάξει σε κατηγορίες. [9][10]

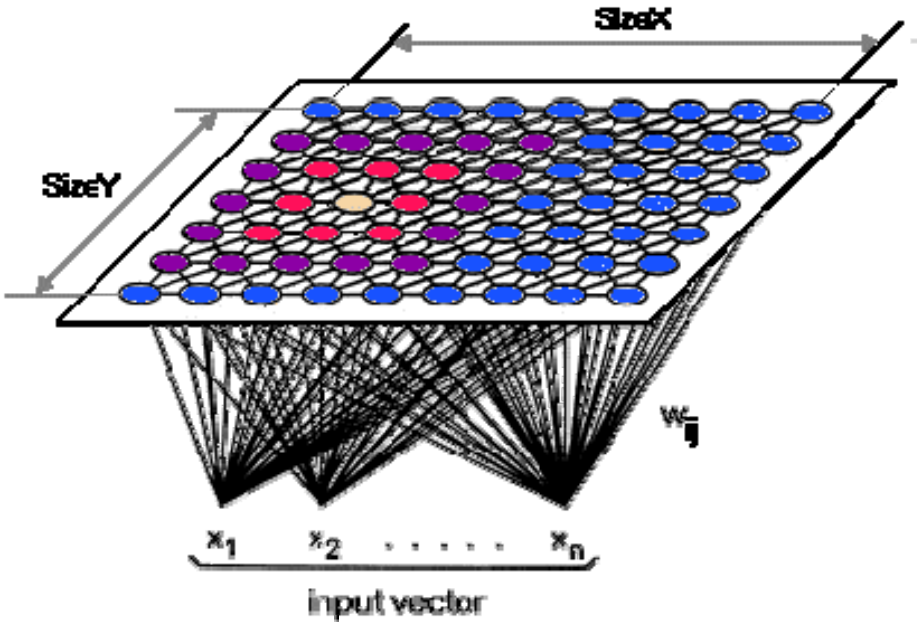
Τα SOM ανήκουν στην κατηγορία των νευρωνικών δικτύων μη επιβλεπόμενης μάθησης. Σε ένα νευρωνικό δίκτυο με επιβλεπόμενη μάθηση ή με ενισχυτική μάθηση, έχουμε το επιθυμητό αποτέλεσμα, το οποίο χρησιμοποιούμε για να βοηθήσουμε το δίκτυο να εκπαιδευτεί. Στα SOM όμως δεν χρειάζεται να γνωρίζουμε το επιθυμητό αποτέλεσμα για να εκπαιδεύσουμε το δίκτυο. Τα δίκτυα αυτά είναι ικανά από μόνα τους να εντοπίσουν τα κοινά χαρακτηριστικά των δεδομένων και να τα κατηγοριοποιήσουν μαζί.[11]

Ο χρόνος όμως που απαιτείται για να μπορέσει να εκπαιδευτεί σωστά, είναι αρκετά μεγάλος. Επίσης, η εύρεση των κατάλληλων παραμέτρων για το δίκτυο γίνεται εμπειρικά, δηλ. πρέπει να τρέξεις το SOM αρκετές φορές με διάφορες παραμέτρους μέχρι να βρεις τις ιδανικές παραμέτρους για το πρόβλημά σου.

Επειδή αρχικά υπήρχε μεγάλος όγκος δεδομένων και θα έπρεπε να δοκιμαστούν διαφορετικές παράμετροι στο δίκτυο, παραλληλοποιήθηκε ο αλγόριθμος του SOM ώστε να μειωθεί ο χρόνος εκτέλεσής του. Για την υλοποίησή του, χρησιμοποιήθηκαν τα pthreads και ως γλώσσα προγραμματισμού η C++. Τα pthreads είναι για περιβάλλον Unix και Shared

Memory μηχανές και έτσι η εφαρμογή αυτή δεν τρέχει σε Windows και σε υπολογιστές με Message Passing αρχιτεκτονικές. Παρόλα αυτά όμως, ο αλγόριθμος μπορεί να υλοποιηθεί και για αυτού του είδους μηχανές.

3.3.1 Αρχιτεκτονική Σειριακού SOM



Εικόνα 3.1 Δομή δικτύου Kohonen παρθέν από 'Lohninger'
(http://www.lohninger.com/helpcsuite/kohonen_network_-_background_information.htm)

Ένα δίκτυο Kohonen αποτελείται από ένα δισδιάστατο πλέγμα από νευρώνες, όπου κάθε νευρώνας είναι και νευρώνας εξόδου. Οι νευρώνες είναι συνδεδεμένοι μεταξύ τους με διάφορες τοπολογίες, ανάλογα και με το πρόβλημα που έχουν να λύσουν. Ανάλογα με τη συνδεσμολογία τους, καθορίζεται και η γειτνίασή τους με τους άλλους κόμβους, κάτι που είναι σημαντικό να γνωρίζουμε κατά την εκπαίδευσή του δικτύου. Κάθε δεδομένο εισόδου, παρουσιάζεται στο δίκτυο ως ένα διάνυσμα εισόδου με το οποίο κάθε νευρώνας του δισδιάστατου πλέγματος είναι συνδεδεμένος μέσω των βαρών του. Συγκεκριμένα, αν το διάνυσμα εισόδου έχει n στοιχεία, τότε κάθε νευρώνας έχει n βάρη που το κάθε ένα συνδέεται με το αντίστοιχο στοιχείο του διανύσματος εισόδου. [10][11]

3.3.2 Εκπαίδευση Σειριακού SOM

Τα SOM ανήκουν στην κατηγορία των νευρωνικών δικτύων και πιο συγκεκριμένα στην κατηγορία των competitive networks without supervision. Δηλαδή δεν υπάρχει διάνυσμα στόχου. Αντίθετα η εκμάθηση του δικτύου γίνεται με διαφορετικό τρόπο. Αρχικά έχουμε μια τυχαία αρχικοποίηση των βαρών κάθε νευρώνα του δισδιάστατου πλέγματος με μικρές τιμές. Στη συνέχεια συγκρίνονται τα βάρη κάθε νευρώνα με κάθε στοιχείο του διανύσματος εισόδου, και ο νευρώνας του οποίου τα βάρη μοιάζουν περισσότερο με το διάνυσμα εισόδου επιλέγεται ως ο νικητής (best matching unit ή BMU). Στη συνέχεια τα βάρη του BMU και των γειτονικών του νευρώνων, προσαρμόζονται ώστε να μοιάζουν περισσότερο με το διάνυσμα εισόδου.

Συγκεκριμένα ο αλγόριθμος είναι ο εξής:

1. Αρχικοποίησε τα βάρη των νευρώνων του δισδιάστατου πλέγματος με τυχαίες μικρές τιμές. Αρχικοποίησε το μέγεθος της γειτονιάς (r_0) και το ρυθμό μάθησης (L_0).
2. Επέλεξε τυχαία ένα διάνυσμα εισόδου από το σύνολο διανυσμάτων που έχουμε ως δεδομένα και θέλουμε να κατηγοριοποιήσουμε.
3. Σύγκρινε τα βάρη κάθε νευρώνα με το αντίστοιχο στοιχείο του διανύσματος εισόδου και υπολόγισε την ομοιότητα μεταξύ των βαρών και του διανύσματος εισόδου (ο έλεγχος γίνεται με βάση μια συνάρτηση και συνήθως γίνεται με την Ευκλείδεια απόσταση).
4. Επέλεξε τον νευρώνα του οποίου τα βάρη «μοιάζουν» περισσότερο από κάθε άλλου κόμβου με το διάνυσμα εισόδου και όρισέ τον ως τον BMU.
5. Προσάρμοσε τα βάρη του BMU και των νευρώνων που βρίσκονται στη γειτονιά του ώστε να μοιάσουν περισσότερο στο διάνυσμα εισόδου. Όσο πιο κοντά είναι κάποιος νευρώνας στον BMU, τόσο μεγαλύτερη αλλαγή υφίστανται τα βάρη του, ενώ ο BMU έχει τη μεγαλύτερη αλλαγή στα βάρη του.
6. Μείωσε το ρυθμό μάθησης L(i) και το μέγεθος της γειτονιάς r(i)
7. Επανέλαβε από το βήμα 2 [9]
8. Τρέξε μια φορά τον αλγόριθμο περνώντας όλα τα δεδομένα με τη σειρά χωρίς να κάνεις αλλαγές στα βάρη και έχοντας ως μέγεθος της ακτίνας το 1. Αύξησε το μετρητή κάθε νευρώνα που επιλέγεται ως BMU και σημείωσε πιο δεδομένο τον ενεργοποίησε.
9. Δημιούργησε ένα αρχείο με το μετρητή κάθε νευρώνα και τη θέση του στο πλέγμα, ένα αρχείο με τα δεδομένα που ενεργοποίησαν κάθε νευρώνα στο βήμα 10 και ένα αρχείο με τις τιμές των βαρών κάθε νευρώνα.

Ο αλγόριθμος επαναλαμβάνεται από το βήμα 2 για ένα αριθμό επαναλήψεων που λέγονται εποχές. Ο αριθμός των εποχών καθορίζεται από το χρήστη και η τιμή αυτή μπορεί να επηρεάσει σημαντικά την κατηγοριοποίηση που θα κάνει το δίκτυο. Για μικρό αριθμό εποχών, το δίκτυο μπορεί να μην προλάβει να εκπαιδευτεί σωστά με αποτέλεσμα να μην έχουμε σωστή κατηγοριοποίηση των δεδομένων. Σε αντίθετη περίπτωση, που ο αριθμός των εποχών είναι πολύ μεγάλος, το δίκτυο μπορεί να μάθει να κατηγοριοποιεί μόνο τα συγκεκριμένα δεδομένα, με αποτέλεσμα αν του δώσουμε κάποιο νέο δεδομένο εισόδου για να δούμε σε ποια κατηγορία ανήκει, να πάρουμε λάθος αποτελέσματα.

Για να γίνει καλύτερα κατανοητός ο αλγόριθμος θα δώσουμε βαρύτητα στο τρόπο επιλογής του BMU , το τρόπο που υπολογίζεται η ακτίνα του BMU με την βοήθεια της συνάρτησης topological neighborhood, όπως επίσης και το πώς αλλάζουν τα βάρη στους κόμβους που βρίσκονται μέσα στην ακτίνα του BMU σε κάθε επανάληψη.

Η επιλογή του BMU μπορεί να γίνει με την βοήθεια πολλών συναρτήσεων. Συνήθως όμως γίνεται η χρήση της Ευκλείδειας απόστασης. Δηλαδή για κάθε μονάδα – νευρώνα εισόδου υπολογίζεται η Ευκλείδεια απόσταση του διανύσματος εισόδου με το διάνυσμα των βαρών του. Ο κόμβος με την μικρότερη απόσταση ορίζεται ως ο BMU. Η Ευκλείδεια απόσταση δίνεται από το μαθηματικό τύπο:

$$\sqrt{\sum_{j=0}^{j=n} (V_j - W_j)^2}$$

Εξίσωση 3.1

Όπου:

- n το μέγεθος του διανύσματος εισόδου και των βαρών του νευρώνα που ελέγχεται,
- Vj το j-οστό στοιχείο του διανύσματος εισόδου,
- Wj το j-οστό στοιχείο του διανύσματος των βαρών του κόμβου εξόδου που ελέγχεται

Για τον υπολογισμό της ακτίνας του BMU και πιο συγκεκριμένα των νευρώνων που ανήκουν στην «γειτονιά» (neighbourhood) του BMU, χρησιμοποιείται συνήθως η ακόλουθη συνάρτηση (μπορεί ο χρήστης του δικτύου να επιλέξει άλλη συνάρτηση):

$$r(\iota) = r_0 \times \exp\left(-\left(\frac{\iota}{\lambda}\right)\right)$$

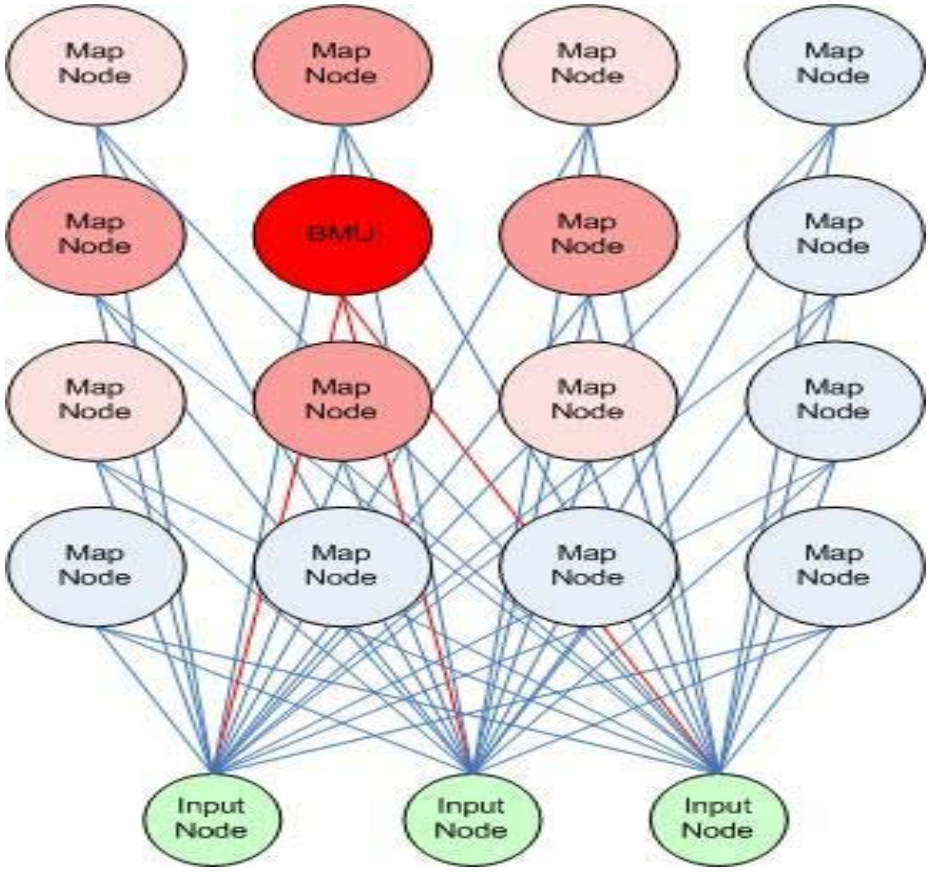
$\iota = 0, 1, 2, \ldots$

Εξίσωση 3.2

Όπου:

- ι ο αριθμός της εποχής που βρίσκεται το δίκτυο μας,
- r0 είναι η αρχική τιμή της ακτίνας (συνήθως έχει το μέγεθος του πλέγματος),
- r(i) είναι η τιμή της ακτίνας κατά την εποχή i,
- λ μια σταθερά που εξαρτάται από την ακτίνα (αριθμός εποχών που επιλέχθηκε/ log (r0)) [9]

Με βάση την τιμή της συνάρτησης αυτής, αυτό που μένει είναι ο έλεγχος για το ποιοι νευρώνες εξόδου ανήκουν σ' αυτή την ακτίνα. Είναι σημαντικό να παρατηρήσουμε ότι και εδώ μετριέται η ευκλείδεια απόσταση στη τοπολογία μας (π.χ. σε ένα δισδιάστατο πλέγμα η απόσταση ενός κόμβου από τον άλλο μπορεί να υπολογιστεί και με βάση το Πυθαγόρειο Θεώρημα). Έτσι έχοντας τη θέση του BMU στο δίκτυο και την τιμή της ακτίνας, μπορούμε εύκολα να προσδιορίσουμε το ποιοι κόμβοι ανήκουν στην «γειτονιά» του BMU. Επιπλέον, ένα μοναδικό χαρακτηριστικό των δικτύων Kohonen είναι ότι η ακτίνα σε κάθε εποχή (επανάληψη) μειώνεται και αυτό επιτυγχάνεται προσθέτοντας στον υπολογισμό της ακτίνας το μέρος exp(- i) . Αν ο χρήστης επέλεξε αρκετό αριθμό εποχών τότε η ακτίνα θα καταλήξει με μέγεθος 1 που θα είναι μόνο ο νευρώνας BMU κάτι που θα σημαίνει ότι έχουμε πετύχει πολύ καλή κατηγοριοποίηση (Βλέπε Εικόνα 3.2). [9] [10] Επίσης πρέπει να γίνει ιδιαίτερη αναφορά στην χρήση της σταθεράς λ. Η σταθερά λ δίνει την δυνατότητα στο χρήστη να ελέγξει το ρυθμό της πτώσης της ακτίνας. Με τον προσδιορισμό της σταθεράς αυτής (μπορεί να οριστεί απευθείας και όχι με βάση το τύπο) ο χρήστης του δικτύου μπορεί να ελέγξει την πτώση της ακτίνας έτσι ώστε να γίνει με πιο ομαλό τρόπο. Με πολύ μεγάλη τιμή στο λ η ακτίνα μειώνεται πιο αργά, το δίκτυο αργεί να συγκλίνει, χρειάζεται μεγαλύτερος αριθμός εποχών αλλά υπάρχει μεγαλύτερη πιθανότητα εύρεσης της βέλτιστης λύσης δηλαδή της κατηγοριοποίησης. Μικρότερη τιμή σημαίνει πιο απότομη πτώση στην ακτίνα, λιγότερος υπολογισμός, γρηγορότερη σύγκλιση αλλά πιθανώς να μην βρεθεί η βέλτιστη λύση



Εικόνα 3.2 Επιλογή BMU και ακτίνας ενός Kohonen Δικτύου παρθέν από 'Generation5' (http://www.generation5.org/content/2004/aisompic.asp?Print=1)

Έχοντας υπολογίσει το σύνολο των κόμβων που βρίσκονται εντός της ακτίνας του BMU (συμπεριλαμβανομένου και του BMU), θα γίνουν οι αλλαγές στα βάρη με βάση την πιο κάτω συνάρτηση:

$$\overrightarrow{W_{i+1}} = \overrightarrow{Wi} + \Theta(i)L(i)\left(\overrightarrow{V_i} - \overrightarrow{W_i}\right)$$

$$i = 0,\ 1,\ 2,\ \ldots$$

Εξίσωση 3.3

Όπου:

- i ο αριθμός των εποχών που βρισκόμαστε
- Wi διάνυσμα βαρών για την εποχή i
- Wi+1 το διάνυσμα βαρών για την εποχή i+1
- Vi το δεδομένο διάνυσμα εισόδου για την εποχή i
- L(i) είναι το learning rate για την εποχή i
- $\Theta(i)$ η τιμή που καθορίζει το πόσο η απόσταση ενός νευρώνα από το BMU επηρεάζει την αλλαγή των βαρών του [9]

Το learning rate L(i) είναι συνήθως μια μικρή τιμή συνήθως μεταξύ 0.1 – 0.9, που μειώνεται όσο αυξάνεται ο αριθμός των επαναλήψεων. Ουσιαστικά ορίζει το ρυθμό με τον οποίο θα μεταβληθούν τα βάρη του BMU καθώς των νευρώνων που ανήκουν στην γειτονιά του. Με μικρή τιμή στο learning rate το δίκτυο αργεί να συγκλίνει και συνεπώς απαιτείται μεγαλύτερος αριθμός εποχών που κοστίζει αρκετά. Συνήθως αυτή την διαδικασία την επιθυμούμε όταν το δίκτυο κάνει την «γενική» κατηγοριοποίηση του και επιθυμούμε το πλήρη διαχωρισμό των κατηγοριών και εμφάνιση των κοινών τους χαρακτηριστικών. Γενικά θέτοντας μεγάλη τιμή στο learning rate επιτρέπουμε στο δίκτυο να προχωρήσει/ συγκλίνει πιο γρήγορα, όμως οι πιθανότητες για να μην βρεθεί η βέλτιστη λύση αυξάνονται δηλαδή το clustering που θα γίνει δεν θα είναι το ιδανικό αφού θα βρει μεν τις κατηγορίες αλλά δεν θα αρκετά ξεκάθαρες. Επιπλέον απαιτείται σε κάθε εποχή η μείωση του learning rate έτσι ώστε οι αλλαγές να γίνονται σε μικρότερο βαθμό μετά από κάθε εποχή, ώστε οι περιοχές του χάρτη να γίνουν πιο ξεκάθαρες οδηγώντας σε ένα καλύτερο συντονισμό των νευρώνων και τελικά σε καλύτερες λύσεις λαμβάνοντας έτσι τα πλεονεκτήματα τόσο μιας μεγάλης τιμής στο learning rate(clustering) τόσο και μιας μικρής τιμής (fine tuning).

$$L(i) = L_0 \times \exp\left(-\frac{i}{\lambda}\right)$$

ι = 0, 1 , 2 ,…

Εξίσωση 3.4

Όπου:

- i ο αριθμός των εποχών που βρισκόμαστε
- λ μια σταθερά που εξαρτάται από την ακτίνα (αριθμός εποχών που περάσαν/αριθμός εποχών)
- L_0 η αρχική learning rate

Η τιμή $\Theta(i)$ εξαρτάται από την γραμμική απόσταση (απόσταση στην τοπολογία) μεταξύ του BMU και του κάθε νευρώνα. Ουσιαστικά καθορίζει το πόσο μεγάλη θα είναι η αλλαγή των βαρών του συγκεκριμένου νευρώνα βάση της απόστασης του κόμβου από τον BMU. Με αυτό το τρόπο ο BMU κάνει τη μεγαλύτερη αλλαγή στα βάρη του ώστε να πλησιάσει όσο το δυνατό περισσότερο στο διάνυσμα εισόδου, ενώ οι νευρώνες που ανήκουν στη γειτονιά του όσο πιο μακριά του βρίσκονται να κάνουν μικρότερες αλλαγές στα βάρη τους.

$$\Theta(i) = \exp\left(-\frac{d^2}{2 \times r^2(i)}\right)$$

ι = 0, 1 , 2 ,…

Εξίσωση 3.5

Όπου:

- i ο αριθμός των εποχών που βρισκόμαστε
- r(i) η τιμή που υπολογίζεται στην συνάρτηση 2
- d η απόσταση του συγκεκριμένου κόμβου από τον BMU που υπολογίζεται στο βήμα 5 του αλγορίθμου.

[9] [11]

3.3.3 Αρχιτεκτονική Παράλληλου SOM

Η αρχιτεκτονική του παράλληλου SOM είναι η ίδια με του σειριακού. Οι συνδεσμολογία, το πλέγμα και το διάνυσμα εισόδου έχουν την ίδια μορφή με το σειριακό SOM και το μόνο που αλλάζει είναι το τι κάνει ο κάθε επεξεργαστής στο δίκτυο.

3.3.4 Εκπαίδευση Παράλληλου SOM

Η γενική ιδέα για την παραλληλοποίηση του αλγορίθμου, είναι ο διαμοιρασμός των εργασιών στους επεξεργαστές που έχουμε διαθέσιμους (domain decomposition). Αυτό επιτεύχθηκε διαμοιράζοντας στους επεξεργαστές τους νευρώνες του δισδιάστατου πλέγματος. Ουσιαστικά, τώρα κάθε επεξεργαστής ελέγχει τους δικούς του νευρώνες με το διάνυσμα εισόδου, βρίσκει τον BMU για τους δικούς του νευρώνες, και στη συνέχεια όλοι οι επεξεργαστές συγκρίνουν τους BMU τους και επιλέγεται ο καλύτερος. Αφού επιλεχθεί ο BMU του δικτύου για το συγκεκριμένο διάνυσμα εισόδου, κάθε επεξεργαστής προσαρμόζει τα βάρη των νευρώνων του που ανήκουν στην γειτονιά του BMU. Οι εξισώσεις που χρησιμοποιήθηκαν για την εκπαίδευση του δικτύου είναι οι ίδιες με αυτές στο σειριακό SOM.

Συγκεκριμένα ο αλγόριθμος είναι ο εξής:

1. Αρχικοποίησε τα βάρη των νευρώνων του δισδιάστατου πλέγματος με τυχαίες μικρές τιμές. Αρχικοποίησε το μέγεθος της γειτονιάς (r_0) και το ρυθμό μάθησης (L_0).
2. Διαμοίρασε τους νευρώνες του δισδιάστατου πλέγματος, ώστε κάθε επεξεργαστής να έχει υπό την ευθύνη του τον ίδιο αριθμό νευρώνων με τους άλλους επεξεργαστές.
3. Επέλεξε τυχαία ένα διάνυσμα εισόδου από το σύνολο διανυσμάτων που έχουμε ως δεδομένα και θέλουμε να κατηγοριοποιήσουμε.

4. Κάθε επεξεργαστής συγκρίνει τα βάρη των νευρώνων του, με το αντίστοιχο στοιχείο του διανύσματος εισόδου και υπολογίζει την ευκλείδεια απόσταση μεταξύ των βαρών και του διανύσματος εισόδου.
5. Κάθε επεξεργαστής βρίσκει τον νευρώνα του με την μικρότερη απόσταση και τον ορίζει ως τον BMU του.
6. Όταν όλοι οι επεξεργαστές βρουν τον BMU τους (γίνεται ένας συγχρονισμός μεταξύ των επεξεργαστών), συγκρίνουν τους BMU τους και αυτός με τη μικρότερη απόσταση από το διάνυσμα εισόδου, επιλέγεται ως ο BMU του δικτύου. Εδώ γίνεται ακόμη ένας συγχρονισμός μεταξύ των επεξεργαστών, ώστε να βεβαιωθούμε ότι έχουν όλοι βρει τον BMU του δικτύου.
7. Κάθε επεξεργαστής, βρίσκει τους νευρώνες του που ανήκουν στη γειτονιά του BMU και προσαρμόζει τα βάρη τους βάση της Gaussian συνάρτησης.
8. Μείωσε το ρυθμό μάθησης L(i) και το μέγεθος της γειτονιάς r(i) και αύξησε την εποχή που βρισκόμαστε.
9. Επανέλαβε από το βήμα 2 μέχρι να εκτελεστεί ο επιθυμητός αριθμός εποχών.
10. Τρέξε μια φορά τον αλγόριθμο περνώντας όλα τα δεδομένα με τη σειρά χωρίς να κάνεις αλλαγές στα βάρη και έχοντας ως μέγεθος της ακτίνας το 1. Αύξησε το μετρητή κάθε νευρώνα που επιλέγεται ως BMU και σημείωσε πιο δεδομένο τον ενεργοποίησε.
11. Δημιούργησε ένα αρχείο με το μετρητή κάθε νευρώνα και τη θέση του στο πλέγμα, ένα αρχείο με τα δεδομένα που ενεργοποίησαν κάθε νευρώνα στο βήμα 10 και ένα αρχείο με τις τιμές των βαρών κάθε νευρώνα.

Ο αλγόριθμος υλοποιήθηκε σε c++ και ο παραλληλισμός έγινε με τη χρήση της βιβλιοθήκης pthreads.h, η οποία είναι για τη δημιουργία threads σε Unix συστήματα.

Ο συγχρονισμός των επεξεργαστών υλοποιήθηκε χρησιμοποιώντας τον αμοιβαίο αποκλεισμό που προσφέρουν τα pthreads. Σε κάθε κρίσιμο σημείο που δημιουργείται δεν μπορούν να μπουν πέραν του ενός επεξεργαστή.

Συγκεκριμένα ο ψευδοκώδικας είναι ο εξής:

```
Synchronize ()
{
        lock( mutex3 );
        unlock( mutex3 );
        lock( mutex1 );
```

```
            ++processCounter;
            if(processCounter == 1)
                  lock( mutex2 );
            if(processCounter == numberOfcores)
            {
                  lock(mutex3 );
                  unlock( mutex2 );
            }
      unlock( mutex1 );

      lock( mutex2 );
            --processCounter;
            if(processCounter == 0)
                  unlock( mutex3 );
      unlock( mutex2 );
}
```

Ουσιαστικά υπάρχουν 3 κρίσιμα σημεία. Για το παράδειγμα θα θεωρήσουμε ότι υπάρχουν 2 επεξεργαστές και είναι στο βήμα 6 του αλγόριθμου.

a. Οι δύο επεξεργαστές έχουν υπολογίσει το BMU τους και περνάνε στη φάση του συγχρονισμού.
b. Την πρώτη φορά που θα περάσουν οι επεξεργαστές από το mutex3 δεν θα σταματήσουν αλλά απλά θα περάσει ο κάθε ένας με τη σειρά του.
c. Ο πρώτος που θα κλειδώσει το mutex1 θα αυξήσει τον μετρητή από 0 και θα τον κάνει 1 και επομένως θα ικανοποιεί την συνθήκη και θα κλειδώσει το mutex2, θα ξεκλειδώσει το mutex1 και θα πάει στο mutex2 όπου θα περιμένει γιατί είναι κλειδωμένο και δεν μπορεί να περάσει.
d. Ο δεύτερος επεξεργαστής, θα κλειδώσει το mutex1 και θα αυξήσει το μετρητή κατά 1 και θα πάρει την τιμή 2. Αυτό σημαίνει ότι ικανοποιεί τη δεύτερη συνθήκη, επομένως θα κλειδώσει το mutex3 και θα ξεκλειδώσει το mutex2. Στη συνέχεια θα ξεκλειδώσει και το mutex1.
e. Ο πρώτος επεξεργαστής, τώρα μπορεί να περάσει στο mutex2 και θα μειώσει το μετρητή ο οποίος θα πάρει την τιμή 1. Όταν βγει από το mutex2 θα ελέγξει την τιμή του BMU του με την τιμή του BMU του άλλου επεξεργαστή, και θα επιλέξει ως BMU του δικτύου εκείνο με την μικρότερη απόσταση από το διάνυσμα εισόδου. Όταν

προσπαθήσει να περάσει στον επόμενο συγχρονισμό, τα περιμένει στο mutex3 γιατί είναι κλειδωμένο από προηγουμένως.

f. Ο δεύτερος επεξεργαστής μπήκε στο mutex2 και μείωσε τη τιμή του μετρητή κατά ένα και πήρε την τιμή 0. Ικανοποιείται η συνθήκη και ξεκλειδώνει το mutex3. Υπολογίζει το BMU του δικτύου όπως ο πρώτος επεξεργαστής και προχωρά στον επόμενο συγχρονισμό.
g. Στον δεύτερο συγχρονισμό επαναλαμβάνονται τα βήματα και διασφαλίζει ότι όλοι οι επεξεργαστές έχουν υπολογίσει τον BMU του δικτύου.

Αν δεν υπήρχε το mutex3, τότε ο πρώτος επεξεργαστής στο βήμα e θα έμπαινε στο mutex1 του επόμενου συγχρονισμού και θα μπορούσε να μεταβάλει την τιμή του μετρητή. Αυτό θα μπορούσε να οδηγήσει σε deadlock λόγο του ότι η τιμή του μετρητή θα γινόταν 2 και με αποτέλεσμα να μην ικανοποιηθεί η συνθήκη και να μην κλειδωθεί το mutex2 από τον πρώτο επεξεργαστή και να μην μπορούν να συγχρονιστούν στη συνέχεια οι επεξεργαστές.

Οι παράμετροι του δικτύου επιλέγηκαν εμπειρικά, κάνοντας διάφορες δοκιμές του δικτύου και του κατά πόσο εκπαιδεύεται με διάφορες τιμές στις παραμέτρους του. Για τον έλεγχο του κατά πόσο εκπαιδεύτηκε το δίκτυο έγινε έλεγχος με το σφάλμα κβαντισμού, το οποίο ανέλαβε ο κύριος Λοΐζος Λοΐζου και ελέγχοντας και το κατά πόσο μεταβάλλονται οι κυριότερες κατηγοριοποιήσεις. Τα δεδομένα εισόδου για τους ελέγχους και την εκπαίδευση είναι σε 3 αρχεία. Το ένα περιέχει μόνο δείγματα από τους ασθενείς, το δεύτερο περιέχει τα δείγματα από τους υγιείς και το τρίτο περιέχει όλα τα δείγματα.

3.3.5 Βελτιστοποιήσεις αλγόριθμου

Παρά το γεγονός ότι η παραλληλοποίηση του αλγορίθμου θα μείωνε το χρόνο εκτέλεσης του αλγορίθμου, έγιναν κάποιες περαιτέρω προσπάθειες για βελτιστοποίηση του αλγόριθμου. Η προσπάθεια βελτιστοποίησης έγινε σε σημεία που γίνονταν πολλοί υπολογισμοί και στα σημεία συγχρονισμού των επεξεργαστών.

3.3.5.1 Συγχρονισμός επεξεργαστών

Ο διπλός συγχρονισμός στο βήμα 6 του SOM είναι απαραίτητος για την ορθή λειτουργία του αλγορίθμου. Για να γλιτώσουμε όμως αυτό το διπλό συγχρονισμό, μπορούμε να εκμεταλλευτούμε τον ένα συγχρονισμό και βάλουμε τις εργασίες που έχουμε να κάνουμε μέσα στον συγχρονισμό. Αυτό μπορεί να γίνει γιατί το mutex2 παραμένει κλειδωμένο μέχρι

όλοι οι επεξεργαστές να περάσουν από το mutex1. Ο ψευδοκώδικας για την υλοποίηση φαίνεται πιο κάτω.

```
synchronizeAndFindBMU()
{
        lock( mutex3 );
        unlock(mutex3 );

        ΥΠΟΛΟΓΙΣΕ ΤΟ BMU ΤΟΥ ΕΠΕΞΕΡΓΑΣΤΗ

        lock( mutex1 );
                ++processCounter;
                if(processCounter == 1)
                        lock( mutex2 );
                if(processCounter == numberOfcores)
                {
                        lock(mutex3 );
                        unlock( mutex2 );
                }
        unlock(mutex1 );
        lock( mutex2 );

                ΥΠΟΛΟΓΙΣΕ ΤΟ BMU ΤΟΥ ΔΙΚΤΥΟΥ

                --processCounter;
                if(processCounter == 0)
                        unlock( mutex3 );
        unlock( mutex2 );
}
```

Για τον έλεγχο της βελτιστοποίησης αυτής της πρότασης, έτρεξα τα πιο κάτω 3 σενάρια με τον παράλληλο αλγόριθμο με τους δύο συγχρονισμούς και με τον παράλληλο αλγόριθμο με τον ένα συγχρονισμό δημιουργώντας 12 threads.

Το πόσο πιο γρήγορη είναι η εφαρμογή με την βελτιστοποίηση σε σχέση με την εφαρμογή χωρίς τη βελτιστοποίηση θα υπολογιστεί με την πιο κάτω εξίσωση.

$$\text{Βαθμός βελτιστοποίησης} = \frac{\text{Χρόνος Εκτέλεσης Εφαρμογής χωρίς βελτιστοποίηση}}{\text{Χρόνος Εκτέλεσης Εφαρμογής με βελτιστοποίηση}}$$

Εξίσωση 3.6

Για είσοδο χρησιμοποίησα ένα αρχείο με 327 SNPs και 1853 subjects.

Σενάριο 1:

Μέγεθος Πλέγματος: 20 * 20

Μέγεθος Ακτίνας: 10

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

Σενάριο 2:

Μέγεθος Πλέγματος: 50 * 50

Μέγεθος Ακτίνας: 25

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

Σενάριο 3:

Μέγεθος Πλέγματος: 100 * 100

Μέγεθος Ακτίνας: 50

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

3.3.5.2 Προϋπολογισμός αποστάσεων

Στη βήμα 7 του SOM, ο κάθε επεξεργαστής πρέπει να αλλάξει τα βάρη των νευρώνων του που ανήκουν στη γειτονιά του BMU, και για να γίνει αυτό, χρειάζεται κάθε φορά να υπολογίζει την απόσταση κάθε νευρώνα του από τον BMU. Για να γλιτώσουμε όλες αυτές τις πράξεις, προϋπολογίζουμε τις αποστάσεις στην αρχή του αλγορίθμου και τις αποθηκεύουμε σε ένα δισδιάστατο πίνακα που έχει το μέγεθος του δισδιάστατου πλέγματος του δικτύου και σε κάθε θέση μπορεί να αποθηκεύσει δύο τιμές. Ο πιο κάτω ψευδοκώδικας υπολογίζει και αποθηκεύει τις αποστάσεις στον πίνακα distances και την απόσταση υψωμένη στη δευτέρα, η οποία χρειάζεται για τον υπολογισμό του πόσο μεγάλη θα είναι η αλλαγή στα βάρη του νευρώνα.

Για κάθε θέση i,j του δισδιάστατου πίνακα

{

Φύλαξε το αποτέλεσμα του sqrt((i*i)+(j*j)) και το (sqrt((i*i)+(j*j)))^2

}

Στη συνέχεια όταν θα πρέπει να ελεγχθεί κατά πόσο ο νευρώνας i ανήκει στην γειτονιά του BMU, τότε απλά πρέπει να ελέγξεις αν ή απόσταση στη θέση X.Y του πίνακα distances είναι μικρότερη της ακτίνας της γειτονιάς. Το X και το Y υπολογίζονται με τον εξής τρόπο:

$X = |Xi - X_{BMU}|$

$Y = |Yi - Y_{BMU}|$

Όπου Xi και Yi η θέση του νευρώνα i στο δισδιάστατο πλέγμα και $X_{BMU,}$ Y_{BMU} η θέση του BMU στο δισδιάστατο πλέγμα.

Για τον έλεγχο της βελτιστοποίησης αυτής της πρότασης, έτρεξα τα πιο κάτω 3 σενάρια με τον παράλληλο αλγόριθμο χωρίς τον προϋπολογισμό των αποστάσεων και με τον παράλληλο αλγόριθμο με τον προϋπολογισμό των αποστάσεων δημιουργώντας 12 threads.

Το πόσο πιο γρήγορη είναι η εφαρμογή με την βελτιστοποίηση σε σχέση με την εφαρμογή χωρίς τη βελτιστοποίηση θα υπολογιστεί με την πιο κάτω εξίσωση.

$$\text{Βαθμός βελτιστοποίησης} = \frac{\text{Χρόνος Εκτέλεσης Εφαρμογής χωρίς βελτιστοποίηση}}{\text{Χρόνος Εκτέλεσης Εφαρμογής με βελτιστοποίηση}}$$

Εξίσωση 3.7

Για είσοδο χρησιμοποίησα ένα αρχείο με 327 SNPs και 1853 subjects.

Σενάριο 1:

Μέγεθος Πλέγματος: 20 * 20

Μέγεθος Ακτίνας: 10

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

Σενάριο 2:

Μέγεθος Πλέγματος: 50 * 50

Μέγεθος Ακτίνας: 25

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

Σενάριο 3:

Μέγεθος Πλέγματος: 100 * 100

Μέγεθος Ακτίνας: 50

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

3.3.5 Σύγκριση χρόνων σειριακού και παράλληλου αλγόριθμου

Το speedup είναι το πόσο πιο γρήγορο είναι ένα παράλληλο πρόγραμμα από ένα σειριακό, και υπολογίζεται από την εξής πράξη:

$$Speedup(p) = \frac{T1}{Tp}$$

Εξίσωση 3.8

Όπου:

T1 = χρόνος εκτέλεση σειριακού αλγόριθμου

Tp = χρόνος εκτέλεσης παράλληλου αλγόριθμου

p = αριθμός επεξεργαστών

Για το υπολογισμό του, έτρεξα τον σειριακό και τον παράλληλο αλγόριθμο, σε μια μηχανή με 12 επεξεργαστές. Ουσιαστικά υπάρχουν δύο chips AMD Opteron 2427 όπου το κάθε chip περιέχει έξι πυρήνες με συχνότητα στα 800MHz και μέγεθος cache 512 KB. Το χρόνο τον πήρα με τη χρήση του time των Unix και χρησιμοποίησα το χρόνο που αναγράφει στο real. Στις δοκιμές που έκανα, οι παράμετροι ήταν κοινές για τους δύο αλγόριθμους και η διαφορά ήταν ότι στον παράλληλο αλγόριθμο μετέβαλλα τον αριθμό των threads που χρησιμοποιούσε.

Συγκεκριμένα, έτρεξα και τους δύο αλγόριθμους με τα εξής 3 διαφορετικά σενάρια.

Για είσοδο χρησιμοποίησα ένα αρχείο με 327 SNPs και 1853 subjects.

Σενάριο 1:

Μέγεθος Πλέγματος: 20 * 20

Μέγεθος Ακτίνας: 10

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

Σενάριο 2:

Μέγεθος Πλέγματος: 50 * 50

Μέγεθος Ακτίνας: 25

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

Σενάριο 3:

Μέγεθος Πλέγματος: 100 * 100

Μέγεθος Ακτίνας: 50

Αριθμός Εποχών: 100

Ρυθμός Μάθησης: 0.1

Όσο αφορά τον παράλληλο αλγόριθμο, έτρεξα το κάθε σενάριο με 1, 2, 4, 8, 12 και 24 threads.

3.4 Μεταεπεξεργασία

Για να μπορέσουμε να αντιληφθούμε τα αποτελέσματα που θα βγάλει ο SOM, θα πρέπει να τα οπτικοποιήσουμε, ώστε να γίνουν πιο εύκολα κατανοητά. Για το λόγο αυτό θα δημιουργηθούν γραφικές παραστάσεις που θα δείχνουν την κατηγοριοποίηση που έγινε. Επίσης επειδή ο αλγόριθμος στην αρχή αρχικοποιεί τα βάρη με τυχαίες τιμές, θα πρέπει να ελεγχθεί ότι οι κατηγοριοποιήσεις που κάνει είναι έγκυρες και ότι δεν είναι διαφορετικές σε κάθε εκτέλεσή του.

3.4.1 Παρουσίαση κατηγοριοποίησης

Η απεικόνιση της κατηγοριοποίησης θα γίνει μέσω γραφική παράστασης, όπου θα φαίνεται ο αριθμός των δεδομένων που ενεργοποίησαν κάθε νευρώνα στο βήμα 10 του αλγόριθμου. Συγκεκριμένα, ο άξονας X και ο άξονας Y θα αντιπροσωπεύουν τη θέση που είχε κάθε νευρώνας στο δισδιάστατο πλέγμα (π.χ. το σημείο με X = 0 και Y = 0 θα αντιπροσωπεύει τον νευρώνα που είχε τη θέση 0.0 στο δισδιάστατο πλέγμα.). Σε κάθε σημείο X.Y στη γραφική όπου X = 0, 1, 2 ..., N και Y = 0, 1, 2 ..., N θα υπάρχει ένα τετράγωνο το οποίο θα

χρωματίζεται ανάλογα με τον αριθμό των δεδομένων που ενεργοποίησαν εκείνο το νευρώνα κατά το βήμα 10 του παράλληλου αλγόριθμου.

3.4.2 Έλεγχος συνέπειας αποτελεσμάτων

Για να ελεγχθεί το κατά πόσο τα αποτελέσματα του δικτύου δεν εξαρτώνται από τις τυχαίες τιμές που παίρνουν τα βάρη των νευρώνων κατά την αρχικοποίησή τους, πρέπει να ελέγξουμε τις κατηγοριοποιήσεις που κάνει το δίκτυο. Για να γίνει αυτό, έτρεξε ο SOM 10 φορές και ελέγχθηκαν τα δεδομένα που ενεργοποίησαν κάθε νευρώνα στο βήμα 10 του SOM. Αφού συγκεντρώθηκαν τα αποτελέσματα των εκτελέσεων σε ένα αρχείο, έγινε έλεγχος για εύρεση κατηγοριών που μεταξύ τους είχαν κάποιο ποσοστό ομοιότητας. Αν οι σημαντικές κατηγορίες που προκύψουν από τον SOM είναι κοινές στις περισσότερες εκτελέσεις, τότε σημαίνει ότι τα αποτελέσματα είναι συνεπή.

Ο αλγόριθμος για τον έλεγχο είναι ο εξής:

1. Πάρε την πρώτη διαθέσιμη κατηγοριοποίηση.
2. Έλεγξε την ομοιότητα της κατηγοριοποίησης αυτής με όλες τις άλλες διαθέσιμες κατηγοριοποιήσεις.
3. Αν μια κατηγορία έχει P ομοιότητα μαζί της, ομαδοποίησέ τις και υπολόγισε το μέγιστο και ελάχιστο αριθμό subjects μεταξύ των κατηγοριών.
4. Θέσε ως μη διαθέσιμες τις κατηγοριοποιήσεις που ομαδοποίησες
5. Επανέλαβε από το βήμα 1 όσο υπάρχουν διαθέσιμες κατηγορίες
6. Δημιούργησε ένα αρχείο με τις ομαδοποιήσεις.

Όπου P το ποσοστό ομοιότητας που θέλουμε να έχουν οι κατηγοριοποιήσεις.

3.4.3 Σύγκριση γραμμικής και δυαδικής κωδικοποίησης

Συγκρίθηκαν οι κατηγορίες που προέκυψαν στις 10 εκτελέσεις κάθε κωδικοποίησης με τη χρήση του πιο πάνω αλγόριθμου για κάθε σενάριο ξεχωριστά. Αφού συγκεντρώθηκαν τα αποτελέσματα των εκτελέσεων σε ένα αρχείο, έγινε έλεγχος για εύρεση κατηγοριών που μεταξύ τους είχαν κάποιο ποσοστό ομοιότητας. Αν οι σημαντικές κατηγορίες που προκύψουν από τον SOM είναι κοινές στις περισσότερες εκτελέσεις, τότε σημαίνει ότι οι δύο κωδικοποιήσεις δίνουν τα ίδια αποτελέσματα.

Κεφάλαιο 4

Αποτελέσματα

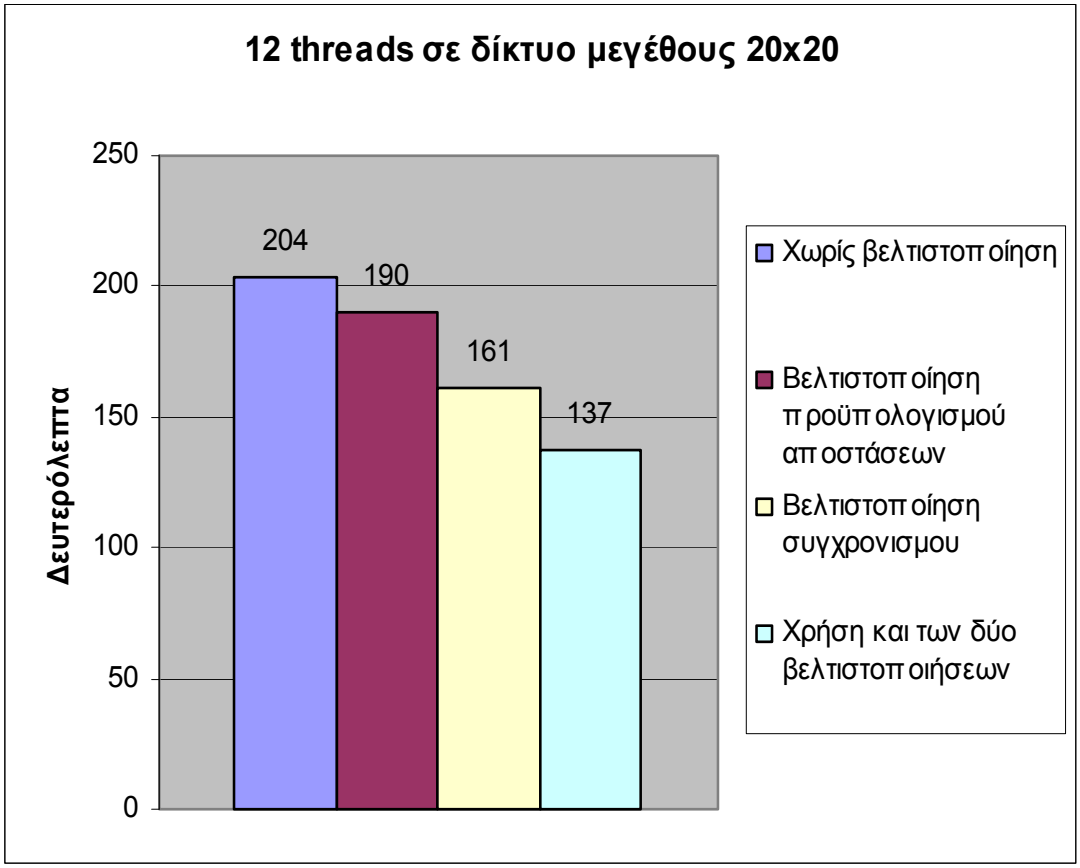
4.1 Επιλογή επιθυμητών δεδομένων

Η επιλογή των SNPs με επίσταση μεγαλύτερη του 45 είχε ώς αποτέλεσμα να μειώσει τον αριθμό τους. Συγκεκριμένα τα αρχικά 327 SNPs που θα χρησιμοποιούσαμε για την κατηγοριοποίηση μειώθηκαν σε 84.

4.2 Βελτιστοποιήσεις

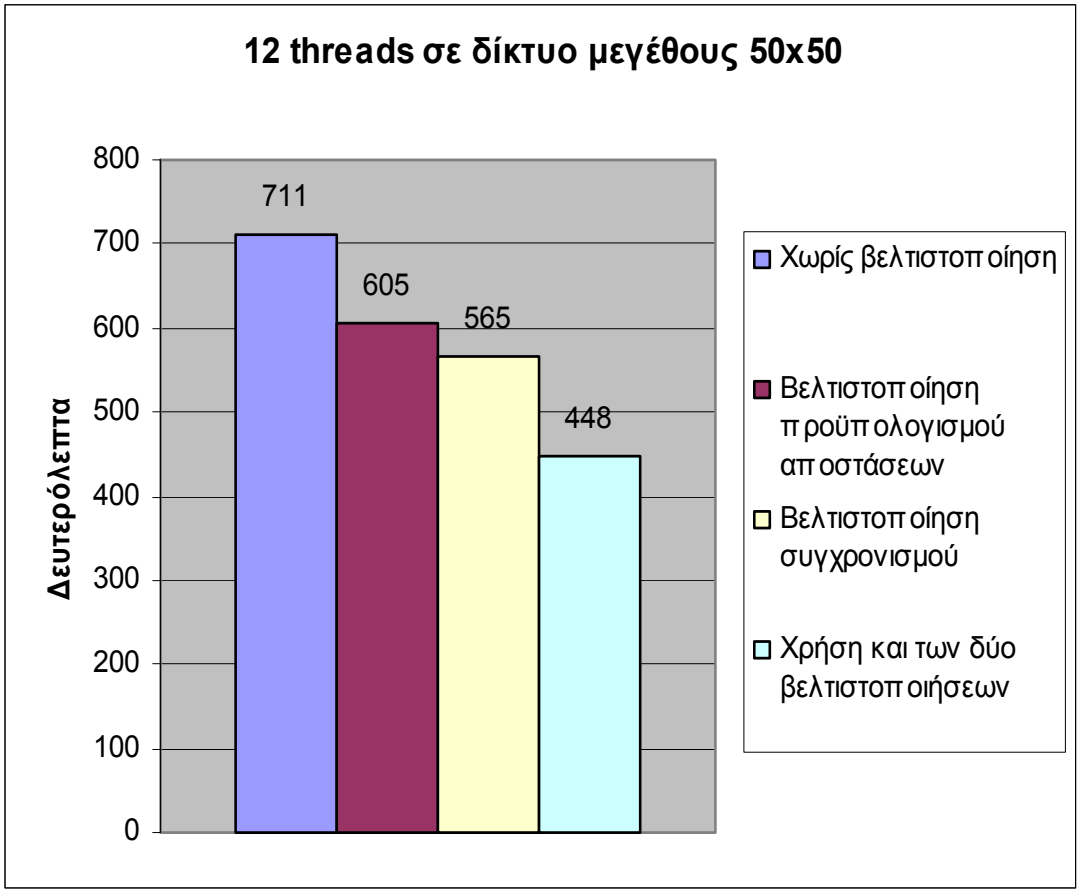
Οι βελτιστοποιήσεις έχουν συγκριθεί ανά μέγεθος δικτύου πόσο έχουν μειώσει το χρόνο εκτέλεσης του αλγόριθμου για τα συγκεκριμένα σενάρια. Η γενική εικόνα είναι ότι η βελτιστοποίηση του συγχρονισμού βοηθά περισσότερο στη μείωση του χρόνου εκτέλεσης,

και έχει πιο σταθερή συμβολή απ' ότι η βελτιστοποίηση με τον προϋπολογισμό των αποστάσεων. Επίσης η χρήση και των δύο βελτιστοποιήσεων φέρνει καλύτερα αποτελέσματα από τη χρήση μιας από αυτές αν και πάλι ο ρυθμός μείωσης του χρόνου εκτέλεσης δεν είναι σταθερός.



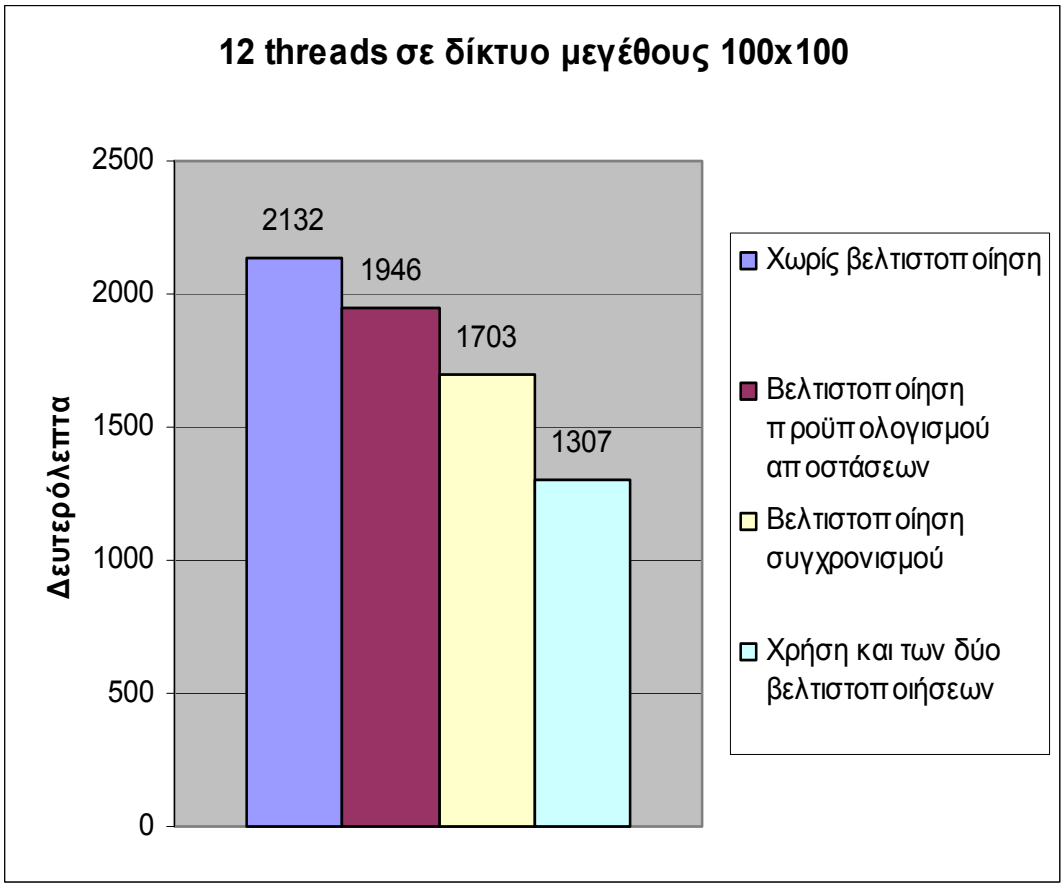
Σχήμα 4.1 Βελτιστοποίηση σε δίκτυο μεγέθους 20x20

Στο σχήμα 4.1 βλέπουμε το χρόνο που χρειάστηκε η εφαρμογή για να τρέξει στο δίκτυο με μέγεθος 20x20 με τις διάφορες βελτιστοποιήσεις. Στο σενάριο αυτό, η βελτιστοποίηση του προϋπολογισμού των αποστάσεων δεν μείωσε σημαντικά το χρόνο εκτέλεσης, και ο βαθμός βελτιστοποίησης της ήταν 1.07 σε αντίθεση με τη βελτιστοποίηση του συγχρονισμού που μείωσε το χρόνο κατά 40 περίπου δευτερόλεπτα και είχε βαθμό βελτιστοποίησης 1.26. Η χρήση και των δύο βελτιστοποιήσεων μείωσε το χρόνο εκτέλεσης κατά 65 περίπου δευτερόλεπτα και ήταν 1.48 φορές πιο γρήγορη η εφαρμογή από αυτή χωρίς βελτιστοποίηση.



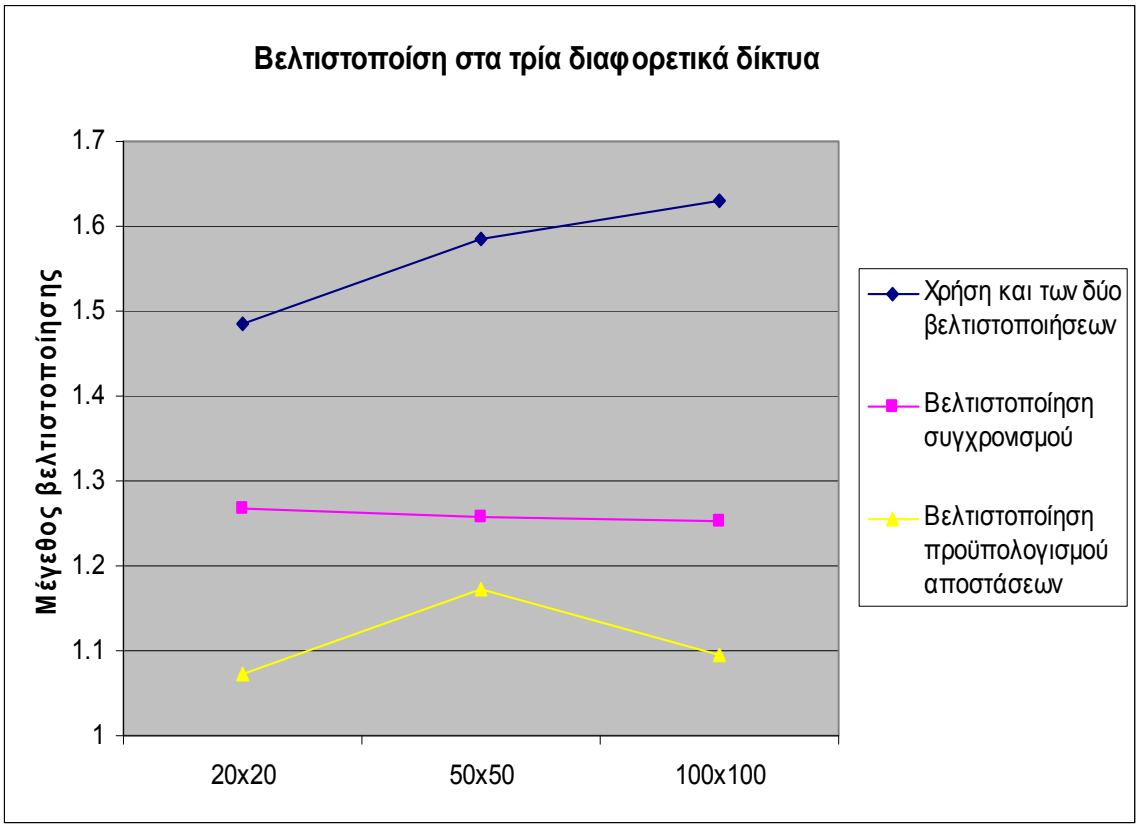
Σχήμα 4.2 Βελτιστοποίηση σε δίκτυο μεγέθους 50x50

Η βελτιστοποίηση της εφαρμογή στο σενάριο με το δίκτυο μεγέθους 50x50 φαίνεται στο σχήμα 4.2. Στο σενάριο αυτό, η βελτιστοποίηση προϋπολογισμού αποστάσεων είχε καλύτερα αποτελέσματα από το προηγούμενο σενάριο, μειώνοντας το χρόνο κατά 105 δευτερόλεπτα περίπου. Και σε αυτή την περίπτωση η βελτιστοποίηση του συγχρονισμού είχε καλύτερα αποτελέσματα, μειώνοντας το χρόνο κατά 145 δευτερόλεπτα περίπου, ενώ η χρήση και των δύο βελτιστοποιήσεων μείωσε το χρόνο στα 448 περίπου δευτερόλεπτα από τα αρχικά 711 δευτερόλεπτα. Συγκριτικά με την εφαρμογή χωρίς βελτιστοποιήσεις, αυτή με τον προϋπολογισμό αποστάσεων ήταν 1.17 φορές πιο γρήγορη και αυτή με το συγχρονισμό 1.26 φορές πιο γρήγορη, ενώ η χρήση και των δύο βελτιστοποιήσεων έκανε την εφαρμογή 1.59 φορές πιο γρήγορη.



Σχήμα 4.3 Βελτιστοποίηση σε δίκτυο μεγέθους 100x100

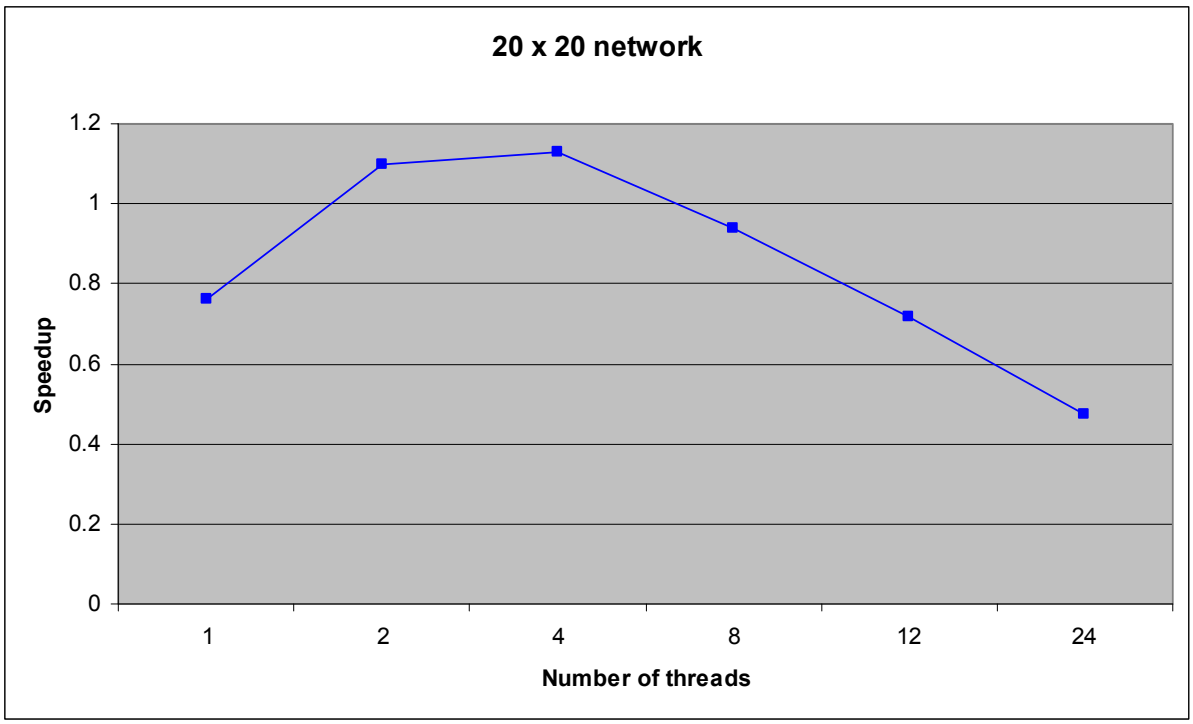
Στο τελευταίο και μεγαλύτερο δίκτυο, η εφαρμογή με τον προϋπολογισμό αποστάσεων ήταν 1.1 φορές πιο γρήγορη από την απλή εφαρμογή, μειώνοντας το χρόνο εκτέλεσης κατά 190 περίπου δευτερόλεπτα. Η εφαρμογή με τη βελτιστοποίηση του συγχρονισμού ήταν 1.25 φορές πιο γρήγορη μειώνοντας το χρόνο εκτέλεση κατά 430 περίπου δευτερόλεπτα. Τέλος η χρήση και των δύο βελτιστοποιήσεων και πάλι είχε τα καλύτερα αποτελέσματα, μειώνοντας το χρόνο εκτέλεσης κατά 825 περίπου δευτερόλεπτα, με αποτέλεσμα να είναι 1.63 φορές πιο γρήγορη από την απλή εφαρμογή.



Σχήμα 4.4 Βαθμός βελτιστοποίησης στα τρία διαφορετικά δίκτυα

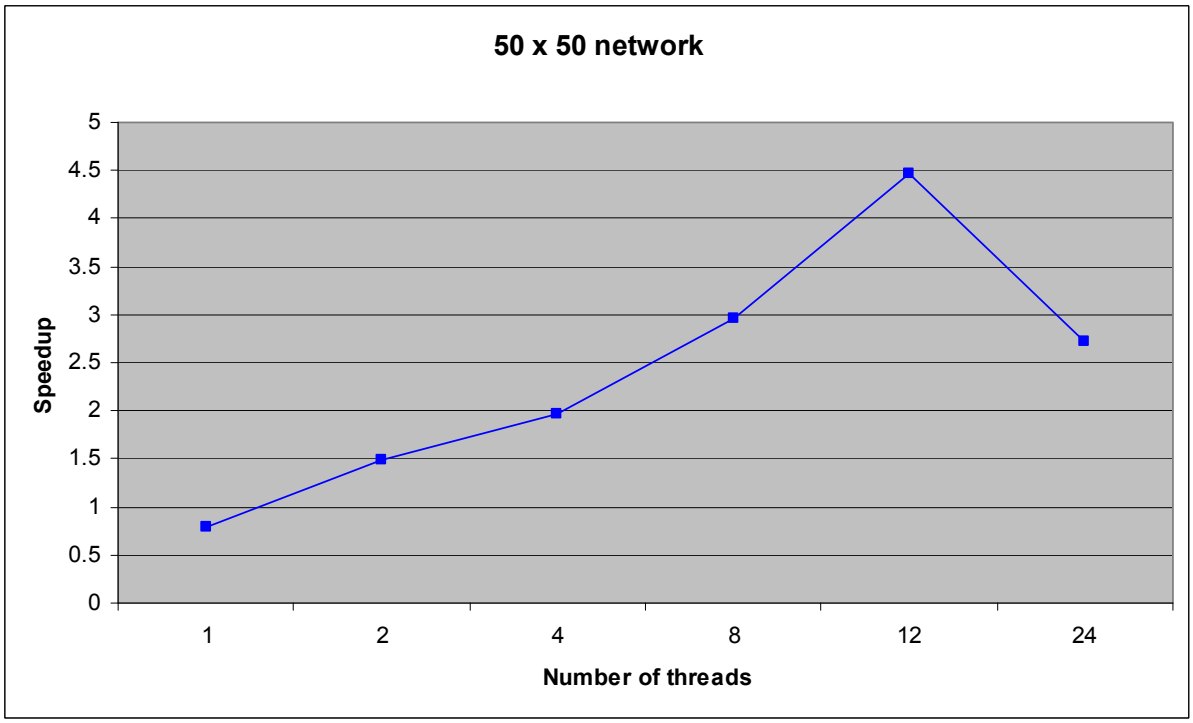
Στο σχήμα 4.4 βλέπουμε το πόσο πιο γρήγορη ήταν η κάθε εφαρμογή στα τρία σενάρια, από την εφαρμογή που δεν είχε καμία βελτιστοποίηση. Στη περίπτωση που χρησιμοποιήθηκαν και οι δύο βελτιστοποιήσεις, βλέπουμε να υπάρχει μια αύξηση της ταχύτητας εκτέλεσης της εφαρμογής όσο μεγαλώνει το σενάριο. Αυτό είναι κάτι που δεν θα έπρεπε να γίνεται κανονικά, αφού η εφαρμογή με τη βελτιστοποίηση του συγχρονισμού έχει σταθερή περίπου τιμή, ενώ η τιμή της βελτιστοποίησης με τον προϋπολογισμό μειώνεται στο τελευταίο σενάριο. Λογικά θα έπρεπε και η εφαρμογή με τις 2 βελτιστοποιήσεις να έχει πτώση στο τελευταίο σενάριο, αλλά αυτό είναι κάτι που θα εξηγηθεί στο επόμενο κεφάλαιο.

4.3 Σύγκριση χρόνων σειριακού και παράλληλου SOM



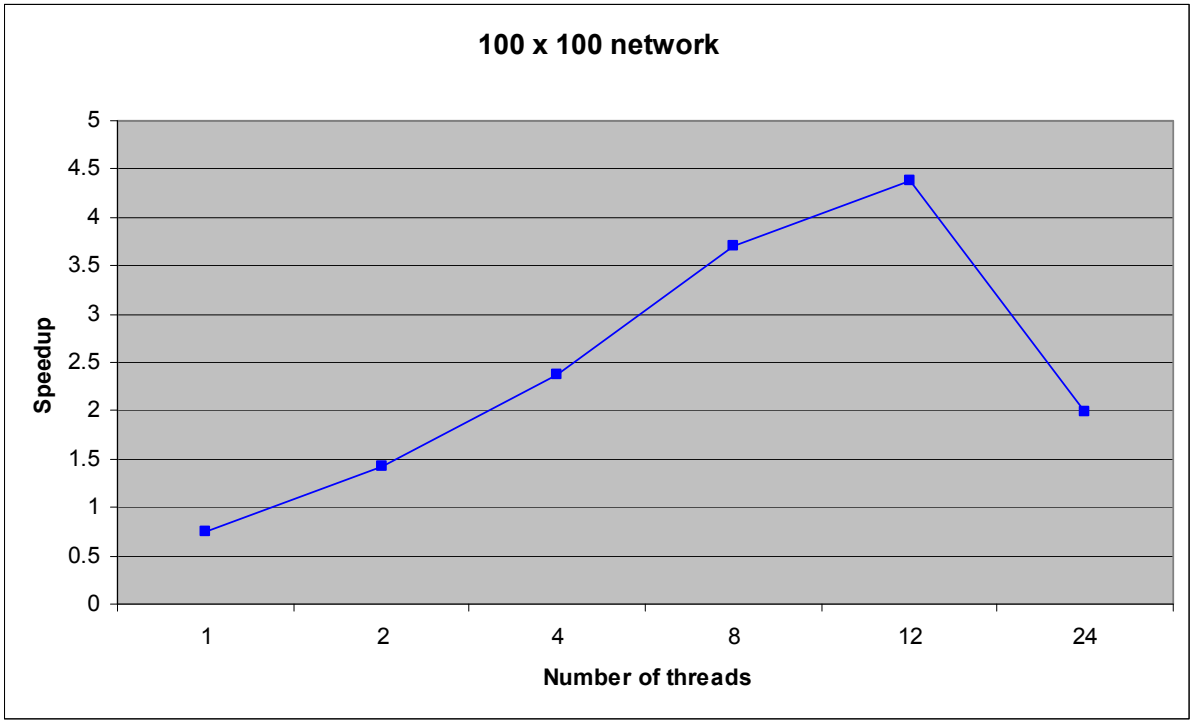
Σχήμα 4.5 Speedup παράλληλου SOM σε 20x20 δίκτυο

Στο σχήμα 4.5 βλέπουμε το speedup του παράλληλου SOM σε 20x20 δίκτυο. Με 1 thread είναι περίπου 0.75 φορές πιο αργός από το σειριακό ενώ με 2 και 4 threads έχει speedup 1.09 και 1.12 αντίστοιχα. Στη συνέχεια αυξάνοντας τα threads σε 8, 12 και 24, το speedup αρχίζει να μειώνεται και γίνεται 0.94, 0.72 και 0.47 αντίστοιχα. Σε αυτό το δίκτυο ο SOM κέρδισε πολύ λίγο χρόνο και μόνο για 2 και 4 threads.



Σχήμα 4.6 Speedup παράλληλου SOM σε 50x50 δίκτυο

Στο δεύτερο σενάριο (σχήμα 4.6), όπου αυξάνεται το μέγεθος του δικτύου σε 50x50, ο παράλληλος SOM παρουσιάζει καλύτερους χρόνους από τον σειριακό. Η μόνη εξαίρεση είναι όταν τρέχει με 1 thread, όπου το speedup είναι περίπου 0.78. Στη συνέχεια, αυξάνοντας τα threads, αυξάνεται και το speedup φτάνοντας το 4.47 με 12 threads. Η αύξηση του speedup δεν είναι γραμμική, αλλά αυξάνεται κάπως ανομοιόμορφα. Στα 2 threads έχουμε 1.48 speedup και στη συνέχεια με 4 και 8 threads 1.96 και 2.96 αντίστοιχα. Στα 12 threads, όπου χρησιμοποιούνται όλοι οι επεξεργαστές του υπολογιστή, υπήρξε μια σημαντική αύξηση του speedup από 2.96 σε 4.47. Όταν τα threads ξεπέρασαν τον αριθμό των επεξεργαστών και έγιναν 24, το speedup μειώθηκε στο 2.72.



Σχήμα 4.7 Speedup παράλληλου SOM σε 100x100 δίκτυο

Στο τρίτο δίκτυο με διαστάσεις 100x100 έχουμε καλύτερη αύξηση του speedup με την αύξηση των threads. Το speedup σε αυτό το σενάριο, ξεκινά με 0.74 και στη συνέχεια αυξάνεται σε 1.43, 2.37, 3.70 και 4.39 με 1, 2, 4, 8 και 12 threads αντίστοιχα. Και πάλι η χρήση 24 threads μειώνει το speedup, το οποίο αυτή τη φορά πέφτει σχεδόν στο 2.0.

4.4 Παρουσίαση συνέπειας αποτελεσμάτων και κατηγοριοποίησης

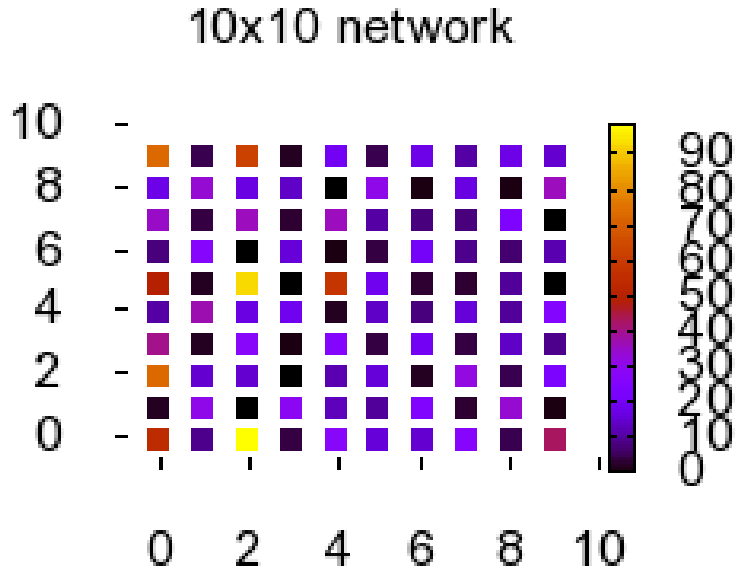
4.4.1 Δίκτυο διαστάσεων 10x10

4.4.1.1 Αποτελέσματα από εκτέλεση με όλους τους subjects

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	10	76	108
2	9	71	79
3	10	64	98
4	10	59	77
5	6	48	73
6	5	47	54
7	6	44	62
8	9	44	59
9	6	42	59

Πίνακας 4.1 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 10x10 με cases και controls και 60% ομοιότητα μεταξύ των κατηγοριών

Η συνέπεια των αποτελεσμάτων για αυτό το δίκτυο χρησιμοποιώντας το αρχείο που περιέχει cases και controls, είναι πολύ μικρή. Για να βρεθούν οι ομάδες που φαίνονται στον πίνακα 4.1, χρειάστηκε να ελέγξω για ομάδες κατηγοριών που βρήκε η εφαρμογή στις δέκα εκτελέσεις με ομοιότητα 60%. Αυτό έχει ως αποτέλεσμα μια ανομοιομορφία στις κατηγορίες που ομαδοποιήθηκαν, έχοντας όμως ένα ποσοστό με κοινούς subjects. Όπως φαίνεται στο πίνακα 4.1, υπάρχουν κατηγορίες με μεγάλο αριθμό subjects όπως οι ομάδες 1,2,3 και 4 από τις οποίες θα μπορούσαν να εξαχθούν κάποια αποτελέσματα ελέγχοντας τα SNPs μεταξύ των κοινών subjects της ομάδας που μας ενδιαφέρει. Υπάρχουν και άλλες ομάδες κατηγοριών αλλά στον πιο πάνω πίνακα παρουσιάζονται αυτές που πιστεύεται ότι είναι πιο σημαντικές.



Σχήμα 4.8 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 10x10 με cases και controls

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας	Αριθμός cases κατηγορίας	Αριθμός controls κατηγορίας	Επικρατέστερος φαινότυπος κατηγορίας
X	Y				
2	0	98	76	22	case
2	5	93	64	29	case
0	9	73	16	57	control
0	2	73	24	49	control
2	9	62	17	45	control
4	5	58	40	18	case
0	0	54	30	24	case
0	5	49	24	25	control
9	0	43	37	6	case
0	3	40	11	29	control
1	4	37	24	13	case

Πίνακας 4.2 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 10x10 με cases και controls

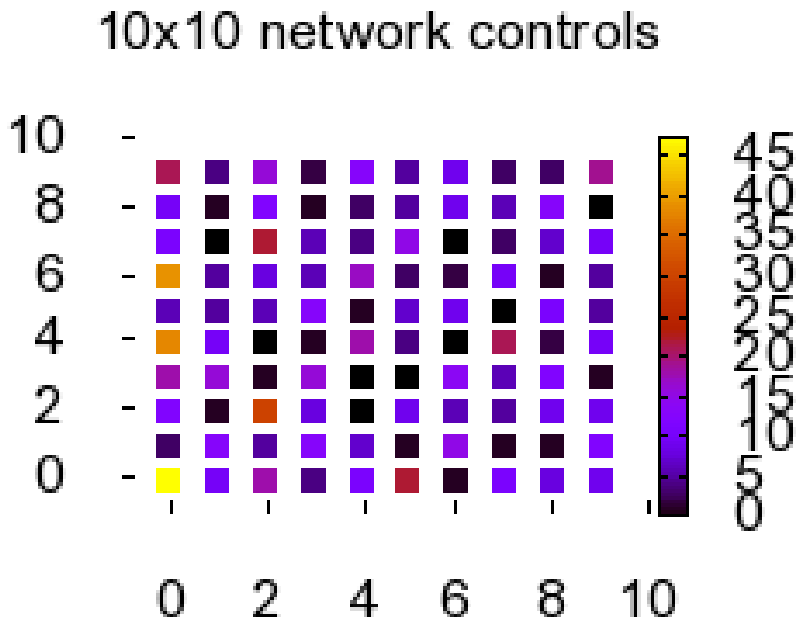
Στο σχήμα 4.8, παρουσιάζεται η κατηγοριοποίηση που έγινε σε μια από τις 10 εκτελέσεις της εφαρμογής στο δίκτυο αυτό με cases και controls και στον πίνακα 4.2 παρουσιάζονται οι σημαντικότερες κατηγορίες που προέκυψαν από την εκτέλεση. Στη συγκεκριμένη εκτέλεση έγιναν 93 κατηγορίες. Η μεγαλύτερη κατηγοριοποίηση που έγινε ήταν στο νευρώνα 2.0 με 98 subjects εκ των οποίων οι 76 είναι cases και οι 22 controls. Η αμέσως επόμενη σε μέγεθος κατηγορία είναι στο νευρώνα 2.5 η οποία έχει 93 subjects, όπου οι 64 είναι cases και οι υπόλοιποι 29 controls. Υπάρχουν δύο νευρώνες με 73 subjects, ο 9.0 και ο 0.2 με 16 cases και 57 controls ο πρώτος και 24 cases και 49 controls ο δεύτερος.

4.4.1.2 Αποτελέσματα από εκτέλεση με controls

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	9	37	50
2	9	28	43
3	8	28	42
4	10	24	41
5	7	18	23
6	5	18	24
7	10	17	21
8	8	16	21
9	10	16	21
10	9	16	29

Πίνακας 4.3 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 10x10 με controls και 60% ομοιότητα μεταξύ των κατηγοριών

Και σε αυτή την εκτέλεση χρειάστηκε να μειώσουμε την ομοιότητα μεταξύ των κατηγοριών κάθε εκτέλεσης στο 60% για να έχουμε κάποιες κοινές κατηγορίες. Σε αυτά όμως τα δεδομένα η συνέπεια των κατηγοριών είναι καλύτερη από τις εκτελέσεις με όλους τους subjects αφού βλέπουμε να εμφανίζονται σε περισσότερες εκτελέσεις οι συγκεκριμένες κατηγορίες του πίνακα 4.3.



Σχήμα 4.9 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 10x10 με controls

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας
X	Y	
0	0	47
0	6	39
0	4	38
2	2	30
2	7	22
5	0	22
0	9	21
7	4	21
9	9	19
0	3	18

Πίνακας 4.4 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 10x10 με controls

Από τον πίνακα 4.4 και το σχήμα 4.9 μπορούμε να δούμε τις σημαντικότερες κατηγοριοποιήσεις μιας από τις δέκα εκτελέσεις που έγιναν για αυτά τα δεδομένα. Εδώ η

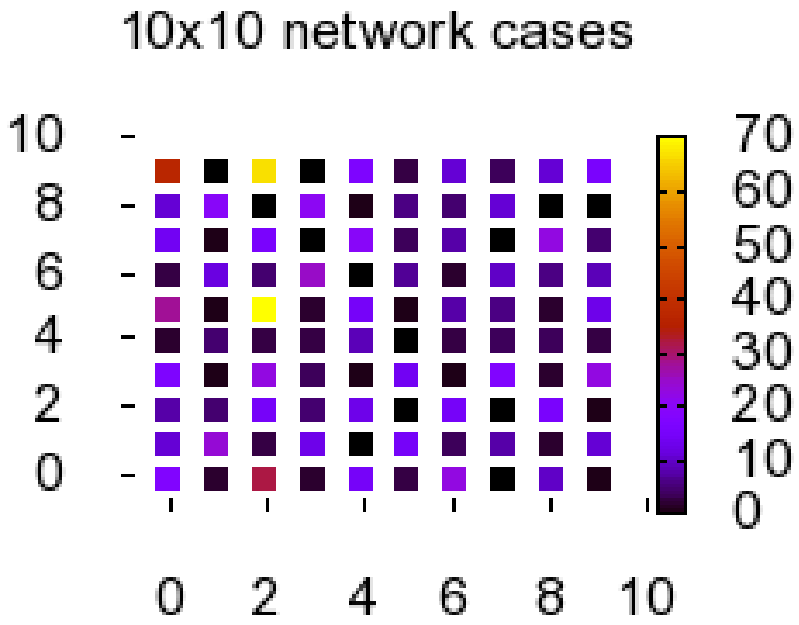
μεγαλύτερη κατηγορία περιέχει 47 subjects, ενώ οι αμέσως επόμενες έχουν 39 και 38 subjects αντίστοιχα. Εδώ έχουμε δύο γειτονικούς νευρώνες, τον 0.4 και τον 0.3 με 38 και 18 subjects αντίστοιχα και υπάρχει μεγάλη πιθανότητα να έχουν αρκετές ομοιότητες στα SNPs τους οι subjects αυτών των νευρώνων. Συνολικά σε αυτή την εκτέλεση έχουν δημιουργηθεί 91 κατηγορίες.

4.4.1.2 Αποτελέσματα από εκτέλεση με cases

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	10	50	76
2	8	47	82
3	10	31	37
4	10	24	31
5	9	22	27
6	8	20	33
7	10	20	29
8	6	20	30
9	10	18	25
10	10	17	23

Πίνακας 4.5 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 10x10 με cases και 60% ομοιότητα μεταξύ των κατηγοριών

Οι κατηγορίες και με αυτά τα δεδομένα έχουν περισσότερη συνέπεια σε σχέση με την εκτέλεση με όλους τους subjects, αλλά και πάλι για να προκύψουν χρειάστηκε να ελέγξουμε για 60% ομοιότητα μεταξύ των κατηγοριών. Στις περισσότερες ομάδες, έχει προκύψει και στις 10 εκτελέσεις η συγκεκριμένη κατηγορία για κάθε ομάδα (πίνακας 4.5) κάτι που είναι ενθαρρυντικό.



Σχήμα 4.10 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 10x10 με cases

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας
X	Y	
2	5	70
2	9	67
0	9	37
2	0	32
0	5	28
3	6	25
1	1	24
2	3	23
6	0	23
8	7	23
9	3	23

Πίνακας 4.6 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 10x10 με cases

Στη εκτέλεση που παρουσιάζονται οι κατηγοριοποιήσεις της στο σχήμα 4.10 και στον πίνακα 4.6 δημιουργήθηκαν συνολικά 87 κατηγοριοποιήσεις. Από αυτές οι κατηγοριοποιήσεις με τον μεγαλύτερο αριθμό subjects παρουσιάζονται στον πίνακα 4.6. Εδώ δεν βλέπουμε κάποια από τις σημαντικές κατηγοριοποιήσεις να γειτονεύει άμεσα με άλλη.

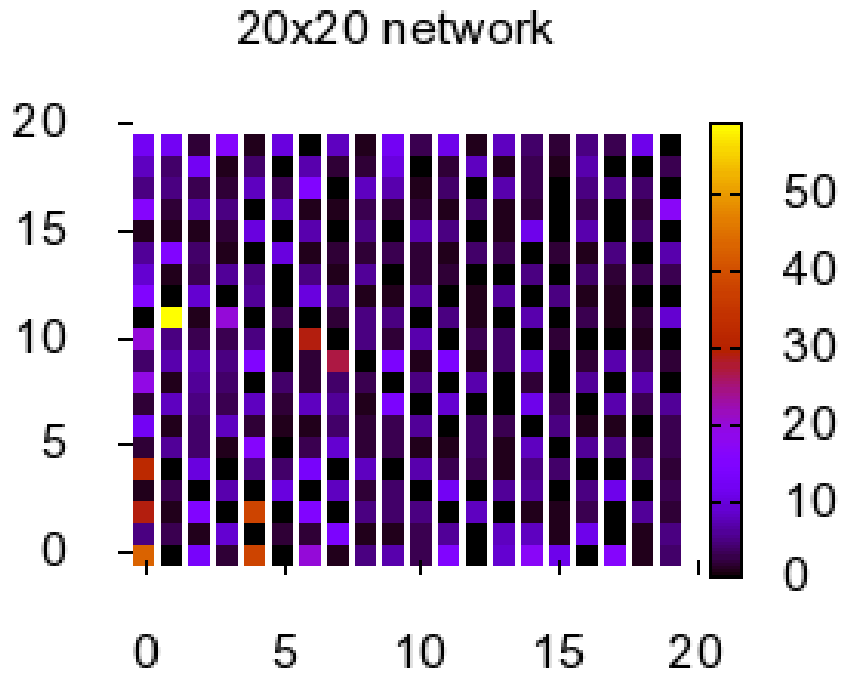
4.4.2 Δίκτυο διαστάσεων 20x20

4.4.2.1 Αποτελέσματα από εκτέλεση με όλους τους subjects

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	10	53	65
2	10	34	43
3	10	31	47
4	10	29	49
5	10	26	32
7	7	24	36
8	10	22	31
9	9	16	20
10	10	16	26

Πίνακας 4.7 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 20x20 με cases και controls και 70% ομοιότητα μεταξύ των κατηγοριών

Μεγαλώνοντας το δίκτυο με αυτά τα δεδομένα είχαμε πιο σταθερή κατηγοριοποίηση όπως βλέπουμε και στον πίνακα 4.7. Αυτή τη φορά είχαμε σχεδόν σε όλες τις εκτελέσεις την εμφάνιση των συγκεκριμένων κατηγοριών, και αυτή τη φορά ο έλεγχος έγινε με 70% ομοιότητα. Η αύξηση των διαστάσεων έχει μειώσει και τον αριθμό των subjects σε κάθε κατηγορία, αφού ενδεικτικά η πρώτη ομάδα σε αυτό το δίκτυο σε σχέση με το 10x10 έχει σχεδόν το μισό αριθμό subjects.



Σχήμα 4.11 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 20x20 με cases και controls

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας	Αριθμός cases κατηγορίας	Αριθμός controls κατηγορίας	Επικρατέστερος φαινότυπος κατηγορίας
X	Y				
1	11	59	48	11	case
0	0	43	37	6	case
4	0	38	26	12	case
4	2	38	30	8	case
0	4	32	24	8	case
0	2	29	23	6	case
6	10	29	9	20	control
7	9	27	10	17	control
0	10	20	15	5	case
3	11	20	14	6	case

Πίνακας 4.8 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 20x20 με cases και controls

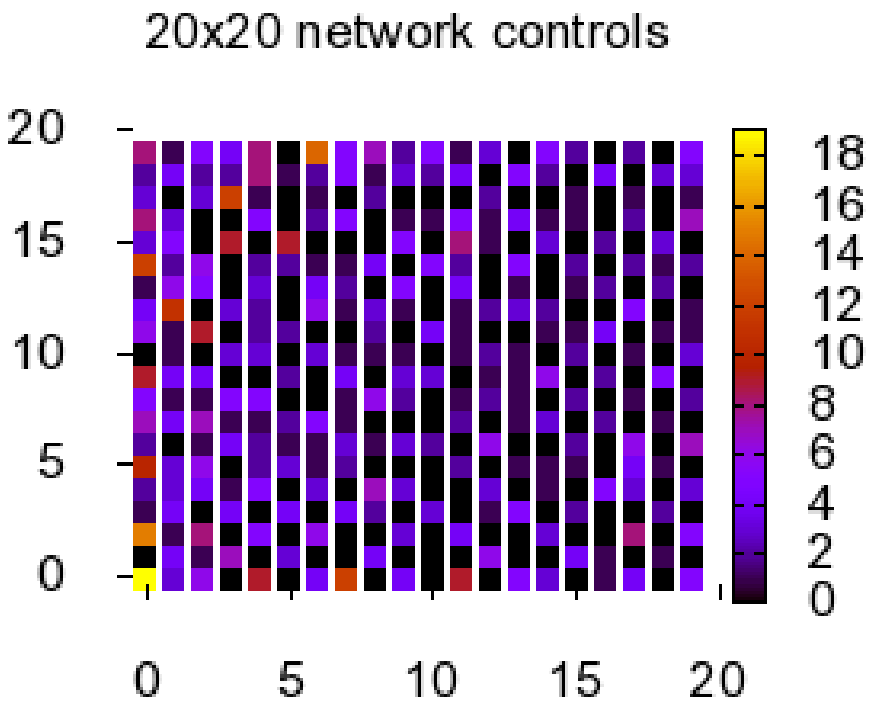
Από τον πίνακα 4.8 και το σχήμα 4.11, βλέπουμε ότι οι κατηγορίες με μεγαλύτερη σημασία βρίσκονται κοντά. Συγκεκριμένα έχουμε κοντά τις κατηγοριοποιήσεις των νευρώνων 0.0, 0.2 και 0.4. Επίσης κοντά βρίσκονται και αυτές των νευρώνων 4.0 και 4.2 ενώ γειτονικά είναι οι κατηγορίες των 6.10 και 7.9. Ο νευρώνας που αντιπροσωπεύει την μεγαλύτερη κατηγορία είναι απομακρυσμένος από τους υπόλοιπους που αναφέρθηκαν έχοντας 59 subjects. Επιπλέον στις σημαντικότερες κατηγοριοποιήσεις αυτού του δικτύου, οι cases είναι περισσότεροι από τους controls, έχοντας περισσότερες κατηγοριοποιήσεις έτσι που να χαρακτηρίζονται ως case. Τέλος, εκτός από τη συνέπεια των αποτελεσμάτων, με την αύξηση του μεγέθους του δικτύου, αυξήθηκε και ο συνολικός αριθμός κατηγοριών που δημιουργήθηκαν σε 307, έχοντας αρκετές με 1 ή 2 subjects μόνο.

4.4.2.2 Αποτελέσματα από εκτέλεση με controls

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	10	14	19
2	9	12	19
3	10	11	19
4	9	10	13
5	10	10	15
6	10	10	16
7	10	9	13
8	6	8	11
9	8	8	14
10	10	8	11
11	7	8	9
12	8	8	10
13	10	7	10
14	10	7	10

Πίνακας 4.9 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 20x20 με controls και 60% ομοιότητα μεταξύ των κατηγοριών

Στα συγκεκριμένα δεδομένα χρειάστηκε να μειώσω την ομοιότητα μεταξύ των κατηγοριών στο 60% ώστε να βρεθούν οι ομάδες του πίνακα 4.9. Επίσης οι κατηγορίες έχουν μικρότερο αριθμό subjects αλλά η συνέπεια των αποτελεσμάτων είναι καλύτερη από την αντίστοιχη του δικτύου με μέγεθος 10x10. Η μεγαλύτερη κατηγορία κυμαίνεται από 14 μέχρι 19 subjects σε αντίθεση με τους 37 με 50 subjects του μικρότερου δικτύου και έτσι δεν προκύπτει κάποια μεγάλη κατηγορία.



Σχήμα 4.12 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 20x20 με controls

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας
X	Y	
0	0	19
0	2	15
6	19	14
0	14	12
3	17	12
7	0	12
1	12	11
0	5	10
0	9	9
2	11	9
3	15	9
4	0	9
5	15	9

11	0	9

Πίνακας 4.10 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 20x20 με controls

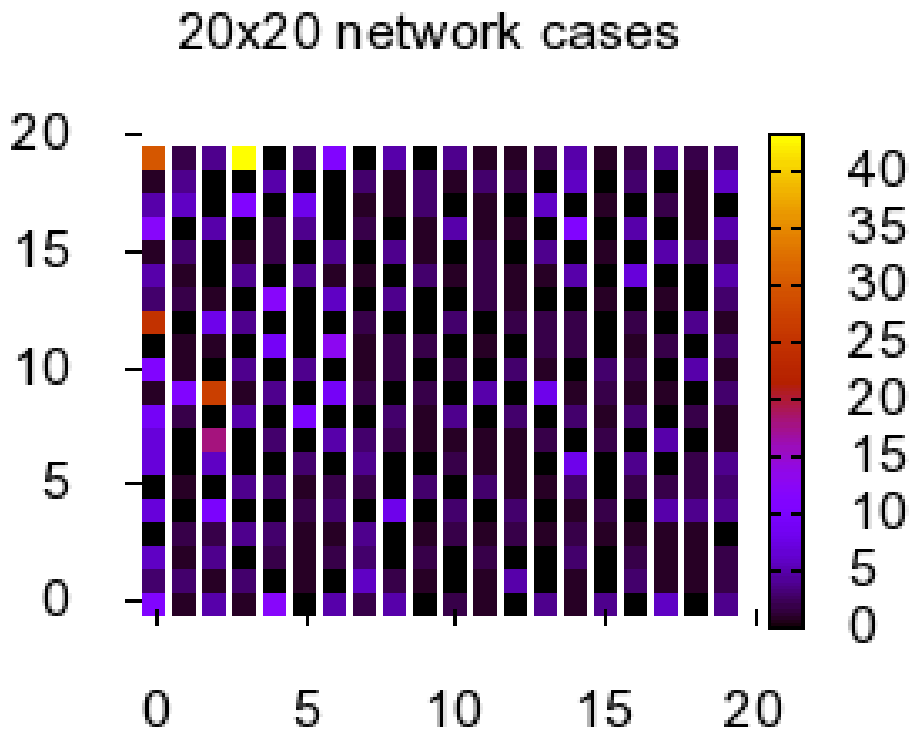
Στον πίνακα 4.10 παρουσιάζεται ως μεγαλύτερη κατηγορία αυτή στο νευρώνα 0.0 με 19 subjects. Η αμέσως επόμενη μεγαλύτερη κατηγορία, είναι αυτή του νευρώνα 0.2 με 15 νευρώνες, η οποία όπως βλέπουμε στο σχήμα 4.12 έχει ένα νευρώνα απόσταση από τη μεγαλύτερη κατηγορία. Γειτονικές σημαντικές κατηγορίες, είναι και αυτές που βρίσκονται στο νευρώνα 1.12 και 2.10 με 11 και 9 subjects αντίστοιχα. Συνολικά δημιουργήθηκαν 256 κατηγορίες έχοντας και πάλι μεγάλο αριθμό κατηγοριών με 1 ή 2 subjects.

4.4.2.3 Αποτελέσματα από εκτέλεση με cases

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	10	42	44
2	10	27	37
3	7	26	28
4	10	20	25
5	10	11	12
6	10	11	14
7	9	10	13
8	9	9	12
9	8	9	12
10	10	9	10
11	10	9	11

Πίνακας 4.11 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 20x20 με cases και 70% ομοιότητα μεταξύ των κατηγοριών

Και σε αυτά τα δεδομένα, με την αύξηση του δικτύου το μέγεθος των κατηγοριών μειώθηκε, δίνοντάς μας περισσότερες κατηγορίες. Το πιο μικρό μέγεθος κατηγοριών μας έδωσε και πιο σταθερά αποτελέσματα σε σύγκριση με τις αντίστοιχες εκτελέσεις στο μικρότερο δίκτυο. Για να έχουμε την παρουσίαση των συγκεκριμένων κατηγοριών του πίνακα 4.11 σε μεγάλο ποσοστό εκτελέσεων χρειάστηκε να συγκρίνω για 70% ομοιότητα μεταξύ τους, σε σύγκριση με το 60% του δικτύου με διαστάσεις 10x10.



Σχήμα 4.13 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 20x20 με cases

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας
X	Y	
3	19	43
0	19	30
2	9	27
0	12	25
2	7	18

Πίνακας 4.12 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 20x20 με cases

Στη συγκεκριμένη εκτέλεση που παρουσιάζεται στο σχήμα 4.13 και στον πίνακα 4.12, υπάρχουν μόνο 5 κατηγοριοποιήσεις που να είναι ευδιάκριτες. Η μεγαλύτερη αποτελείται από 43 subjects, ενώ η μικρότερη από αυτές από 18. Γενικά οι κατηγορίες αυτές είναι πιο απομακρυσμένες η μια από την άλλη με μόνη εξαίρεση αυτή που βρίσκεται στο νευρώνα 2.7 και στο νευρώνα 2.9 με 18 και 27 subjects αντίστοιχα, και ενδεχομένως να έχουν κάποια κοινά χαρακτηριστικά στα SNPs τους. Συνολικά δημιουργήθηκαν 271 κατηγορίες με αποτέλεσμα πολλές από αυτές να περιέχουν πολύ μικρό αριθμό subjects.

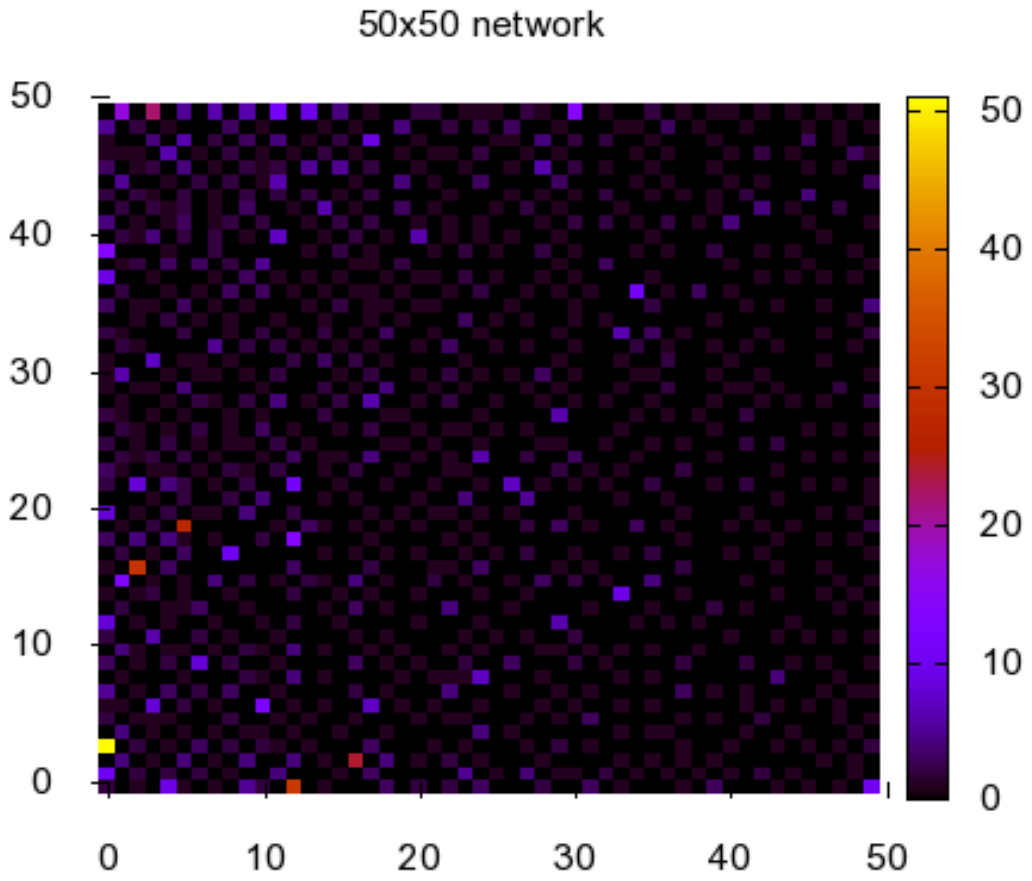
4.4.3 Δίκτυο διαστάσεων 50x50

4.4.3.1 Αποτελέσματα από εκτέλεση με όλους τους subjects

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	10	51	51
2	10	31	31
3	10	30	30
4	10	26	27
5	10	24	24
6	7	20	20

Πίνακας 4.13 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 50x50 με cases και controls και 95% ομοιότητα μεταξύ των κατηγοριών

Σε τόσο μεγάλο δίκτυο, πλέον οι κατηγορίες γίνονται αρκετά ανεξάρτητες, και μειώνεται σημαντικά ο αριθμός κατηγοριών που αποτελούνται από πολλούς subjects. Από την άλλη, τα αποτελέσματά μας πλέον είναι συνεπή (πίνακας 4.13), αφού το ποσοστό ομοιότητας πλέον είναι στο 95% και οι συγκεκριμένες κατηγορίες παρουσιάστηκαν σε όλες τις εκτελέσεις. Μόνη εξαίρεση είναι η ομάδα 7 που παρουσιάστηκε στις 7 από τις 10 εκτελέσεις με 95% ομοιότητα.



Σχήμα 4.14 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 50x50 με cases και controls

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας	Αριθμός cases κατηγορίας	Αριθμός controls κατηγορίας	Επικρατέστερος φαινότυπος κατηγορίας
X	Y				
0	3	51	42	9	case
12	0	31	27	4	case
2	16	30	23	7	case
5	19	27	17	10	case
16	2	24	20	4	case
3	49	22	8	14	control
1	49	17	6	11	control

Πίνακας 4.14 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 50x50 με cases και controls

Η εκτέλεση που παρουσιάζεται στο σχήμα 4.14 και στον πίνακα 4.14, δημιούργησε συνολικά 914 κατηγορίες. Έχουμε σχεδόν τον τριπλάσιο αριθμό κατηγοριών και μικρότερο αριθμό subjects στις κατηγορίες. Οι περισσότερες κατηγορίες και πάλι έχουν ως επικρατέστερο φαινότυπο τους cases, ενώ controls είναι οι δύο τελευταίες κατηγορίες του πίνακα 4.14. Η αναλογία της κατηγορία με τους 51 subjects των cases και controls έχει p-value = $4.36x10^{-6}$ και δίνει μεγάλη στατιστική αξία στην κατηγορία αυτή όσο αφορά τη βιολογική της σημασία. Οι κατηγορίες βρίσκονται σε μεγάλες αποστάσεις μεταξύ τους, και συγκεκριμένα η κατηγορία με τους 51 subjects, είναι σε μεγάλη απόσταση από όλες τις άλλες που παρουσιάζονται στον πίνακα 4.14.

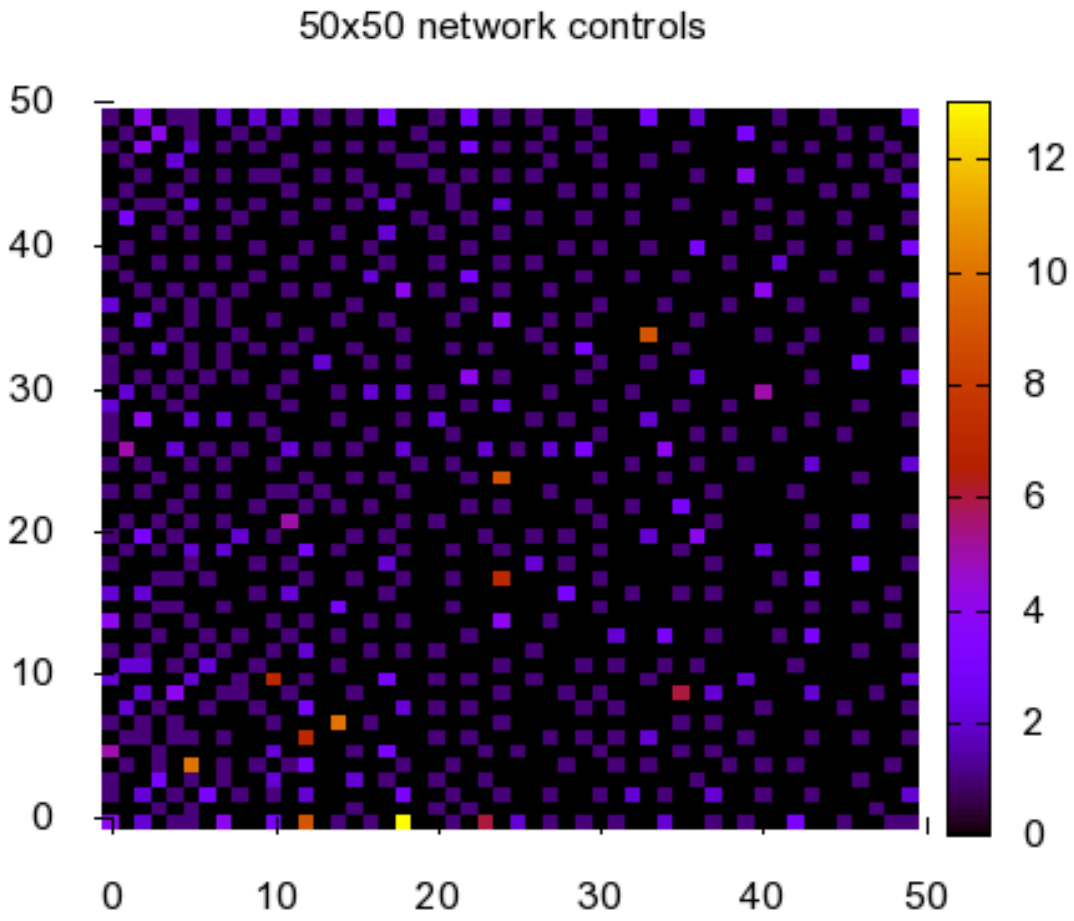
4.4.3.2 Αποτελέσματα από εκτέλεση με controls

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	7	12	12
2	9	10	10
3	10	10	10
4	9	9	9
5	8	9	9
6	9	8	8
7	10	7	7
8	10	7	7
9	10	7	7

Πίνακας 4.15 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 50x50 με controls και 90% ομοιότητα μεταξύ των κατηγοριών

Ο αριθμός των subjects στις κατηγορίες σε σχέση με την εκτέλεση στο 20x20 δίκτυο, δεν έχουν μεγάλη διαφορά. Η διαφορά όμως είναι στο ότι σε αντίθεση με το 60% ομοιότητας που ελέγχαμε μεταξύ των κατηγοριών που προέκυψαν από τις 10 εκτελέσεις για το προηγούμενο δίκτυο, σε αυτό ελέγχουμε για 90% και προκύπτει ότι υπάρχει συνέπεια μεταξύ των κατηγοριοποιήσεων με

πολύ μικρή διαφορά μεταξύ τους (πίνακας 4.15). Συγκεκριμένα με αυτό τον αριθμό subjects στις κατηγορίες και έλεγχο για 90% ομοιότητα, οι κατηγορίες που ανήκουν στην ομάδα 1, επιτρέπεται να έχουν μόλις ένα subject διαφορά μεταξύ τους, ενώ στις υπόλοιπες ομάδες δεν επιτρέπεται να υπάρχει διαφορά στις κατηγορίες που τις αποτελούν.



Σχήμα 4.15 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 50x50 με controls

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας
X	Y	
18	0	13
5	4	10
14	7	10
12	0	9
24	24	9
33	34	9
10	10	7
12	6	7
24	17	7

Πίνακας 4.16 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 50x50 με controls

Από την εκτέλεση που παρουσιάζεται στο σχήμα 4.15, φαίνεται ότι οι κατηγορίες έχουν μεγαλύτερες αποστάσεις μεταξύ τους και ότι παρουσιάζονται αρκετοί νευρώνες στο δίκτυο χωρίς

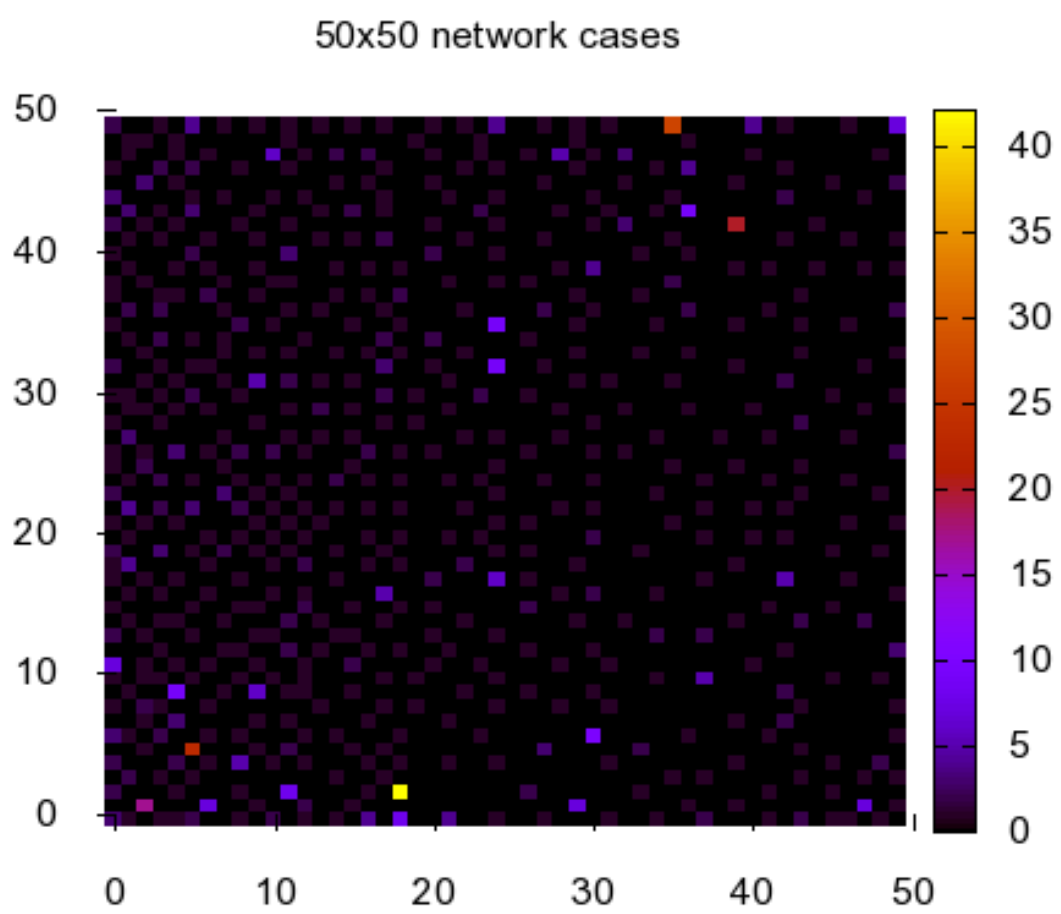
κατηγορίες. Στον πίνακα 4.16 παρουσιάζονται οι σημαντικότερες κατηγορίες που προέκυψαν από αυτή την εκτέλεση. Οι κατηγορίες έχουν μικρό αριθμό subjects, αφού η μεγαλύτερη κατηγορία περιέχει 13, ενώ ακολουθούν δύο με 10 subjects. Συνολικά δημιουργήθηκαν 599 κατηγορίες και το γεγονός ότι οι σημαντικότερες είχαν το πολύ 13 subjects, είναι φανερό ότι μεγάλος αριθμός κατηγοριών περιέχει πολύ μικρό αριθμό subjects.

4.4.3.3 Αποτελέσματα από εκτέλεση με cases

Αριθμός ομάδας κατηγοριών	Πόσες φορές βρέθηκε η κατηγορία	Ελάχιστος αριθμός subjects	Μέγιστος αριθμός subjects
1	10	42	42
2	10	27	27
3	10	23	23
4	10	20	20
5	10	17	17
6	10	10	10
7	10	9	9

Πίνακας 4.17 Συνέπεια αποτελεσμάτων σε δίκτυο διαστάσεων 50x50 με και 95% ομοιότητα μεταξύ των κατηγοριών

Χρησιμοποιώντας μόνο τους cases ως είσοδο στο SOM με αυτό το δίκτυο, η κατηγορίες που προέκυψαν εμφανίστηκαν σε όλες τις δοκιμαστικές εκτελέσεις που έγιναν για τον έλεγχο της συνέπειας των αποτελεσμάτων. Επιπλέον η ομοιότητα μεταξύ των κατηγοριών που ελέγχθηκαν ήταν στο 95%, δίνοντάς μας τα καλύτερα αποτελέσματα ως προς το αν οι κατηγορίες που προκύπτουν είναι τυχαίες.



Σχήμα 4.16 Αναπαράσταση ενεργοποίησης νευρώνων σε δίκτυο διαστάσεων 50x50 με cases

Συντεταγμένες Νευρώνα		Αριθμός subjects κατηγορίας
X	Y	
18	2	42
35	49	27
5	5	23
39	42	20
2	1	17
30	6	10

Πίνακας 4.18 Σημαντικότερες κατηγορίες σε δίκτυο διαστάσεων 50x50 με cases

Όπως φαίνεται στον πίνακα 4.18 και στο σχήμα 4.16, οι σημαντικές κατηγορίες είναι απομακρυσμένες η μία από την άλλη, ενώ βλέπουμε την κατηγορία που βρίσκεται στο νευρώνα 39.42 να μην έχει γύρω της καμία άλλα κατηγορία ακόμη και με μικρό αριθμό subjects. Η σημαντικές αυτές κατηγορίες βρίσκονται όλες και στο σενάριο με τη χρήση της βάσης με cases και controls. Και με αυτά τα δεδομένα ως είσοδο στο SOM, υπάρχουν αρκετοί νευρώνες που δεν περιέχουν κάποια κατηγορία. Συνολικά δημιουργήθηκαν 584 κατηγορίες στην εκτέλεση που παρουσιάζεται εδώ, ενώ το δίκτυο περιέχει 2500 νευρώνες. Δηλαδή περίπου τα τέσσερα πέμπτα των νευρώνων δεν περιέχουν καμία κατηγορία, ενώ μεγάλος αριθμός κατηγοριών περιέχει πολύ μικρό αριθμό subjects.

3.4.3 Σύγκριση γραμμικής και δυαδικής κωδικοποίησης

Στο δίκτυο με διαστάσεις 10x10, η γραμμική κωδικοποίηση είχε 60% ομοιότητα στους subjects κάθε κατηγορίας, ενώ η δυαδική κωδικοποίηση είχε 70% ομοιότητα. Επίσης στο 20x20 δίκτυο, η γραμμική κωδικοποίηση είχε 70% ομοιότητα, σε αντίθεση με τη δυαδική κωδικοποίηση που οι κατηγορίες της είχαν 80% ομοιότητα. Τέλος στο μεγαλύτερο δίκτυο, και οι δύο κωδικοποιήσεις είχαν 95% ομοιότητα στους subjects των κατηγοριών τους ανεξάρτητα.

Στον έλεγχο που έγινε για την ομοιότητα μεταξύ των subjects που προέκυψαν σε κάθε κατηγορία και από τις δυο κωδικοποιήσεις, προέκυψε 60% ομοιότητα στο 10x10 δίκτυο, 75% ομοιότητα στο 20x20 δίκτυο και 95% ομοιότητα στο 50x50 δίκτυο. Ελέγχοντας τους subjects που ήταν κοινοί στις 10 εκτελέσεις κάθε κωδικοποίησης ξεχωριστά και μετά μεταξύ τους, προκύπτει ότι η διαφορά στους subjects είναι πολύ κοντά στο 5%.

Κεφάλαιο 5

Συζήτηση

5.1 Βελτιστοποιήσεις

Από τα αποτελέσματα της χρήσης των βελτιστοποιήσεων φαίνεται ότι μπορούν όντως να βοηθήσουν στη μείωση του χρόνου εκτέλεσης της εφαρμογής. Η βελτιστοποίηση της αντικατάστασης των δύο συγχρονισμών με ένα που να περιλαμβάνει μέσα του τις δύο διεργασίες που θέλαμε να συγχρονίσουμε, είχε σταθερή απόδοση όσο αφορά τη βελτίωση της ταχύτητας εκτέλεσης της εφαρμογής. Αυτό οφείλεται στο γεγονός ότι αναγκαστικά όλοι οι επεξεργαστές θα περάσουν τον ίδιο αριθμό φορών από το συγκεκριμένο συγχρονισμό, έτσι το όφελος του δεν επηρεάζεται από άλλους παράγοντες της παραλληλοποίησης όπως το πόσους και ποιους νευρώνες θα αναλάβει ο κάθε επεξεργαστής ή τον αριθμό των επεξεργαστών που χρησιμοποιούνται. Αντίθετα, η βελτιστοποίηση με τον προϋπολογισμό των αποστάσεων δεν έχει σταθερή απόδοση, αλλά έχει αυξομειώσεις. Αυτό προκύπτει από τον τρόπο που διαμοιράζονται οι νευρώνες στους επεξεργαστές. Κάθε επεξεργαστής αναλαμβάνει τον ίδιο αριθμό νευρώνων με τους υπόλοιπους, και συγκεκριμένα αν είχαμε ένα δίκτυο 10x10 και δύο επεξεργαστές, τότε κάθε επεξεργαστής θα αναλάμβανε από 50 νευρώνες. Οι νευρώνες αυτοί θα ήταν συνεχόμενοι στο δίκτυο, δηλ. ο πρώτος θα αναλάμβανε τις πρώτες 5 σειρές και ο δεύτερος τις υπόλοιπες 5. Αν για παράδειγμα ο BMU και το μεγαλύτερο μέρος της γειτονιάς του βρίσκεται στις πρώτες 5 σειρές, τότε ο πρώτος επεξεργαστής θα έχει να προσαρμόσει τα βάρη περισσότερων νευρώνων. Επομένως ο ένας επεξεργαστής θα τελειώσει πιο γρήγορα και θα περιμένει τον άλλο και θα χάσεις μέρος από το όφελος του παραλληλισμού. Αυτό στην περίπτωση της βελτιστοποίησης με τον προϋπολογισμό των αποστάσεων μπορεί να επηρεάσει στο συνολικό χρόνο βελτιστοποίησης,

γιατί υπάρχουν περιπτώσεις που μπορεί να τύχει ο ένας επεξεργαστής να περιμένει τον άλλο να τελειώσει και παρά το ότι κάνει πιο γρήγορα τις αλλαγές των βαρών με τις προϋπολογισμένες αποστάσεις, χάνει χρόνο στο να περιμένει τον άλλο επεξεργαστή να τελειώσει τις προσαρμογές στα βάρη των δικών του νευρώνων. Το πρόβλημα αυτό λοιπόν έχει σχέση με τον ισοζυγισμό της εργασίας που έχει να κάνει κάθε επεξεργαστής (load balancing). Αυτό εξηγεί και το λόγο που έχουμε μεγαλύτερη βελτιστοποίηση από την αναμενόμενη στο δίκτυο 100x100 στο σχήμα 4.4, γιατί ενώ στην εκτέλεση με το 100x100 δίκτυο χρησιμοποιώντας μόνο τη βελτιστοποίηση του προϋπολογισμού των αποστάσεων έτυχε να μην υπάρχει καλό load balancing, στο αντίστοιχο δίκτυο με την εφαρμογή και με τις δύο βελτιστοποιήσεις έτυχε να υπάρχει καλύτερο.

5.2 Σύγκριση χρόνων σειριακού και παράλληλου αλγόριθμου

Γενικά στη σύγκριση των χρόνων, η χρήση του παράλληλου SOM με ένα thread κοστίζει περισσότερο απ' ότι τη χρήση του καθαρά σειριακού SOM. Αυτό οφείλεται στο ότι στον παράλληλο SOM υπάρχει η φάση του συγχρονισμού, στην οποία παρότι σε αυτή την περίπτωση δεν χρειάζεται να περιμένει κάποιο άλλο thread, χρειάζεται να γίνουν οι έλεγχοι στις μεταβλητές του αμοιβαίου αποκλεισμού. Επίσης για να είναι εφικτή η διαμοίραση του χώρου, ο δισδιάστατος πίνακας μετατρέπεται σε ένα μονοδιάστατο πίνακα με NxN στοιχεία, όπου το κάθε ένα έχει φυλαγμένες τις πραγματικές του συντεταγμένες στο δισδιάστατο πλέγμα. Η πρόσβαση κάθε φορά για την εύρεση των συντεταγμένων κοστίζει στο χρόνο εκτέλεσης. Επιπλέον, η χρήση 24 threads μας δίνει χειρότερα αποτελέσματα απ' ότι αν χρησιμοποιούσαμε 12, που είναι και ο αριθμός των επεξεργαστών που υπάρχουν στη μηχανή που τρέχαμε τον παράλληλο SOM. Αυτό οφείλεται στο γεγονός ότι χρειάζεται να κάνει εναλλαγή των threads στους επεξεργαστές για να τρέξουν όλα τα threads που ορίσαμε και αυτό κοστίζει στη συνολική εκτέλεση της εφαρμογής.

Στο 20x20 δίκτυο είχαμε μια οριακή αύξηση της ταχύτητας του αλγορίθμου στα 2 και 4 threads, ενώ στη συνέχεια με την αύξηση του αριθμού των threads μειωνόταν και το speedup. Ο λόγος που δεν έχουμε καλά αποτελέσματα σε αυτό το μέγεθος δικτύου είναι το ότι οι επεξεργαστές που είχαμε ήταν αρκετά γρήγοροι, με αποτέλεσμα οι νευρώνες που είχαν να ελέγξουν να είναι λίγοι για την απόδοσή τους. Δηλαδή, οι επεξεργαστές τελείωναν πολύ γρήγορα τους υπολογισμούς και τις αλλαγές τιμών που είχαν στους νευρώνες τους, και το κόστος του συγχρονισμού ήταν περισσότερο από το όφελος που είχαν από τον παραλληλισμό.

Στα επόμενα δύο σενάρια που το μέγεθος του δισδιάστατου πλέγματος ήταν μεγαλύτερο, το speedup ήταν καλύτερο και με την αύξηση του αριθμού των threads αυξανόταν και αυτό. Η αύξηση του speedup δεν ήταν γραμμική και αυτό οφείλεται στο πρόβλημα του load balancing που αναφέρθηκε προηγουμένως, αφού δεν έγινε πλήρης εκμετάλλευση του παραλληλισμού.

5.3 Παρουσίαση συνέπειας αποτελεσμάτων και κατηγοριοποίησης

5.3.1 Δίκτυο διαστάσεων 10x10

Στο δίκτυο αυτό τα αποτελέσματα δεν ήταν τόσο συνεπή, αφού για να προκύψουν κάποιες κοινές κατηγορίες στις δέκα εκτελέσεις αυτού του δικτύου, χρειάστηκε να μειωθεί η επιθυμητή ομοιότητα μεταξύ των κατηγοριών των εκτελέσεων στο 60%. Το δίκτυο αυτό μας έδωσε τις κατηγορίες με τους περισσότερους subjects και στις 3 βάσεις δεδομένων (cases, controls, cases και controls). Το ότι ήταν μικρό το μέγεθος του δικτύου, ανάγκασε την εισαγωγή στις κατηγορίες που δημιουργήθηκαν και subjects που δεν είχαν πολύ μεγάλη ομοιότητα με τους υπόλοιπους βασικούς subjects της κατηγορίας. Αυτός είναι και ο λόγος που δεν είχαμε μεγάλη συνέπεια στις τελικές κατηγοριοποιήσεις μεταξύ των 10 εκτελέσεων που έγιναν για κάθε βάση δεδομένων. Συγκεκριμένα, υπήρχαν subjects που μπορεί να είχαν ένα κομμάτι κοινό με τη μια κατηγορία και ένα άλλο κομμάτι κοινό με κάποια άλλη, με αποτέλεσμα σε κάποιες εκτελέσεις να καταλήγουν στη μια κατηγορία και σε άλλες να καταλήγουν στην άλλη. Επίσης υπήρχε γενικά μια ισότητα στον αριθμό των σημαντικών κατηγοριών που είχαν περισσότερους cases ή controls στις εκτελέσεις στη βάση δεδομένων που περιείχε όλους τους subjects. Συγκεκριμένα υπήρχαν 6 κατηγορίες που ήταν επικρατέστερος ο φαινότυπος των cases και 5 που ήταν επικρατέστερος ο φαινότυπος των controls.

5.3.2 Δίκτυο διαστάσεων 20x20

Το γεγονός ότι δεν υπήρχαν σημαντικές κατηγορίες όσο αφορά τους controls, είναι σημαντικό γιατί βιολογικά δεν θα έπρεπε να βρεθούν μεγάλες κατηγορίες σε αυτούς, κάτι που μας δείχνει ότι ο αλγόριθμος κατηγοριοποίησε σωστά τα δεδομένα. Σε αυτό το δίκτυο, είχαμε μια βελτίωση στη συνέπεια των αποτελεσμάτων, αφού οι σημαντικές κατηγορίες, υπήρχαν σχεδόν σε όλες τις εκτελέσεις που έγιναν για το συγκεκριμένο έλεγχο. Επίσης αυξήθηκε και η ομοιότητα των κατηγοριών από το 60% στο 70%, με μόνη εξαίρεση τα αποτελέσματα που αφορούσαν τη βάση δεδομένων που περιείχε μόνο controls, όπου υπήρχε

60% ομοιότητα. Επίσης μειώθηκε ο αριθμός των subjects στις σημαντικές κατηγορίες σε σχέση με το προηγούμενο δίκτυο. Αυτή η αύξηση στη συνέπεια και η μείωση του αριθμού των subjects στις κατηγορίες, υπήρξε λόγο του ότι πλέον το δίκτυο ήταν μεγαλύτερο και υπήρχε η δυνατότητα οι διάφορες κατηγορίες να ανεξαρτηκοποιηθούν και να συγκεκριμενοποιηθούν περισσότερο. Για παράδειγμα, αν υπήρχε κάποιος μικρός αριθμός subjects με κάποια κοινά χαρακτηριστικά στα SNPs τους, αντί να αναγκαστούν να μπουν στην ίδια κατηγορία με κάποια άλλα που δεν έχουν και τόσα κοινά χαρακτηριστικά μεταξύ τους, τώρα ο SOM μπορεί να δημιουργήσει μια ξεχωριστή κατηγορία για αυτά. Είναι χαρακτηριστικό το γεγονός ότι σε αντίθεση με τις 90 περίπου κατηγορίες που δημιουργήθηκαν στο δίκτυο μεγέθους 10x10, σε αυτό το δίκτυο δημιουργήθηκαν περίπου 300.

5.3.3 Δίκτυο διαστάσεων 50x50

Στο τελευταίο και μεγαλύτερο δίκτυο, είχαμε τα καλύτερα αποτελέσματα όσο αφορά την εύρεση των σημαντικών κατηγοριών. Συγκεκριμένα η εύρεση της κατηγορίας με τους 51 subjects στους cases και controls με p-value = 4.36×10^{-6} δίνει και στατιστική αξία στα αποτελέσματά μας όσο αφορά τη βιολογική τους σημασία. Επίσης η μη εύρεση μεγάλων κατηγοριών στους controls μας επιβεβαιώνει ότι ο αλγόριθμος λειτούργησε σωστά, αφού βιολογικά δεν αναμενόταν να βρεθούν κοινές περιοχές σε μεγάλο αριθμό subjects που δεν πάσχουν από τη συγκεκριμένη ασθένεια. Σχεδόν σε όλες τις εκτελέσεις, οι σημαντικές κατηγορίες είχαν ομοιότητα 95% με μόνη εξαίρεση τις εκτελέσεις που αφορούσαν τη βάση δεδομένων με τους controls, οι οποίες είχαν 90% ομοιότητα. Εδώ λόγο του ότι το δίκτυο ήταν πολύ μεγάλο, δόθηκε η δυνατότητα της πλήρης συγκεκριμενοποίησης των κατηγοριών, με αποτέλεσμα σε κάθε κατηγορία να υπάρχουν μόνο subjects με μεγάλη ομοιότητα στα SNPs τους. Έτσι σε κάθε κατηγορία είναι πιο εύκολο να αναγνωρίσεις τα κοινά τους χαρακτηριστικά και στη συνέχεια να τα συγκρίνεις με τα κοινά των υπόλοιπων κατηγοριών ώστε να βρεις τυχόν διαφορές μεταξύ τους. Αυτή η συγκεκριμενοποίηση μείωσε όμως και τον αριθμό των subjects με αποτέλεσμα να υπάρχουν πολλές κατηγορίες με πολύ μικρό αριθμό subjects και οι οποίες δεν μπορούν να χρησιμοποιηθούν.

5.4 Σύγκριση γραμμικής και δυαδικής κωδικοποίησης

Μεταξύ των δύο κωδικοποιήσεων υπήρχαν κάποιες διαφορές στα πιο μικρά δίκτυα, αλλά στο μεγάλο δίκτυο όπου είχαν και ανεξάρτητα μεγάλη συνέπεια οι κατηγορίες τους, είχαν και μεταξύ τους πολύ μεγάλη ομοιότητα. Συγκεκριμένα οι κατηγορίες που προέκυψαν από τις

δύο διαφορετικές κωδικοποιήσεις, είχαν μεταξύ τους 95% ομοιότητα, και στις μεγάλες κατηγορίες οι διαφορές ήταν σε ένα με δύο subjects το πολύ. Από αυτό προκύπτει ότι οι δύο κωδικοποιήσεις δίνουν σχεδόν τα ίδια αποτελέσματα, οπότε μπορούν να χρησιμοποιηθούν και οι δύο στα συγκεκριμένα δεδομένα χωρίς να δημιουργήσουν κατηγορίες με σημαντικές διαφορές.

Κεφάλαιο 6

Συμπεράσματα

6.Γενικά

Ο SOM γενικά με τις βελτιστοποιήσεις και την παραλληλοποίηση, είχε πολύ καλά αποτελέσματα όσο αφορά το χρόνο εκτέλεσής του, κάτι που είναι χρήσιμο σε αυτή την εφαρμογή, αφού είναι αρκετά χρονοβόρα. Οι δύο βελτιστοποιήσεις έχουν βοηθήσει αρκετά και πιστεύεται ότι μπορεί να γίνει καλύτερη εκμετάλλευση της παραλληλοποίησης αν γίνει και καλύτερη κατανομή των νευρώνων στους επεξεργαστές.

Από τα αποτελέσματα που προέκυψαν στο δίκτυο με διαστάσεις 50x50, είναι εμφανές ότι η εφαρμογή δημιούργησε σωστά τις κατηγορίες και μας επιτρέπει να ασχοληθούμε περαιτέρω με αυτή την εργασία, για εξαγωγή καλύτερων αποτελεσμάτων. Βάση των αποτελεσμάτων που είχαμε στα 3 διαφορετικά σενάρια που τρέξαμε, είναι φανερό ότι με την αύξηση του μεγέθους του δικτύου αυξάνεται η συνέπεια των αποτελεσμάτων του αλγορίθμου, ενώ μειώνεται και ο αριθμός subjects κάθε κατηγορίας. Ανάλογα με το πως θα χρησιμοποιήσεις τα δεδομένα μπορείς να ακολουθήσεις δύο μεθόδους. Μπορείς είτε να έχεις πιο γενικές κατηγορίες και σε αυτές που σε ενδιαφέρουν περισσότερο να ξανατρέξεις το SOM ώστε να ξεχωρίσει περαιτέρω τις κατηγορίες αυτές, είτε να έχεις τις κατηγορίες όσο πιο ανεξάρτητες γίνεται χρησιμοποιώντας ένα δίκτυο μεγάλου μεγέθους, και στη συνέχεια να συγκρίνεις τα κοινά χαρακτηριστικά κάθε κατηγορίας για να εντοπίσεις τις διαφορές τους.

6.2 Μελλοντική εργασία

Ένας από τους μελλοντικούς στόχους, είναι η καλύτερη κατανομή των νευρώνων στους επεξεργαστές, ώστε το speedup του αλγόριθμου να είναι πιο σταθερό. Επίσης θα δοκιμαστεί η παραλληλοποίηση του batch αλγόριθμου ο οποίος θα περιέχει λιγότερους συγχρονισμούς και πιθανόν να μας δώσει πολύ καλύτερο speedup. Τέλος θα μελετηθεί το πώς μπορεί να

χρησιμοποιηθεί στη σήμανση κάθε κατηγορίας και η πληροφορία των γενετικών δεδομένων των subjects.

Βιβλιογραφία

[1] Holger Schwender& Katja Ickstadt ,” Identification of SNP interactions using logic regression”. Biostatistics Advance Access, 2007

[2] Alberts, B., Bray, D., Johnson, A., Lewis, J., Raff, M., Roberts, K., et. al., “Βασικές Αρχές Κυτταρικής Βιολογίας: Εισαγωγή στη Μοριακή Βιολογία του Κυττάρου,” 2000

[3] Abecasis, G.R., Cookson, W.O. & Cardon, L.R, “Selection Strategies for disequilibrium mapping of quantitative traits in nuclear families,” Am. J. Hum. Genet., vol. 65, pp. A245, 1999.

[4] Cordell, Heather J."Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans". *Human Molecular Genetics* 11 (20): 2463–8 (2002).

[5] S. E. Baranzinin, J. Wang, et al, "Genome-wide association analysis of susceptibility and clinical phenotype in multiple sclerosis," *Human Molecular Genetics.*, vol. 18, no. 4 pp. 767-778, November 2008

[6] G. R. Abecasis, S. S. Cherny, W. O. Cookson and L. R. Cardon, “Merlin-rapid analysis of dense genetic maps using sparse gene flow trees,” Nature Genetics vol. 30, pp. 97-101, 2002.

[7] Athos Antoniades, L. Loizou, A. Aristodimou, C.S. Pattichis, A binary format for genetic data designed for large whole genome studies that enable both marker and strand based analyses, in CD-ROM Proceedings of the 8th IEEE International Conference on BioInformatics and BioEngineering (BIBE 2008), October 8-10, Athens, Greece, 6 pages, 2008

[8] S. Purcell, B. Neale, K. T. Brown, L. Thomas, M. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. W. de Bakker, M.J. Daly and P. C. Sham, PLINK: a toolset for whole-genome association and population-based linkage analysis. [Open Source]. Massachusetts General Hospital: American Journal of Human Genetics vol. 81, 2007.

[9] Mat Buckland, "Kohonen's Self Organizing Feature Maps," ai-junkie.com, 2002. [Online]. Available: http://www.ai-junkie.com/ann/som/som1.html [Accessed: July. 06, 2009].

[10] Jeremy Wyatt, "Kohonen Networks," cs.bham.ac.uk, July 1999. [Online]. Available:http://www.cs.bham.ac.uk/~jlw/sem2a2/Web/Kohonen.htm[Accessed: July. 07, 2009].

[11] Bashir Magomedov, "Self-Organizing Feature Maps (Kohonen maps)", codeproject.com, 07 November 2006. [Online]. Available: http://www.codeproject.com/KB/recipes/sofm.aspx [Accessed: July. 08, 2009].

Παράρτημα Α

Στο παράρτημα αυτό, θα γίνει μια αναφορά στις δημοσιεύσεις που προέκυψαν από την μελέτη αυτή. Έχει ήδη εκδοθεί ο αλγόριθμος συμπίεσης στο στάδιο της προεπεξεργασίας

[17]. Η επόμενη δημοσίευση θα αφορά τα αποτελέσματα των ομάδων που προέκυψαν με την χρήση του παράλληλου αλγόριθμου σε γραμμικά δεδομένα εισόδου στις βάσεις δεδομένων που χρησιμοποιήθηκε, σε συνδυασμό με τις επιδόσεις που εμφανίζει ο αλγόριθμος αυτός. Η τρίτη δημοσίευση θα αφορά τις βελτιστοποιήσεις που έγιναν στον παράλληλο αλγόριθμο, και το γεγονός ότι η μια μπορεί να εφαρμοστεί και στον απλό σειριακό αλγόριθμο μάθησης χαρτών αυτό-οργάνωσης.

Ακολουθεί η δημοσίευση που έχει ήδη εκδοθεί.

A binary format for genetic data designed for large whole genome studies that enable both marker and strand based analyses

Athos Antoniades, Loizos Loizou, Aristos Aristodimou,
Constantinos S. Pattichis, *Senior Member, IEEE*

Manuscript received August 8, 2008.

Athos Antoniades is with the Department of Computer Science, University of Cyprus, Nicosia, CY (corresponding author: +357-22-892685; e-mail: athos@ cs.ucy.ac.cy).

Loizos Loizou is with the Department of Computer Science, University of Cyprus, Nicosia, CY. (e-mail: cs06ll1@cs.ucy.ac.cy).

Aristos Aristodimou is with the Department of Computer Science, University of Cyprus, Nicosia, CY (e-mail: cs06aa2@cs.ucy.ac.cy).

Constantinos S. Pattichis is with the Department of Computer Science, University of Cyprus, Nicosia, CY (e-mail: pattichi@cs.ucy.ac.cy)

***Abstract*— Recent advances in genotyping technology have enabled large studies with data from thousands of subjects to contain half a million or more of single nucleotide polymorphisms (SNPs) marker per subject. This rapid increase in the size of data has generated the need to compress the data in order to reduce the storage capacity requirements and the memory required at run time to perform analysis on the data. The availability of so many markers across the whole genome has created opportunities for new methodologies to be implemented that take advantage of the relatively high density of the markers to perform analyses that take into account the Linkage Disequilibrium (LD), an effect where some combinations of genetic markers are non-randomly associated. Classical techniques for transforming genotypic data into a binary format are already in use by several applications however we demonstrate in this paper that the traditional transformations are not adequate for certain types of analyses as some information key to new methodologies of analyses is lost. We propose a new protocol for formatting binary genotypic data that can be used in all types of analyses while still offering a very high compression rate.**

Introduction

Until recently in genetic studies, due to limitations in genotyping technology, only a small subset of the genome could be analyzed. The introduction of affordable high throughput genotyping technologies allowed the assay of more than half a million loci variants (SNPs) per subject across the whole genome. Genetic association studies applying such technology have now become common as they allow investigation of the majority of common loci variants in the genome; such studies are typically called genome wide association scans (GWAS). In this paper we present a non lossy format of encoding the data in the datasets generated from GWAS to a binary form.

The substance that encodes the genetic instructions of living organisms is Deoxyribonucleic acid (DNA). DNA consists of two long units called strands having a shape of a double helix [1], [2]. The genetic code is specified by the four nucleotide "letters" A (adenine), C (cytosine), T (thymine), and G (guanine). There are multiple different types of variations that can occur on the DNA strands, however traditionally the genetic analyses seem to focus on analyzing Single Nucleotide Polymorphisms (SNPs). SNP variation occurs when a single nucleotide, such as an A, replaces one of the other three nucleotide letters—C, G, or T [3]. For a variation to be considered a SNP it also needs to be present in at least 1% of the population. Due to the Linkage Disequilibrium effect (LD), SNPs serve as biological markers for pinpointing a region on the genome associated with a phenotypic trait even thought the SNP itself is not necessarily the variation responsible for the association. Rather it's one or more of the variations that are in LD with the SNP that is causing the association.

The need to reduce the size of the data is owed to the 100 fold increase in the genotyping capacity available today combined with the massive reduction in cost. Today's technology has the ability to analyze datasets up to 550,000 or even 1 million SNPs per subject with >99% accuracy, at a rate of >100 K genotypes per day and at a cost of around 20–30 cents per genotype [4], [5], [6].

Traditionally the genetic data in these studies is stored in the QTDT format introduced in the program Merlin [7]. Input files describe relationships between individuals in a dataset, store marker genotypes, disease status and quantitative traits and provide information on marker locations and allele frequencies.

TABLE I
PLINK BINARY PED FILE ENCODING [8]

Allele On Strand +	Allele On Strand -	Marker Encoding
A	A	00
A	a	01
a	A	
A	a	11
Missing	Missing	10

There is already a technique for compressing this data by utilizing a binary encoding [8]. However, that technique was focused at encoding SNPs as markers to be used in analyses rather than strand based information. Today as more GWAS datasets became available researchers are developing innovative new methodologies to analyze them. Some of these methodologies are not relying so much on the SNPs as markers; rather they look at the sequence of genotyped alleles on each strand of DNA separately. The methodology used in plink to encode the data looses the information of which strand holds each allele's genotype for heterozygote SNPs, therefore making it impossible to run analyses that use strand information using the binary input format for GWAS. Also, recent technological advances in genotyping are enabling the detection of deletion regions. This may result in future datasets to incorporate markers in them that may have an allele missing not due to a genotyping error but due to a deletion on one of the two strands. The encoding format proposed in this paper will offer an efficient lossles compression that is also capable of encoding deletions separately from errors in genotyping or quality control removed markers.

The structure of the paper is as follows: Section II Presents the traditional methodology of formatting GWAS data as well as the current alternative to compressing the data and the format proposed in this paper. Section III Offers a comparison of the three formats identifying advantages and disadvantages of each. Section IV concludes describing the situations under which each format is optimal.

Methods and material

The commonly used format QTDT Merlin [7]

The QTDT file format described in Merlin is the one traditionally used for this type of data. It's split into three files; Pedigree files contain phenotypes for discrete and quantitative traits and marker genotypes for a specific number of subjects. They are usually white-space delimited files. The first (usually 6) columns contains information about the subject (Family ID, Individual ID, Paternal ID, Maternal ID, Sex, Phenotype). The combination of the information of each subject must be unique. The next columns contain biallelic markers; typically SNPs. Marker genotypes are encoded as two consecutive integers, one for each allele, or using the letters "A", "C", "T" and "G". To denote missing alleles a sentinel value is used, typically "0".

Map files contain information for each single nucleotide polymorphism. They are used to analyze genetic markers into the equivalent pedigree file. Each line per marker usually contains 3, 4 or 5 columns (chromosome, SNP identifier, morgans or centimorgans and base-pair position). Each column is separated by white space.

Dat files describe the pedigree file. They include one row per data item in the pedigree file, indicating the data type providing a one-word label for each item.

Plink's Method for binary Ped Files [8]

Plink is an excellent, open source program offering a comprehensive range of basic large-scale whole genome association analysis methodologies. It has been widely adopted since high dimensionality GWAS have become available as it enables researchers to efficiently analyze these large datasets in a computationally efficient manner.

In plink there is an encoding format for transforming QTDT Merlin data into binary formatted files. The approach used in plink uses 2 bits for encoding biallelic markers with 4 possible states. Plink uses the encoding for each genotypegiven in Table I. [8]

Testing on plink binary format showed that the exported binary file was 15 times smaller than the original file. The drawback however is that encoding of the heterozygote allele is the same regardless of what strand it's actually derived from. Therefore any analyses that rely on the sequence of the alleles on the strand will be missing this information.

One analyses technique that needs the lost information is imputation. Imputation analysis is the practice of 'filling in' missing data with plausible values. It is a method for uncovering the genetic basis of human disease and it is used for inferring genotypes at observed or unobserved SNPs that can detect causal variants that have not been directly genotyped [9]. It is in essence an in-silico approach to discover the probability of the existence of a specific genotype for loci that haven't being directly genotyped but are known to be in LD with genotyped markers.

Proposed Method

Since imputation analyses require knowledge of which strand the alleles of heterozygote SNPs exist on, we need to encode each allele on each strand separately for all cases. There are a minimum of three states each allele can be in, these include the two possible nucleotides (commonly denoted as A and a) as well as the possibility of missing data at that location. The smallest number of bits that can encode the 3 states of an allele is 2, however with 2 bits we can actually encode a fourth state. In many existing studies this may not be used, even though it will have no impact on the capacity requirements the databases will have for storage. However, we propose that the fourth state is set to denote markers that are in deleted regions. This utilizes the extra available coding identifying a deleted allele from a missing allele due to quality control concerns, or genotyping error.

An analytical technique that enables the detection of these deletions that has started being applied is copy number variants (CNV) analyses. It refers to the genetic trait of differences in the number of copies of a particular region (for example a gene) present in the genome of an individual [10], [11]. To perform CNV analyses most algorithms rely on raw data from the genotyping platform. CNV algorithms are able to detect deletion regions as well as regions that are duplicated that may exist in some individual's strands.

However it should be clearly noted that CNV incorporates more information than just deleted regions. It can also detect regions that exist in more than one copy per strand. That information is not reflected in the proposed protocol; therefore methodologies that use that information would still rely on an external file with CNV regions.

In the proposed format the data is structured as a two dimensional vector of alleles. The first dimension's size is equal to the number of strands the subject has. Typically in humans all chromosomes have two strands with the exception of X and Y chromosomes in males that each have 1 strand. Even in these cases a second strand can exist listing the alleles of the second strand on males as missing.

Each element in the vector of each strand will encode an allele. The allele will be 2 bits long enabling encoding of a total of 4 states per allele, missing data, nucleotide 1, nucleotide 2 or deleted.

Table II presents the 4 different states that can be coded per allele. The term "Unknown" is used rather than the more typical "missing" to denote

alleles that it's unclear what their genotypes are or if they are deleted so as not to confuse it with the deleted state that defines alleles that do not exist on that strand.

Table III shows how two alleles are encoded and create a biallelic allele such as SNPs, while still enabling the identification of deleted alleles from missing values if that information is available. By comparing two alleles together and using the encoding as proposed above the resulted encoding is shown in Table III.

The two strands in each Subject's vector need to be perfectly aligned, that is, the *i* th element of each vector will point to the same Marker's alleles one for each strand. This makes it easy to access information in the way it was traditionally accessed. To access the *i*'th marker's alleles the two bits at position *i* in each strand will carry a total of 4 bits, using the encoding column of Table III the genotype of Marker *i* can then be identified.

Experimental results

To compare the different formats we will consider the size of the resulting files in relation to the original QTDT Merlin format, the amount of information lost through the encoding of the original QTDT Merlin format and the ability of each format to retain different types of information available today. Table IV provides a summary of the comparison.

Information loss

On one hand loosing information due to encoding is undesired; however, if the encoding is used to simply speed up analyses and reduce scratch space then it's not an issue so long as the original raw file is kept for future analyses. However if you are developing an algorithm that needs the information of which strand each heterozygote marker's alleles are on , or if there are deletion regions overlapping the markers in either strand, then utilizing the encoding methodology proposed in this paper will in a single table or file encode all of that information efficiently. Another issue is the actual storage of the data for long term use or for transferring over the internet. Utilizing a non-lossy approach to compressing the data that incorporates all genetic information into a single file can reduce the resources required.

Added Information Capability

In this paper we focused on bi-allelic markers as they are the ones that current high throughput genotyping technologies are able to genotype. However it should be noted that the format of QTDT Markers enables encoding of tri or quad allelic markers as well since each allele is encoded as an ASCII character. Neither plink's binary ped file nor the format proposed could handle tri or quad allelic markers. Large datasets with tri-allelic markers are not existing today (to the knowledge of the authors) while information on deleted markers is available through CNV analyses and other methodologies available to the various genotyping

TABLE II
PROPOSED ALLELE ENCODING

Allele	Encoding
Unknown	00
A	01
a	10
Deleted	11

TABLE IV
COMPARISON OF FILE FORMATS

QTDR Merlin	Proposed Protocol	Plink 's protocol
Information Loss for bi-allelic markers		
Used as Reference	None	strand location of heterozygote alleles Aa,aA Missing one of the two alleles A0, 0A, a0,0a
Added Information capability		
tri or quad allelic markers	Alleles deleted from a specific strand	None
Size*		
Pedigree File 3.6 GB Map File 12.6 MB **Total:** 3.612 GB	One binary file 496 MB	.bed file : 229.6 MB .fam file : 30 .bim file : 14.1 MB **Total** : 243.7 MB

TABLE III
PROPOSED ENCODING FOR BI-ALLELIC MARKERS

Allele 1	Allele 2	Encoding
Unknown	Unknown	0000
Unknown	A	0001
Unknown	a	0010
Unknown	Deleted	0011
A	Unknown	0100
A	A	0101
A	a	0110
A	Deleted	0111
a	Unknown	1000
a	A	1001
a	a	1010
a	Deleted	1011
Deleted	Unknown	1100
Deleted	A	1101
Deleted	a	1110
Deleted	Deleted	1111

platforms. Therefore tri-allelic and quad allelic markers were ignored in this encoding until high throughput technologies genotyping them are available making it necessary to generate a new protocol for that.

Size of encoded data

The compression rate can easily be estimated since both encodings are deterministic, however we also provide a as an example an actual test we performed using an average dataset size of 1806 subjects and 532,579 markers. The plink binary ped file was able to compress the file to $1/15^{th}$ of it's original size while the proposed method compressed the file to exactly double the size of that achieving just $1/7.5^{th}$ of the original size. However, both compressions produce enough of a compression to overcome the issue of the large data since the data can now fit on the average computer's physical memory (RAM) as today's typical computer has at least 1 GB.

Conclusion

The proposed format encodes genotypic data into a binary form in order to compress it and at the same time preserve all the information relating to the bi-allelic markers and the strand location of each allele. However it does double in size the resulting datasets from existing encoding methodologies that are lossy, but loose information only necessary to certain type of analytical approaches. The analytical approaches that would benefit from the proposed format of encoding are primarily the ones that take into account the strand on which heterozygote alleles are based on, the existence of the marker on just one of two possible strands, identifying if the second wasn't available due to genotyping errors or a deletion over the marker on that strand. Due to the need to use two bits per allele for encoding while only needing 3 states for each allele we were left with an available fourth state. That fourth state we proposed that is used to denote markers that were deleted as that information is becoming commonly available from genotyping platforms available already; however, future researchers may choose to use the fourth state to code a different state an allele can be in. Also in cases where compression of the data in a non-lossy way for storage, backup or data transfer is used the methodology proposed would be ideal.

REFERENCES

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walters, "Molecular Biology of the Cell 4th ed." New York and London: Garland Science, ch. 1, 2002.

[2] J. M. Butler, Forensic DNA Typing. Elsevier. ISBN 978-0-12-147951-0. pp. 14–15, 2001.

[3] M. P. Weiner, T. J. Hudson, Introduction to SNPs: Discovery of markers for disease, Biotechniques Suppl:4-7, 2002.

[4] S. Jenkins and N. Gibson, "High-Throughput SNP Genotyping," Comparative and Functional Genomics, vol. 3, no. 1, pp. 57-66, 2002.

[5] P. Y. Kwok, "High-throughput genotyping assay approaches," Pharmacogenomics, vol. 1, pp. 95–100, Feb 2000.

[6] J. N. Hirschhorn, M. J. Daly, Genome-wide association studies for common diseases and complex traits. Nature Review Genetics vol.6 pp.95–108, 2005.

[7] G. R. Abecasis, S. S. Cherny, W. O. Cookson and L. R. Cardon, "Merlin-rapid analysis of dense genetic maps using sparse gene flow trees," Nature Genetics vol. 30, pp. 97-101, 2002.

[8]

[9] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. R. Ferreira, D. Bender, J. Maller, P. Sklar, P.I.W De Bakker, Daly MJ and Sham PC, "PLINK: a toolset for whole-genome association and population-based linkage analysis," American Journal of Human Genetics, vol. 81, pp 559-75, Sep. 2007.

[10] J. Redon, "Global variation in copy number in the human genome," Nature, vol. 444, No. 7118, pp. 444-454, Nov. 2006.

[11] R. E. Fay, "Valid inference from imputed survey data," Proceedings of the Section on Survey Research Methods, American Statistical Association, pp. 41-48, 1993.

[12] J. L. Freeman, et al . "Copy number variation: New insights into genome diversity". Genome Research vol.16 pp. 949–61, 2006.