Detection of shared Copy Number Variation in patients co-hybridised to the same array

**Introduction**

Array comparative genomic hybridisation (CGH) is a commonly used diagnostic test in clinical genetics. The test utilises many probes (from 44,000 to 1,000,000) affixed to a glass slide. Each probe consists of many copies of an oligonucleotide 40-60 base pairs long, designed to target a specific region of the genome. Samples are hybridised to the array in a competitive reaction, usually a diagnostic sample in competition with a reference sample.

The diagnostic and reference samples are labelled with a fluorescent dye (patient Cy5 and reference Cy3). Once hybridised the array is scanned, capturing the excitation of each dye, producing a high resolution image of the array (Ahn et al., 2010). This image undergoes feature extraction to calculate the signal intensities at each probe and associated quality scores.

This data can then be analysed for copy number variation (CNV) by applying one of a number of available algorithms: The probe scores are normalised, split into segments of equal copy number and each segment assigned a copy number status (Hupe, 2004).

During data normalisation the signal intensities are calculated and given quality scores. Dividing the signal intensity of Cy5 by that of Cy3 produces a ratio where equal copy number is 1. This ratio is then logged (base 2) to produce the log2 ratio where equal copy number is 0, loss of material is indicated by a negative number and gain of material positive.

CNV is a deviation in copy number from a reference genome which typically contains 2 copies of a DNA segment (Roy and Motsinger Reif, 2013).

CNVs can occur in recombination and replication events. CNVs can be benign polymorphisms or associated witth Mendelian, sporadic and complex disease possibly through gene dosage, disruption, fusion or positional effects (Zhang et al., 2009).

In 2009 Array CGH started replacing karyotyping as the method for detection of CNV in NHS diagnostic genetic services. Array CGH offers a higher resolution, less reliance on analyst interpretation/skill and the advantage of a high throughput practical workflow, however few trusts actually have this test commissioned due to the higher cost. Guy’s and St Thomas’ NHS Trust adopted a patient to patient hybridisation approach (using 8x60K Agilent array design) which halves the cost of consumables by replacing the reference sample with another patient sample (Ahn et al., 2010).

As array CGH compares two samples the use of a reference sample infers any CNV detected is from the patient. Hybridising two patients removes this assumption, producing two challenges:

1. Is a CNV is a duplication in one patient, or a deletion in the second patient?
2. If both patients have the same CNV relatively no difference would be found, not detecting the CNV.



The first challenge can be overcome by comparing the signal intensities of each dye across the imbalance and across normal regions. The sample where the signal intensities within the imbalance are markedly different from the normal regions contains the imbalance.

The second challenge requires “a careful consideration of patient referral information” to reduce the risk of this occurring. Hybridisation partners are mismatched on phenotype (Ahn et al., 2010). This assumes that patients with differing referral reasons eg heart vs. renal defects will not have the same underlying CNV.

The product of the increased resolution of arrays is a higher abnormality pick up rate than karyotyping (25% vs 3.7%) (Ahn et al., 2013; Jordan, 2012).

**Project Aims**

The aim of this project is to create a tool which is run during data processing/analysis and detects CNV independent of the hybridisation partner.

Any imbalance found within each hybridisation partner can be compared to identify any overlapping regions which may not be identified using the signal ratio.

**Array design**

The arrays, reagents, equipment and software used to process and analyse are manufactured by Agilent Technologies (Agilent Technologies, Accessed 7 January 2015 a). The probe design is stored on the online probe catalogue/custom array design tool eArray (Agilent Technologies, Accessed 14 January 2015 b).

A request/suggestion for a tool similar to that which this project aims to create was sent and declined by Agilent.

The array is an 8x60K array with a median resolution of 120kb.

**Raw array data – Feature extraction file**

After the arrays have been processed in the laboratory and scanned the array image undergoes feature extraction

The feature extraction software converts the raw signal image into measurements for each probe. At each probe the excitation of each dye is captured as the signal intensity. This signal is processed, including a normalisation step across all probes producing a processed signal intensity normalised to 1000.

The two signal intensities are then converted into a log ratio (Log10)dividing the Cy5 signal intensity by the Cy3 signal intensity

Log10 ( Cy5 )

Cy3

The result of feature extraction is one feature extraction file per array, a tab delimited text file of 10-15mb in size.

The feature extraction file can be created in four levels of verbosity: full, compact (default), QC and minimal.

This file is split into three sections: metadata about the slide and experiment, array wide measurements and probe specific measurements. Each probe has a name, genomic location and physical position on the array and > 35 measurements including the log ratio, signal intensity and probe saturation for each dye.

**Use of algorithms in CNV calling.**

If all goals of this project are met the tool produced will have much the same role as the aberration detection algorithms, taking feature extracted data and looking for CNV.

There are a number of algorithms which are in use for CNV calling from CGH data. These algorithms are recursive binary segmentation methods which break down chromosomes into segments of equal copy number. Examples include Z-scores, circular binary segmentation (CBS), aberration detection method (ADM 2), nexus and hidden Markov model (HMM).

**Z-score algorithm**

Z-scores are a statistical measure of deviation from the mean for a normally distributed population. The Z-score algorithm starts off by giving a Z-score to signal log ratio for each probe. Any probes above a user defined Z-cut off can be classified as an outlier, or significantly away from mean.

The number of probes classified as above (R) and below this threshold (R’) and total number of measurements (N) are recorded and used when calculating the moving average of small windows of the genome.

These ‘windows’ can be specified as a number of adjacent measurements or a fixed size eg every 1MB. Within each window the abundance of probes which log ratios which deviate from the mean is measured (r:r’).

A Z-score is then calculated measuring the significance of the over-abundance of probes with a deviant score in this window (Agilent Technologies, 2012).

**Aberration Detection Method (ADM-1/ADM-2)**

These algorithms are designed by Agilent. These algorithms do not used fixed windows but segments the genome into intervals of equal copy number using signal log ratio scores from adjacent probes to best define interval breakpoints.

ADM-1

Firstly the data is normalised by subtracting the mean log ratio and dividing by the variance, creating a normally distributed population with a mean of 0.

Each chromosome is then broken into intervals of equal copy number and intervals are assigned a score (*S(I)*) which denotes the difference from the mean. A user defined threshold is set and any intervals which are above this are called as an aberration.

The interval with the highest score is selected and the same process is performed on this segment to further define breakpoints. This is repeated for all intervals with an *S(I)* above the threshold.

ADM-2

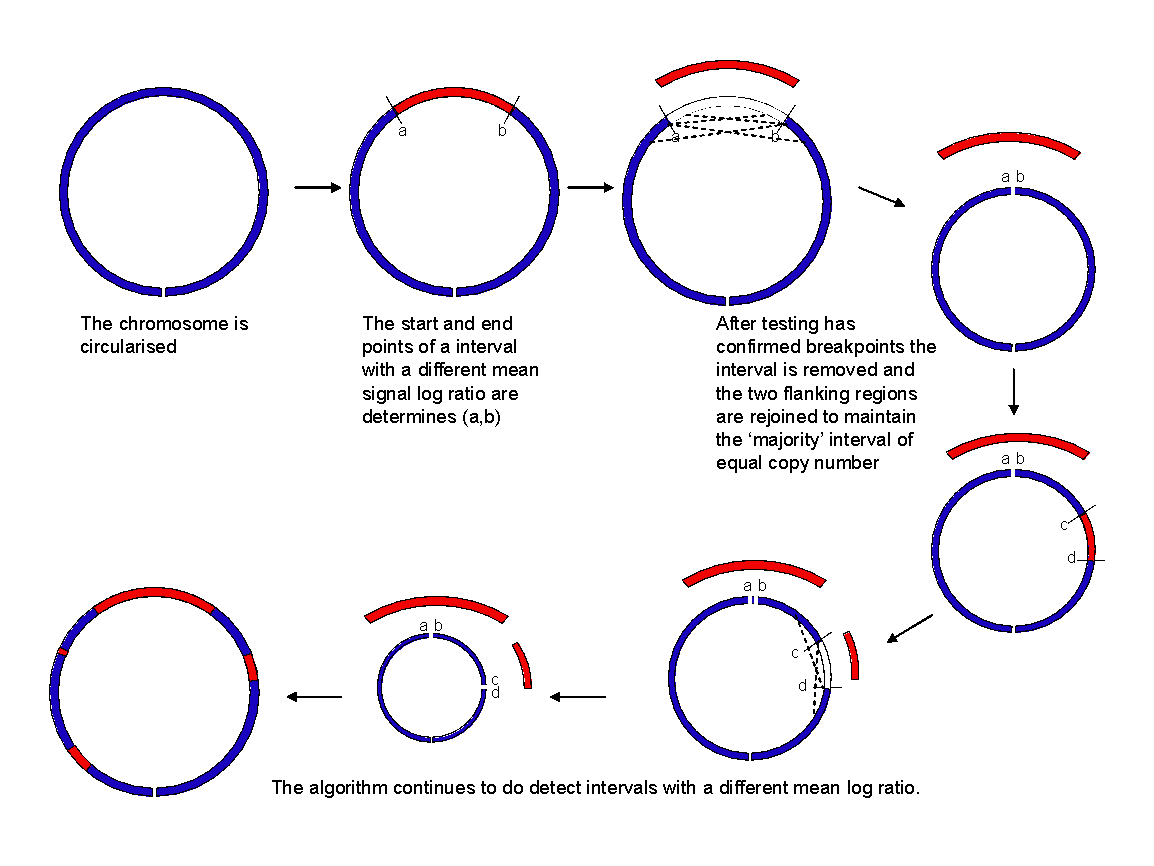
The ADM-2 algorithm builds on ADM-1 by including probe quality information to weight probe signal log ratios when assigning *S(I)*.

**Circular binary segmentation (CBS)**

Circular binary segmentation (CBS) also uses adjacent probe log ratio scores to create intervals which, as opposed to ADM-1/2 (which classifies segments as aberrant or not), are grouped into intervals with equal copy number.

Each chromosome is made into a circle (Figure 1). This allows for two break points to be identified which increases the resolution of detection. When two breakpoints are detected the interval of potential different copy number is removed, and the flanking regions are joined to form a new circle. This allows a t-test to be performed between the mean log ratios on each interval.

Definition of an interval is determined using permutation testing, creating intervals using various breakpoints and looking for the most significant P value. If this P value is above a threshold an interval is created and the process is repeated for all intervals until no more changes are found.

A number of checks are performed to ensure the correct end points have been found including edge effect correction, change point pruning and estimation of the log score distribution.

**Figure 1 - Circular binary segmentation algorithm**

**Hidden Markov Model**

The hidden Markov model (HMM) can also be applied to determine the aberration state of probes. HMM uses observations (signal log ratios) to determine the state of a hidden or latent value (copy number). The HMM has a chain property which takes into account the state of the previous probe.

Firstly the data is segmented: the Haar wavelet is used to normalise the data before breakpoints are defined using preset parameters (FDR threshold). This step also calculates probabilities and probability distribution parameters used in later steps.

The HMM is then applied. The forward-backward algorithm is used to calculate the posterior probabilities of the states. Baum-Welch learning uses these to ensure that each state has a well defined value.

Finally the Viterbi algorithm is used to assign a state to each probe.

**Algorithm Performance**

Each algorithm performs differently depending on the signal:noise ratio and the size of the aberration (Willenbrock and Fridlyand, 2005).

Circular binary segmentation (CBS) is rated as one of the most consistent performers albeit one of the most computationally demanding (Lai et al., 2005; Willenbrock and Fridlyand, 2005). Some algorithms performance varies greatly depending on user defined parameters which may be difficult to establish (Lai et al., 2005).

The algorithms described above are global segmentation methods which compare segments to the whole genome. Recently local segmentation methods have been described which looks for CNV using high resolution data sets (Niu and Zhang, 2012). These algorithms have higher power than global algorithms but are less robust so an approach utilising both algorithms may produce the best performance (Roy and Motsinger Reif, 2013).

# **The rationale for the project – sensitivity and specificity of a clinical tool.**

Financial pressures on the NHS are requiring increasingly innovative approaches. Adopting a patient to patient hybridisation approach halves the cost of performing the test.

One fear preventing adoption of a patient to patient approach in a clinical setting is the fear of missing a diagnosis due to shared CNV in hybridisation partners (Dunlop and Miller, 2015).

In a clinical setting it is important to have a high test sensitivity and specificity. Any missed diagnoses due to the patient to patient approach would be a false negative, a type two error, reducing the power of the test. A test must have a high specificity where as a screen can afford to have a higher level of false positives if followed by a test.

Any missed diagnosis misses a chance to be diagnosed, extending the patients diagnostic odyssey, or may result in a normal report incorrectly being issued for a prenatal test, potentially resulting in an affected child.

The chance of two patients having the same aberration (without phenotype mismatching) has been calculated as 1 in 6000

Whilst the chance of this occurring is low, and is mediated with strategies such as mismatched phenotypes, with 4000 arrays performed each year this could occur every 1.5 years (without phenotype mismatching).

Creation of a worksheet at the start of an array run must involve phenotype mismatching performed by a registered clinical scientist. The process occurs twice a week and takes 3 hours for a full run of 96 samples and the efficacy can be affected by incorrect, limited or vague phenotype information.

Simplifying the creation of a worksheet will save time, possibly reassigned to a lower grade staff member and may also be automated. This will further reduce the cost of performing an array test.

More importantly further reducing the chance of missing an CNV improves the sensitivity of the test.

**Data storage and manipulation**

This tool will require data to be stored, curated and accessed. A database is an efficient tool for this role.

Relational database

Relational databases are collections of tables where each row is an instance and each column is an attribute. Tables can be linked to store data in a simple and efficient manner. This can include the use of look up tables to speed up data searches.

Relational databases are stored in a relational database management system (RDBMS) server such as MySQL or MS-SQL which can be installed locally or on a server to allow access from remote machines. A graphical user inferface or ‘front end’ enables user interaction with the data, commonly seen in Microsoft Access or an internet browser.

Database design and data manipulation/retreival utilises the structured query language (SQL).

Non-relational database

MongoDB (REF) is an alternative database format to relational databases, storing data in BSON format, a binary form of the JSON format. JSON stores data as nested key-value pairs.

MongoDB does not require relationships between tables and fields to be defined, removing the need for complex joins and enables documents with differing formats to be easily stored and compared. MongoDB is faster and easier to scale than relational databases.

Interacting with databases

Programming languages such as Python have connectors (including MySQLdb (Dustman, 2014), SQLAlchemy (REF) and PyMongo(ref)) which allows interrogation of database from Python.

Modules, or packages, can be added to increase the functions available within Python including mathematical packages NumPy and SciPy (van der Walt et al., 2011).

Additonal software packages

An additional software package which may prove useful is statistical software R (R Foundation for Statistical Computing, 2014). Many of the CNV algorithms described above can be implemented within R such as CBS within DNAcopy (Seshan and Olshen, Accessed 15 January 2015 ).

**-----------------------------------------Section 2---------------------------------------------**

**Project background**

**Project scope**

The scope of this project is to investigate, design, build and trial a tool capable of automatic detection of CNV shared by both hybridisation partners.

**Requirements gathering**

Requirements were agreed with key stakeholders in the Array and Bioinformatics teams within the Viapath Genetic Laboratories:

Functional requirements

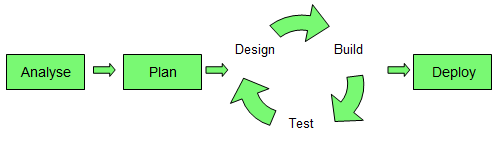
* Uses the signal intensities from the feature extraction file identify a region of abnormal copy number independently of the hybridisation partner
* An region of abnormal copy number is defined as a minimum of three consecutive abnormal probes.
* Report any common or overlapping regions of abnormal copy number present in both hybridisation partners.
* The test is a screen, sensitivity favoured over specificity.

Non functional requirements

* In line with the existing departmental resources, experience and skill sets data will be stored in an MS-SQL database hosted by a local server. This will ensure long term support.
* Any scripting should be performed in Python, in line with the existing department skill set to ensure the department can continue to support and develop the code and perform code review as per best practice guidelines.
* Any Python code must be written in Python must conform with the departmental Python style guide.
* Code must also be version controlled.
* Processing should be timely and not require excessive computational power
* The feature extraction files are required for analysis by Clinical Scientists so must not be modified or moved from the current location.
* The tool must be compatible with the trust operating system (Windows XP, with a future upgrade to Windows 10 planned before 2018)
* The tool must be run from multiple computers within the department by multiple users
* The analysis must be reproducible.
* The database must be in 3rd normalised form to ensure data integrity.
* Documentation
* Validation

**Project management???**

* Use an Agile development methodology to regularly test and refine the project.



* Use smartsheet (https://www.smartsheet.com/01) to set goals and monitor progress with stakeholders
* An electronic lab book is to be kept within the departmental wiki
* Weekly meetings will be held with key stakeholders.

**-----------------------------------------Section 3---------------------------------------------**

**Shared imbalance detection (SID)**

The Shared Imbalance Detection tool (SID) is able to detect shared imbalances by importing a feature extraction file into a database using a python script.

A Z-score is calculated for each sample at each probe. Any probes with a Z score outside a threshold is defined as abnormal. Any sequence of three or more consecutive probes called as abnormal in both samples are reported as a shared imbalance.

**Array design**

The arrays used are a custom array design, manufactured by Agilent. Other array designs are also in use within the department but are used for specific cases and are not included in the initial focus of this project

The design consists of 46,554 probes, 3886 are control probes and 42668 are targeted.

**Raw array data**

Feature extraction files consist of three sections, the parameters, stats and features. The feature extraction files are created in the minimal format.

Each slide contains 8 arrays. The parameters section contains 38 fields covering information relevant to the whole slide such as the scanner make and model, time, date and protocols or parameters used to scan the slide.

Stats contains 189 fields covering array specific measurements, such as average intensities and the derivative log ratio spread (DLRS) which is an indicator of quality used during analysis.

The majority of the file is made up by the features section with one row per probe consisting of 42 fields. 9 fields are identifiers such as the probe catalogue name, the genomic locations and flags to identify probes used as controls. The remaining 33 fields include the raw signal intensity measurement for Cy3 and Cy5 dyes, the log ratio, background signal intensities and flags to mark a probe as having an extreme signal intensity, saturated or non-uniform.

To maximise speed and reduce the storage requirements only the minimum required fields were selected to be imported into the database.

* Repeated probes, number of non-control probes etc.
* Stored in a directory used by other software so cannot be moved.

**Database structure – how the data will be held**

* SQL

Access to the MS-SQL hosted on the department server was not available during development. An MS-SQL installation was not possible on the development PC so development was performed using a MySQL server (v5.6) running locally on a development PC running Windows 10.

* Heidisql client

The HeidiSQL client (v9.1) was used to create the tables, view data and run queries.

* Describe the database structure

To maximise the performance data import was kept to a minimum.

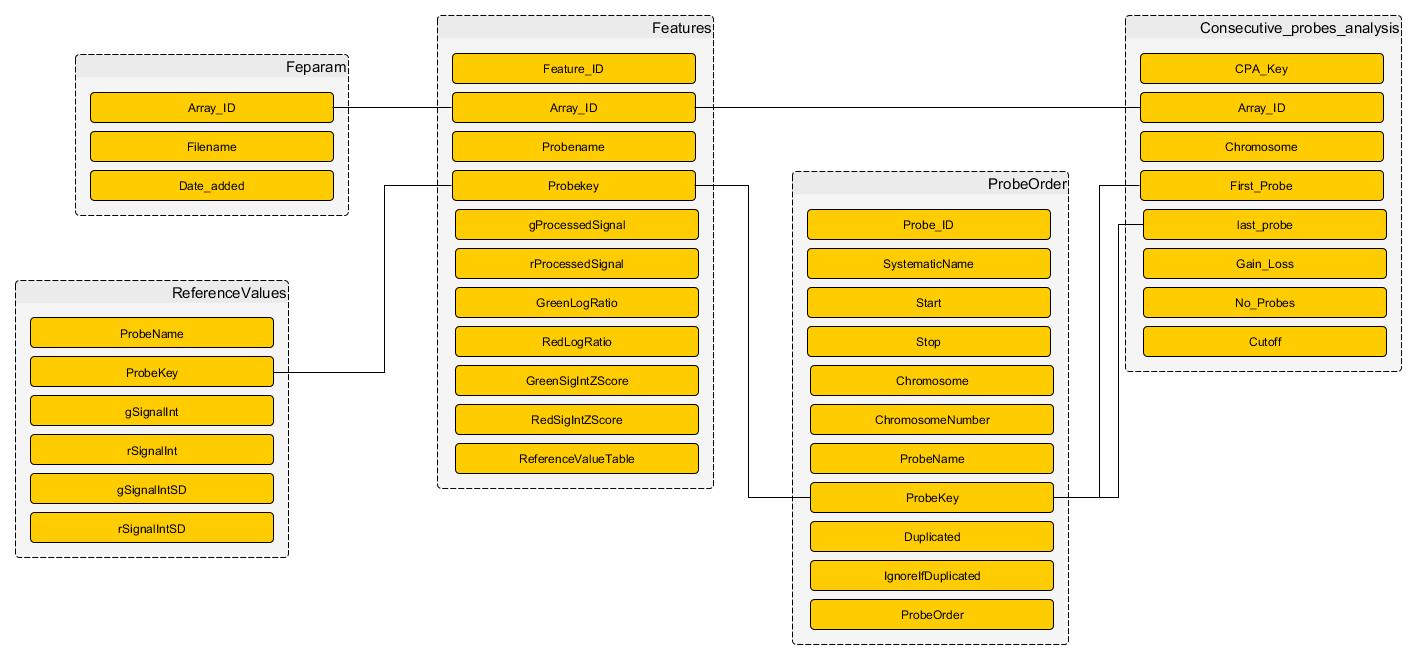
5 tables are required (fig).The Feparam table holds the filename to identify the array file used and upon insertion a unique identifier is used (Array\_ID). An automatic time stamp records when the analysis was performed.

Each probe on the array design is stored in the Probeorder table which gives each probe a numerical key (probekey) and orders the probes in order of genomic position. Probeorder only contains non-control probes and any duplicated probes are removed.

The ReferenceValues table holds the average signal intensity and the standard deviation from the normal control population.

The features table holds the analysis for each probe. Only the array\_ID, probename and processed signals are imported and the remaining fields are populated using information from other tables.

Results of analysis are entered into the consecutive\_probe\_analysis table enabling the aberration to be fully characterised.



* Indexes/keys

Each table has a primary key which is unique and indexed.

Any columns which are joined to the primary key of another table are foreign keys.

Columns involved in joins are also indexed.

* Compatibility/Conversion/transfer from development server (MYSQL) to production server (MSSQL)???

There are differences between MS-SQL and MySQL servers however conversion tools and strategies to migrate exist.

Test the queries in MSSQL

Also need to discuss a MSSQL python connector

**Importing the FE files – how to get data into db**

* Python

Feature extraction files were parsed using Python (v2.7). The Anaconda Python distribution was installed to utilise the conda package management system.

Anaconda has a number of additional packages preinstalled such as NumPy, SciPy and math packages. Conda was used to install the MySQLdb package (mysql-python v1.2.5) to access the MySQL database from python.

The Eclipse IDE (Luna v 4.4.1) was used with the PyDev package (v 3.9.0) to develop and run python scripts.

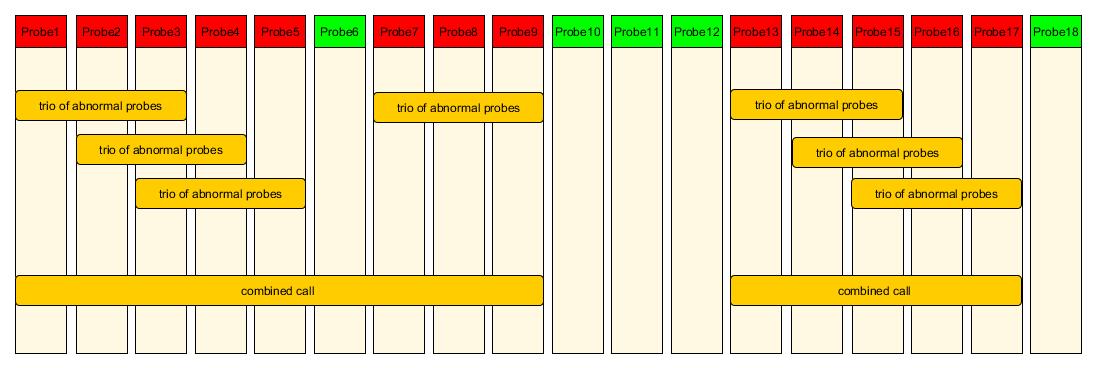
The Pylint Python package (v 1.4.2) and autopep8 were used within Eclipse to apply the local Python style guide to the code.

* Benchmarking?
* Version control

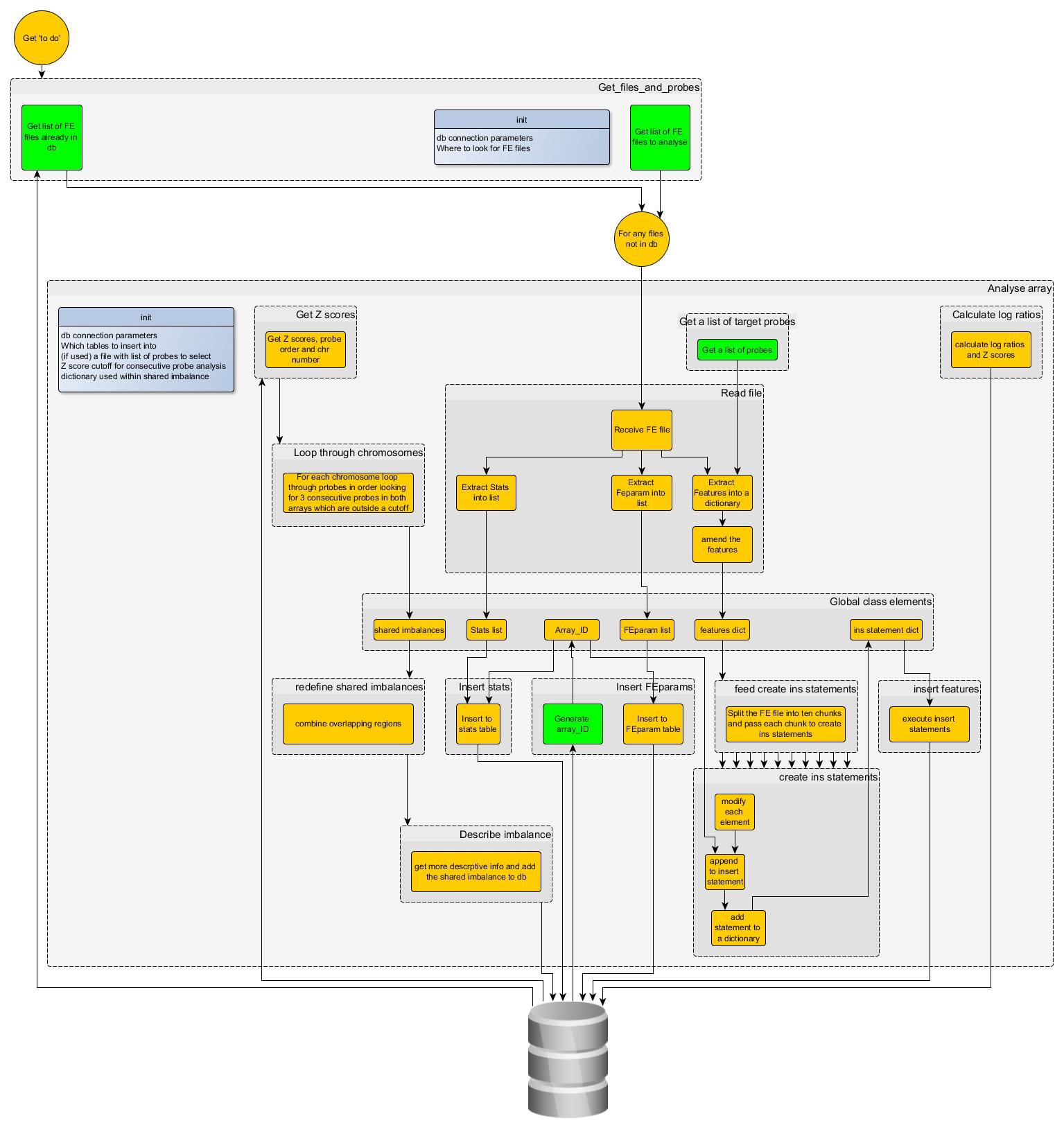
Python code was version controlled using Git and backed up using GitHub (<https://github.com/aledj2/FeatureExtraction>).

**How the tool works**

1. SID takes a list of files and processes each file one at a time. Currently the list is the contents of a given directory but can also be a text file containing a list of feature extraction filenames.
2. The first step is to check if the file has already been processed comparing the filename with those in the feparam folder. Any files which have already been processed are skipped.
3. The feature extraction file is then opened and each feature is added to an array.
4. The filename is added to the feparam table which generates the Array\_ID.
5. This key is captured within the python script.
6. The features are broken into 10 groups and an insert statement created for each group. This inserts the Array\_ID, probename and green and red processed signals into the features table.
7. An SQL update statement fills in the remaining fields in the features table adding the probekey and calculating the log ratios and Z scores.
8. The probes are then assessed looking for consecutive probes where the Z scores is above or below a set threshold which defines the probe as abnormal.
   1. For each chromosome look at each probe (x) in order of genomic position.
      1. If probe x has a Cy3 Z score above a threshold then look if probe x+1 is also above the threshold and if so assess probe x+2.
      2. If the same three probes also have a Cy5 Z score above the threshold the trio of probes is recorded as an abnormal trio and added to an array.
      3. repeat assessing if below the threshold.
   2. This is repeated for each probe, creating overlapping tiles of abnormal trios.
9. Any abnormal trios which are overlapping or separated by less than three normal probes are combined in line with the standard analysis process.
10. This combined segment is reported in the consecutive\_probe\_analysis.



A workflow of the SID steps is shown in FIG.



Choices:

* + How to select the files to add.
  + how insert is happening
  + Duplicated probes – average the SI
  + minimum size of segments
  + How to select which probes/regions to import/analyse
  + what if 5 consecutive probes have 4 abn probes but the two samples have different normal probes = the above algorithm will miss some.
  + number of arrays kept in features
  + different cutoffs for gain or loss

**-----------------------------------------Section 4---------------------------------------------**

**Analysis of probes – finding shared aberrations.**

**Choice of algorithm (Z score)**

A Z-score analysis method was selected for the analysis as the calls do not have to be as accurately defined as would be if was performing normal analysis. (screen not test)

Also wanted to keep the validation and maintenance simple.

Implementing a new algorithm would require a major validation of the algorithm.

A Z score provides a measure of how many standard deviations the signal intensity of a probe is from the mean of a normal population. A threshold can then be applied to score probes as normal or abnormal.

Z = Signal Intensity - Mean

SD

**Creating a normal reference range**

100 normal array cases were selected to make a reference range. Care was taken not to select only high quality arrays as this may result in many false positive calls in poor quality arrays.

The 100 cases were imported into the database. A python script was written to calculate the average signal intensity and the standard deviation for each dye at each probe.

These values are held in the reference\_values table.

The reference\_values table used is specified within SID, with the table name recorded in the features table. This enables the reference range to be periodically recalculated whilst ensuring analysis could be repeated if required.

The reference may require recalculating after the microarray scanner is serviced.

* Show signal intensity is normally distributed (Kurtosis scores) or pdf?(Ghasemi and Zahediasl, 2012)

In order to use a Z-score need to ensure reference range is normally distributed. show kurtosis.

**Creation of training set (true positives)**

A Python script was written to create 98 true positive feature extraction files using feature extraction files with reported aberrations.

The probes within the reported region were identified and a new FE file was created where the Cy3 signal intensity was copied into the Cy5 score.This was repeated copying the Cy5 signal intensity into Cy3.

This creates one true positive, with signal intensities representing an aberration in both hybridisation partners and one true negative file where both partners have normal signal intensities.

**Creation of test cases (EvE)**

N test cases were created using an updated strategy and Python script. Two samples which had exactly the same imbalance reported were hybridised post-hoc.

To ensure the signal intensities would be compared against the correct reference range a sample labeled with Cy3 was paired with one labelled with Cy5.

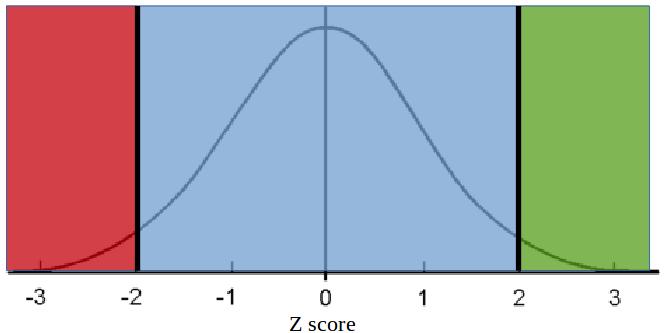
The python script parses both feature extraction files, extracts all measurements for that sample for each probe, and creates a new feature extraction file where the Cy3 and Cy5 measurements are populated from the Cy3 and Cy5 sample.

**Determining if a probe is normal or abnormal**

The Z score for each dye at each probe is calculated during the import of data and stored within the features table.

The Z score is a measure of many standard deviations from the mean the signal intensity is. A threshold is required to define a probe as abnormal.

Z scores are two tailed so probes with a Z scores above the threshold are defined as gains and Z scores below the negative threshold are defined as loss.

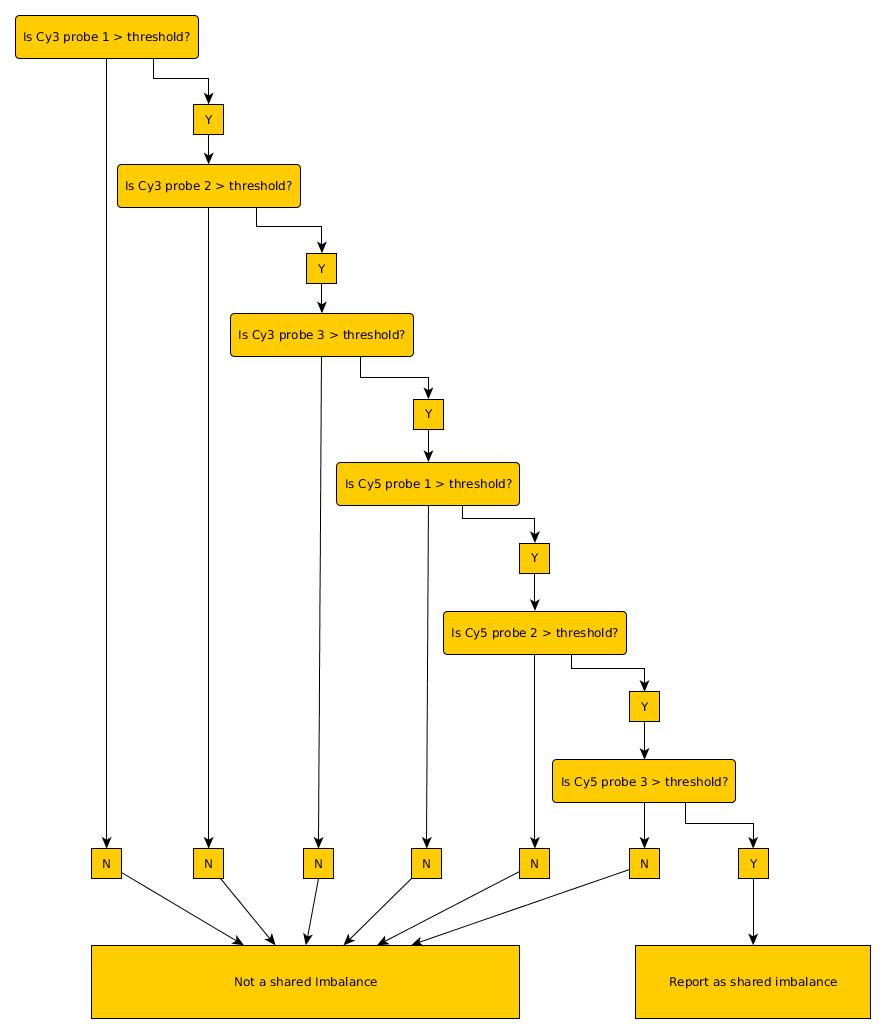


**Determining if a region is normal or abnormal**

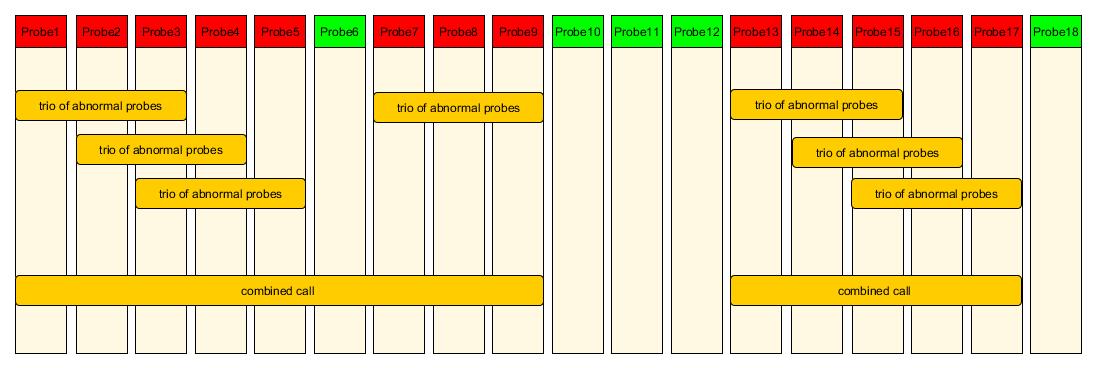
<This chapter discusses how probes were grouped into segments /regions >

To combine abnormal probes into segments a slightly different approach was taken to that described in Agilent’s aberration detection document (REF).

Aberrations below three probes are not reported so a sliding window of three probes was used, identifying shared imbalances of three consecutive probes (see image). The window moved along the entire chromosome creating overlapping tiles (image )



Regions of shared imbalance are then combined into larger regions in line with the analysis performed by the clinical scientists. combining trios that are separated by less than 3 normal probes.



**To test SID**

SID underwent three stages of development.

The training set of true positive was used to investigate and define the parameters required to call a probe and region as abnormal.

A test set of true positive cases was then used to ensure the parameters were not overfitted to the training set.

Finally a third set of cases was used as a prospective trial.

RESULTS

RESULTS OF Training CASES

= z score threshold

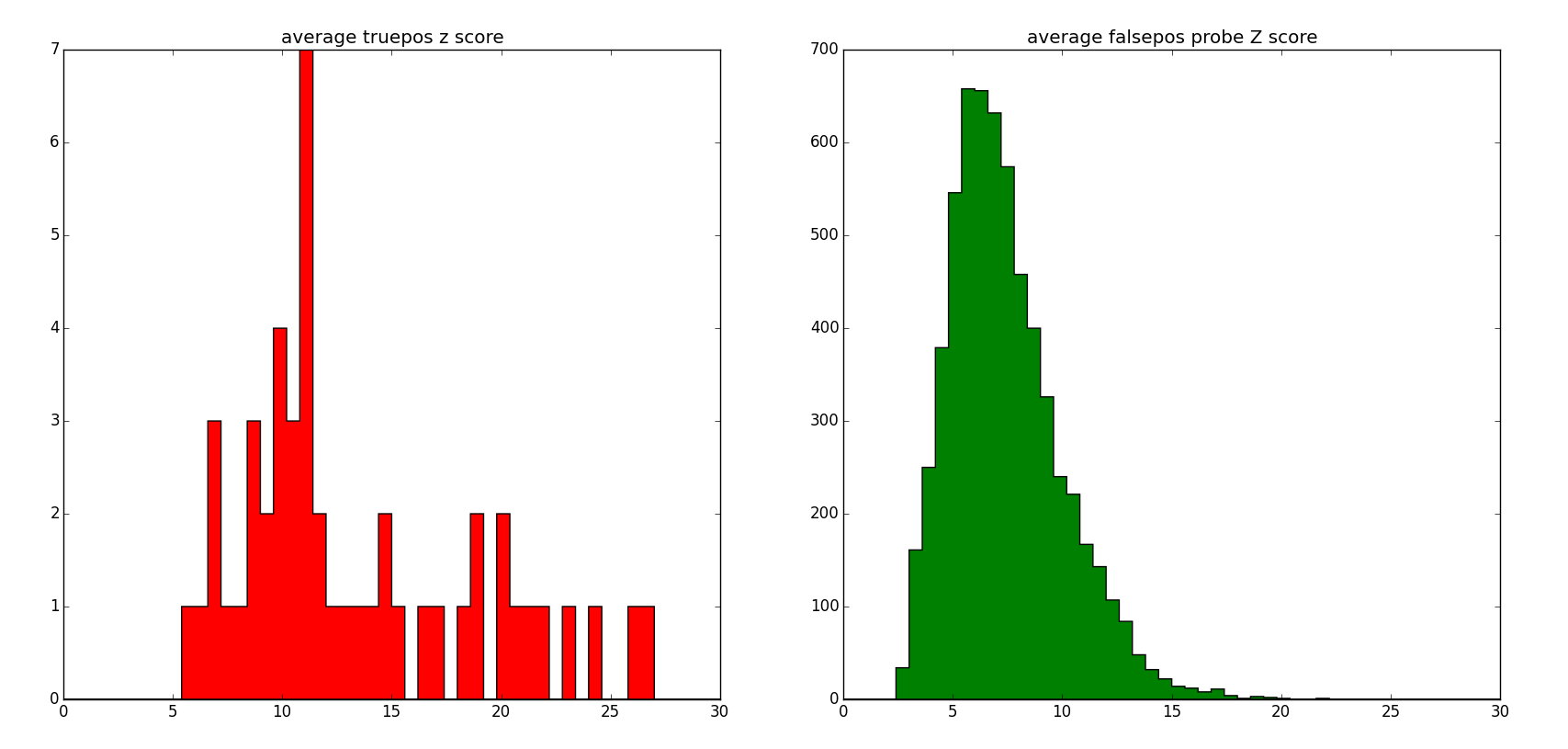
A Z score threshold of ±2.374 was used initially, defining abnormal probes as the upper and lower 1% of the reference range.

results

* false positives
* true positives
* specificity

The number of false positive calls per array varied greatly.

The false positives were investigated further to investigate if a different Z score or minimum number of probes within a segment could remove these FPs.



But the cutoffs without removing true positives still kept many FPs.

Could increase the min number of probes but want to keep in line with the analysis (3 probes).

Test cases

The training files are not truly representative of a true positive. Cy3 and Cy5 dyes are incorporated with differing efficacy and to determine if a probe is the signal intensity of one dye is to be compared with the reference range created from the other dye.

The same thresholds were applied to the test cases which are more representative of real abnormalities.

**Validation (real cases)**

<Normal arrays were run to check how it performed with normal ‘routine’ arrays without shared imbalances ie a trial run on normal arrays >

Hoping not to see any positive calls!

**Unit testing**

Create a truth set from the training set and an automated test.

**Longer term management/support**

(Automatically?) Truncate features table (can always be re-imported) to prevent db becoming too large and slow. The reference range table used can be amended to the feparam table (this is required to check which fe files are to be imported)

Periodically recalculate the reference range – the SI can be affected by scanner servicing.

If the array design were to change would need to update probeorder table and recalculate reference range.

**Documentation**

User manual

Development manual

Validation documentation (if get that far)

**-----------------------------------------Section 4---------------------------------------------**

**Discussion**

ROI vs consecutive probes

Briefly toyed with DNAcopy (R) but would be an extra thing to validate/dependency of the tool so not pursued.

Summary of tool and benefits (rationale and specification)

How to implement

* How to build link between the LIMS and this.
* How to run the program – automate (but what about different array designs)

Also talk about why I have taken the approach I did.

**References**

**Agilent Technologies (2012) *Aglient Workbench - CGH Interactive Analysis User Guide*. [online]. Available from: http://www.chem.agilent.com/library/usermanuals/public/g3800-90042\_cgh\_interactive.pdf (Accessed 20 October 2014).**

**Agilent Technologies (Accessed 7 January 2015 ) *CGH & CGH+SNP Microarrays* [online]. Available from: http://www.genomics.agilent.com/en/CGH-CGH-SNP-Microarrays/;jsessionid=Hz91Jt8CfVWDlHljTS2ytPCJG1R3P9xvNZqtS8JrtxP0wrThNLbH!867729385?pgid=AG-PG-9 (Accessed 7 January 2015).**

**Agilent Technologies (2015) *eArray* [online]. Available from: https://earray.chem.agilent.com/earray/ (Accessed 14 January 2015).**

**Ahn, J. W. et al. (2013) Array CGH as a first line diagnostic test in place of karyotyping for postnatal referrals - results from four years’ clinical application for over 8,700 patients. *Molecular Cytogenetics*. [Online] 616.**

**Ahn, J. W. et al. (2010) Validation and implementation of array comparative genomic hybridisation as a first line test in place of postnatal karyotyping for genome imbalance. *Molecular Cytogenetics*. [Online] 39.**

**Dunlop, J. & Miller, J. (2015) *Why haven’t you adopted a patient to patient approach?***

**Dustman, A. (2014) *MySQLdb1* [online]. Available from: https://github.com/farcepest/MySQLdb1 (Accessed 15 January 2015).**

**Ghasemi, A. & Zahediasl, S. (2012) Normality Tests for Statistical Analysis: A Guide for Non-Statisticians. *International Journal of Endocrinology and Metabolism*. [Online] 10 (2), 486–489.**

**Hupe, P. (2004) *GLAD-Introduction* [online]. Available from: http://bioinfo-out.curie.fr/projects/glad/node2.html (Accessed 7 January 2015).**

**Jordan, B. (ed.) (2012) *Microarrays in diagnostics and biomarker development: current and future applications*. Heidelberg ; New York: Springer.**

**Lai, W. R. et al. (2005) Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics (Oxford, England)*. [Online] 21 (19), 3763.**

**Niu, Y. S. & Zhang, H. (2012) THE SCREENING AND RANKING ALGORITHM TO DETECT DNA COPY NUMBER VARIATIONS. *The annals of applied statistics*. [Online] 6 (3), 1306–1326.**

**R Foundation for Statistical Computing (2014) *R: A Language and Environment for Statistical Computing*. Vienna, Austria: . [online]. Available from: http://www.R-project.org/.**

**Roy, S. & Motsinger Reif, A. (2013) Evaluation of calling algorithms for array-CGH. *Frontiers in Genetics*. [Online] 4. [online]. Available from: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3829466/ (Accessed 6 January 2015).**

**Seshan, V. & Olshen, A. (Accessed 15 January 2015 ) *DNAcopy : DNA copy number data analysis. R package version 1.40.0*. [online]. Available from: http://master.bioconductor.org/packages/release/bioc/html/DNAcopy.html (Accessed 15 January 2015).**

**van der Walt, S. et al. (2011) The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering*. [Online] 13 (2), 22–30.**

**Willenbrock, H. & Fridlyand, J. (2005) A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics*. [Online] 21 (22), 4084–4091.**

**Zhang, F. et al. (2009) Copy Number Variation in Human Health, Disease, and Evolution. *Annual Review of Genomics and Human Genetics*. [Online] 10 (1), 451–481.**