DETECTION OF SHARED COPY NUMBER VARIATION IN PATIENTS CO-HYBRIDISED TO THE SAME ARRAY

90565930

A dissertation submitted to the University of Manchester for the degree of Master of Science in the Faculty of Medical and Human Sciences;

The School of Medicine

2016

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CGH	Comparative Genomic Hybridisation
CNV	Copy Number Variation
CBS	Circular Binary Segmentation
HMM	Hidden Markov Model
AFU	Arbitrary Fluorescence Units
ADM2	Aberration Detection Method
RDBMS	Relational Database Management System
SID	Shared Imbalance Detection
PGD	Preimplantation Genetic Diagnosis
Cy3	Cyanine 3
Cy5	Cyanine 5
SQL	Structured Query Language

csv	comma separated values
GUI	graphical user interface
SOP	Standard Operating Protocol

something.

Dedicated to

ABSTRACT

Short summary of the contents...

DECLARATION

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning;

Aled Jones

INTELLECTUAL PROPERTY

ACKNOWLEDGMENTS

Something

1 INTRODUCTION

1.1 ARRAY CGH

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. Two 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 Cyanine 5 (Cy5) and reference Cyanine 3 (Cy3)). Once hybridised the array is scanned, capturing the excitation of each dye, producing a high resolution image of the array [1]. This image undergoes feature extraction to calculate the Cy5 and Cy3 signal intensities at each probe and associated quality scores.

This data can then be analysed for copy number variation by applying one of a number of available algorithms: Probe scores are normalised, split into segments of equal copy number and each segment assigned a copy number status [2].

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, a decrease in copy number is indicated by a negative value and increased copy number a positive value.

1.2 COPY NUMBER VARIATION (CNV)

Copy Number Variation (CNV) is a deviation in copy number from a reference genome which typically contains 2 copies of a DNA segment [3]. CNV can occur in recombination and replication events. CNV can be benign polymorphisms or associated with Mendelian, sporadic and complex disease possibly through gene dosage, disruption, fusion or positional effects [4].

1.3 APPLICATION OF ARRAY CGH IN CLINICAL SETTING

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 as a first line test due to the higher cost. Guy's and St Thomas' NHS Trust adopted a patient to patient hybridisation approach, halving the cost of consumables by replacing the reference sample with another patient sample [1].

The product of the increased resolution of arrays is a higher abnormality pick up rate than karyotyping (25% vs 3.7%) [5, 6].

1.4 PATIENT TO PATIENT HYBRIDISATION

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 a duplication in one patient, or a deletion in the second patient?

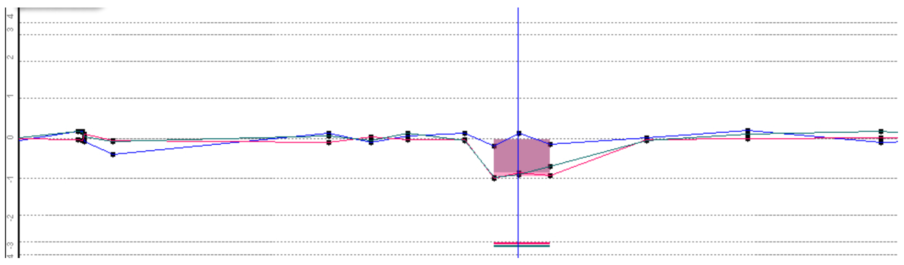


Figure 1: Three array traces are shown. Two arrays (Red and green traces) have the same three probe deletion on chromosome 6. The dark blue trace shows the two arrays hybridised together insilico. The log ratio appears completely normal and the deletion is not detected.

2. If both patients have the same CNV relatively no difference would be seen, resulting in a 'normal' log ratio so the CNV would not be detected.

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

The second challenge is currently overcome with "a careful consideration of patient referral information" to reduce the risk of this occurring. Hybridisation partners are mismatched on phenotype [1]. This assumes that patients with differing referral reasons eg heart vs renal defects will not have the same underlying CNV. Furthermore, CNV is rare and therefore the risk of two patients with the same CNV being hybridised together is low to start with, even without phenotype mismatching.

1.5 THE RATIONALE FOR THE PROJECT

Financial pressures on the NHS are requiring increasingly innovative approaches. Adopting a patient to patient hybridisation approach halves the cost of performing the test [1].

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 [7].

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 sensitivity 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.

A false negative misses a chance of diagnosis, extending the patient's 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 [8].

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).

An array run starts with the creation of a worksheet. This worksheet defines which patients will be hybridisation partners. Hybridisation partners must be phenotype mismatched, a process performed by one clinical scientist and checked by a second. For a run of 96 samples worksheet creation and checking can take three hours. 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 automatable, further reducing the cost of processing an array.

More importantly further reducing the chance of missing an CNV improves the service the patient receives, ending, or preventing a diagnostic odyssey.

1.6 PROJECT AIMS

The aim of this project is investigate a tool which is run during data processing and acts as a screen to detect CNV before the array present in both hybridisation partners.

1.7 ARRAY DESIGN

The arrays, reagents, equipment and software used to process and analyse are manufactured by Agilent Technologies [9].

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

The array was custom designed using the online probe catalogue eArray [10]. The array design consists of 46554 probes. 3886 are control probes and 42658 are targeted of which 29 are duplicated leaving 42629 unique probes.

A request for a tool similar to the aims of this project was sent to and declined by Agilent.

1.7.1 *Feature Extraction*

After the arrays have been processed in the laboratory and scanned (Figure 2) 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 intensity is processed, including a normalisation step across all

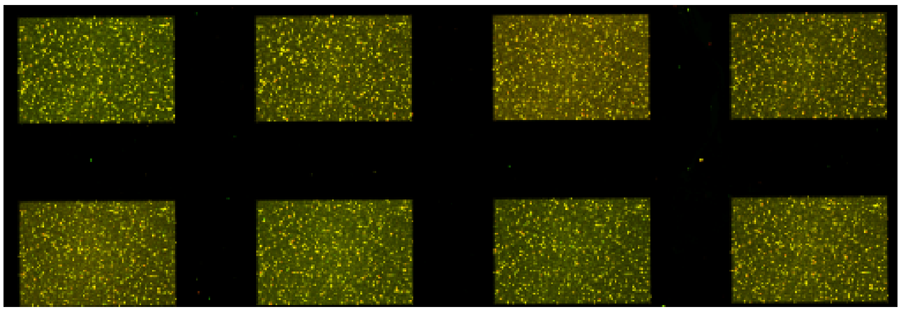


Figure 2: A scanned array slide containing 8 arrays

probes producing a processed signal intensity normalised to 1000 Arbitrary Fluorescence Units (AFU).

At each probe a log ratio is calculated from the two signal intensities (equation 1)

$$\text{Log10}\,\frac{\text{Cy5 signal intensity}}{\text{Cy3 signal intensity}} \tag{1}$$

1.7.2 *Feature extraction file*

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.

Each feature extraction file consists of three sections, the parameters, stats and features.

1.7.2.1 *Parameters*

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

1.7.2.2 *Stats*

The stats section 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.

1.7.2.3 *Features*

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.

1.8 USE OF ALGORITHMS TO DETECT COPY NUMBER VARIATION

If all goals of this project are met the tool produced will have much the same role as the aberration detection algorithms, taking raw signal intensities and identifying CNV.

There are a number of algorithms which call CNV from CGH data. These algorithms are recursive binary segmentation methods which break chromosomes into segments of equal copy number. Examples include Z scores, Circular Binary Segmentation (CBS), Aberration Detection Method (ADM2), Nexus and Hidden Markov Model (HMM).

1.8.1 *Z score algorithm*

A Z score is a statistical measure of deviation from the mean for a normally distributed population. The Z score algorithm assigns each probe a Z score based on the log ratio. Any probes outside a user defined cut off (R) are classified as an outlier, or significantly away from mean.

The number of probes classified as above (R) and below (R′) this threshold and total number of measurements (N) are recorded across 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 [11].

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

These algorithms are designed by Agilent[11]. These algorithms do not used fixed windows but segment the genome into intervals of equal copy number using log ratio scores from adjacent probes to best define interval breakpoints.

1.8.2.1 *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.

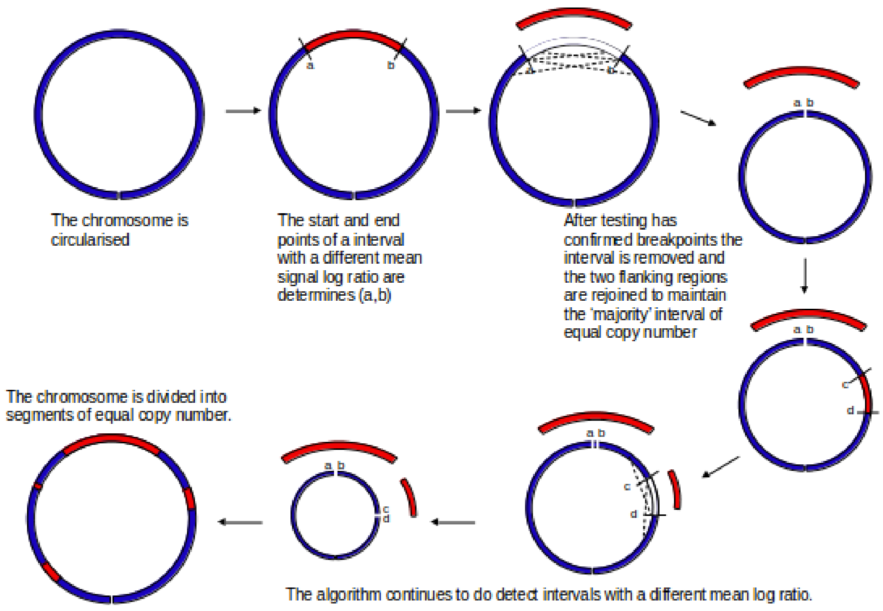
The interval with the highest score is selected and the same process is performed on this segment to further define breakpoints. This is repeated for all intervals with an S(I) above the threshold.

1.8.2.2 *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).

1.8.3 *Circular binary segmentation (CBS)*

CBS uses log ratios from adjacent probes 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 [12].



The chromosome is circularised
a
b
The start and end points of a interval with a different mean signal log ratio are determines (a,b)
After testing has confirmed breakpoints the interval is removed and the two flanking regions are rejoined to maintain the 'majority' interval of equal copy number
a b
c
d
The chromosome is divided into segments of equal copy number.
The algorithm continues to do detect intervals with a different mean log ratio.

Figure 3: Circular binary segmentation algorithm

Each chromosome is made into a circle (Figure 3). This allows for two break points to be identified, increasing 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.

1.8.4 *Hidden Markov Model*

The hidden Markov model (HMM) [11] uses observations (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.

1.8.5 *Algorithm Performance*

Each algorithm performs differently depending on the signal:noise ratio and the size of the aberration[13].

Circular binary segmentation (CBS) is rated as one of the most consistent performers albeit one of the most computationally demanding [14, 13]. Some algorithms' performance varies greatly depending on user defined parameters which may be difficult to establish [14].

The algorithms described above are global segmentation methods which compare segments to the whole genome. Recently local segmentation methods have been described which look for CNV using high resolution data sets [15]. These algorithms have higher power than global algorithms but are less robust so an approach utilising both types of algorithm may produce the best performance [3].

1.9 DATA STORAGE AND MANIPULATION

The tool requires data to be stored, curated and accessed. A database is an efficient tool for this role.

1.9.1 *Relational database*

A relational database is a collection 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.

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 interface (GUI) or 'front end' enables user interaction with the data, commonly seen in Microsoft Access or an internet browser.

Database design and data manipulation/retrieval utilises the Structured Query Language (SQL).

1.9.2 *Non-relational database*

MongoDB [16] 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.

1.9.3 *Interacting with databases*

Python is an object orientated programming language. Python programs can be written to perform a wide range of functions, such as parsing text files, performing complex mathematical and statistical calculations utilising additional modules or packages such as NumPy [17] and SciPy [18].

Connectors are available (including MySQLdb [19], SQLAlchemy [20], PyODBC [21] and PyMongo [22]) enabling Python to interrogate and interact with databases.

1.9.4 *R statistical software package*

The R statistical software package [23] has a repository of packages (Bioconductor [24]) which includes packages which implement CNV algorithms such as CBS within DNAcopy [25].

2 METHODS

2.1 Shared Imbalance Detection Tool (SID)

The Shared Imbalance Detection (SID) tool consists of two components, a database and a python script. The database holds information about the array design, values used during the analysis and tables which are filled during analysis. The python script parses feature extraction files, importing information to the database and performing the analysis with a combination of SQL commands and python functions.

2.2 Requirements gathering

To ensure the tool is fit for purpose requirements were agreed with key stakeholders in the Array and Bioinformatics teams within the Viapath Genetic Laboratories:

2.2.1 *Functional requirements*

- Use the signal intensities from the feature extraction file to identify CNV independently of the hybridisation partner
- An region of CNV is defined as a minimum of three consecutive abnormal probes which is ended by three consecutive normal 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.

2.2.2 *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, ensuring 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.
- Python code must conform with the departmental Python style guide.
- Code must be version controlled using Git.
- Processing should be timely and able to be performed on existing desktop computers.
- The feature extraction files are required for analysis by Genetic Scientists so must not be modified or moved from the current location.
- Compatible with the trust operating system (Windows XP, with an upgrade to Windows 10 planned before 2018).
- Able to be run from multiple computers within the department by multiple users.
- Analysis must be reproducible.
- The database must be in third normalised form to ensure data integrity.
- Standard Operating Protocol (SOP) and validation documentation in line with ISO accreditation requirements.
- Validated in line with departmental and ISO quality procedures.

2.3 PROJECT MANAGEMENT

- Use an Agile development methodology to regularly test and refine the project (Figure 4).

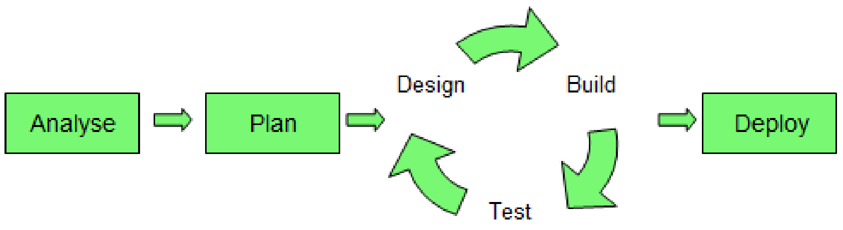


Figure 4: The agile software design methodology promotes regular testing to ensure the design is fit for purpose

- Use smartsheet (`https://www.smartsheet.com`) 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.

2.4 DATABASE INFRASTRUCTURE

2.4.1 *SQL Server*

Access to the MS-SQL hosted on the department server was not available during development. Installation of an MS-SQL server on the development system was also not possible at the start of the project so development was performed using a MySQL server (v5.6)[26] running locally on a development PC running Windows 10.

2.4.2 *Migration from development server (MYSQL) to production server (MSSQL)*

The database was later transferred to a MS-SQL server (SQL-Express 2014 [27]) on the development using Microsoft SQL Server Migration Assistant (v5.3) for MySQL [28]

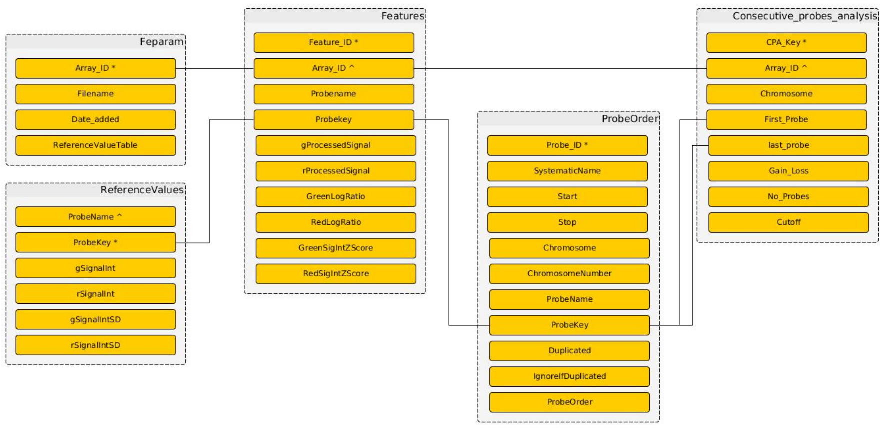


Figure 5: The SID database schema. Five tables are required. d̂enotes an indexed field, * denotes a primary key. Lines denotte how tables are joined.

The MySQLdb python connector was replaced by PyODBC MSSQL connector [21] and tested to ensure no loss of function.

2.4.3 *HeidiSQL client*

Database development was performed using the HeidiSQL client (v9.1)[29].

2.5 SID - DATABASE DESIGN

To minimise processing time fields from the feature extraction file imported to the database was kept to a minimum. The database is designed in third normalised form, storing numeric keys to prevent anomalies and to maximise the performance of joins. Array wide values are stored a single time in the feparams table, not for each feature.

Five tables are required (Figure 5).The Feparam table holds the filename. Upon insertion an auto incrementing number is created as the primary key which is used as the unique identifier for that array (Array_ID). The Feparam table also records the reference table used and an automatic time stamp.

The array design is stored in the Probeorder table. Each probe has a numerical key (probekey) and it's order in genomic position (probeorder). Probeorder only contains non-control probes and marks replicated probes.

The ReferenceValues table holds the average signal intensity and the standard deviation from the normal control population (see 2.8.2).

The Features table holds the analysis for each probe. Of the 42 fields only the probename and processed signals are imported. 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.

2.5.1 *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. Any columns used to join table are also indexed to improve performance.

2.6 IMPORTING AND PROCESSING FEATURE EXTRACTION FILES

2.6.1 *Python*

Feature extraction files were parsed using Python (v2.7)[30]. The Anaconda Python distribution (v2.3.0) [31] was selected to make use of the preinstalled packages such as NumPy, SciPy and math packages. The package manager Conda was used to install the MySQLdb package (mysql-python v1.2.5)[19] to access the MySQL database from python.

The Eclipse IDE (Luna v 4.4.1) [32] was used with the PyDev package (v 3.9.0) [33] to develop and run Python scripts.

The Pylint Python package (v 1.4.2) [34] and autopep8 were used within Eclipse to apply the local Python style guide to the code.

2.6.2 *Version control*

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

2.7 SID - PROCESSING FEATURE EXTRACTION FILES

Figure 6 shows an overview of the SID python script.

2.8 DETECTION OF CNV

2.8.1 *Choice of algorithm (Z score)*

A Z score analysis method was selected for the analysis as the characterisation of breakpoints is not as critical as in routine analysis and this algorithm is very simple to implement, maintain and troubleshoot, without the need for any additional software eg R.

A Z score (equation 2) 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.

$$Z = \frac{Signal\ Intensity - Mean}{SD} \tag{2}$$

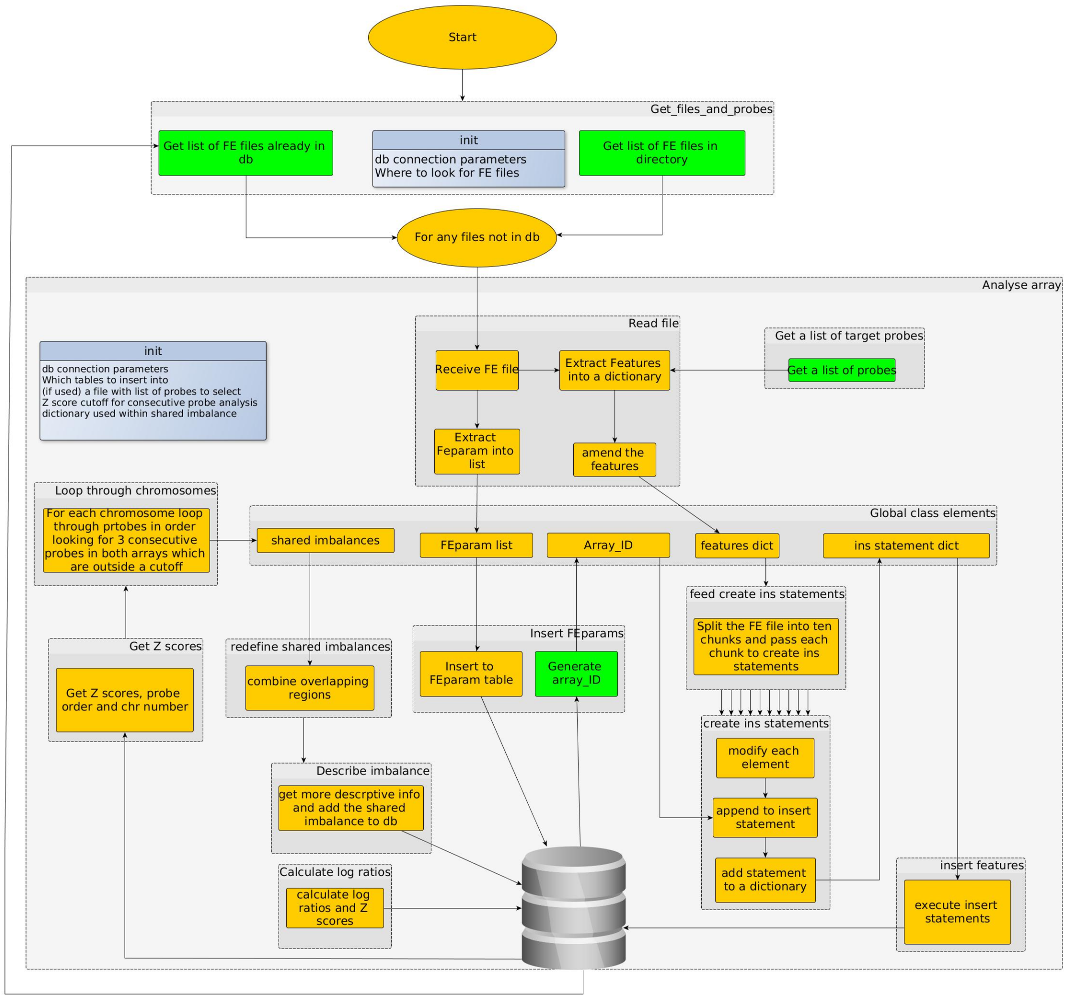

Start
Get_files_and_probes
Get list of FE files already in db
init
db connection parameters
Where to look for FE files
Get list of FE files in directory
For any files not in db
Analyse array
Read file
Receive FE file
Extract Features into a dictionary
Extract Feparam into list
amend the features
Get a list of target probes
Get a list of probes
init
db connection parameters
Which tables to insert into
(if used) a file with list of probes to select
Z score cutoff for consecutive probe analysis
dictionary used within shared imbalance
Loop through chromosomes
For each chromosome loop through prtobes in order looking for 3 consecutive probes in both arrays which are outside a cutoff
Global class elements
shared imbalances
FEparam list
Array_ID
features dict
ins statement dict
feed create ins statements
Split the FE file into ten chunks and pass each chunk to create ins statements
Get Z scores
Get Z scores, probe order and chr number
redefine shared imbalances
combine overlapping regions
Insert FEparams
Insert to FEparam table
Generate array_ID
create ins statements
modify each element
append to insert statement
add statement to a dictionary
Describe imbalance
get more descrptive info and add the shared imbalance to db
Calculate log ratios
calculate log ratios and Z scores
insert features
execute insert statements

Figure 6: A visual representation of the SID python script. The Get_files_and_probes class compares the list of feature extraction files within a directory to the feparam table.
Each array yet to be analysed is passed to Analyse_array class. The read file module extracts the relevant information into dictionaries. The insert FEparams module enters the filename to the database and captures the primary key assigned (the array_ID). The feature section of the feature extraction file is broken into 10 and inserted to the table with the array_ID.
The calculate log ratios module applies an update statement to calculate Z scores and log ratios. The get Z scores module takes these Z scores and identifies shared trios of abnormal probes, combines these into larger calls which are also stored in the database. Functions coloured green read from the database, yellow process data and blue are __init__ modules which initialise variables for the class

2.8.2 *Creating a normal reference range*

100 array cases with no reported CNV were selected to create 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 calculated the average signal intensity and standard deviation for each dye at each probe. These values are held in the ReferenceValues table which is versioned using the date as the reference may require periodically recalculating.

The ReferenceValues table used is specified within the SID Python script and recorded in the Feparams table, allowing analysis to be repeated after the reference range has been recalculated.

2.8.2.1 *Normal distribution of reference range*

Z scores require normally distributed reference ranges. However some probes will not be normally distributed for example if they are within a common CNV.

A histogram of the signal intensities of all probes from 30 arrays (Figure 7) shows a slightly positively skewed distribution. This it to be expected as the signal intensities cannot be negative and gains in copy number is seen more commonly than losses [4].

2.8.3 *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 how many standard deviations from the mean the signal intensity of the probe is. A threshold is required to define a probe as abnormal.

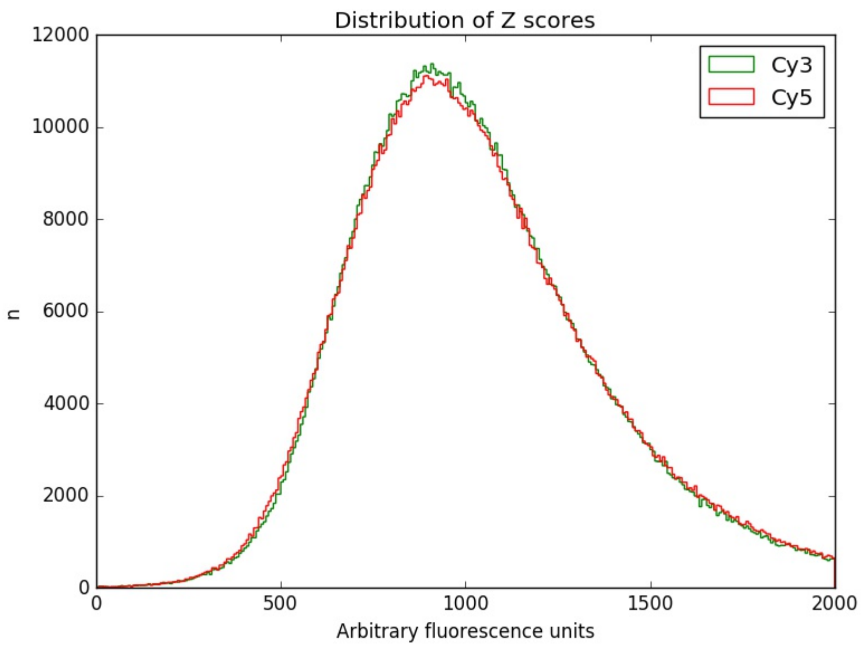


Figure 7: The distribution of signal intensities from all probes from 30 arrays is slightly positively skewed

Z scores are two tailed so probes with a Z score above the threshold are defined as gains and Z scores below the negative threshold are defined as loss (Figure 8).

2.8.4 *Duplicated probes*

For the 29 duplicated probes an average of the signal intensities is taken.

2.8.5 *Determining if a region is normal or abnormal*

To combine abnormal probes into segments a slightly different approach was taken to the Z score algorithm described by Agilent [11].

Aberrations below three probes are not reported so a sliding window of three probes was used, identifying shared CNV of three consecutive probes (Figure 9). The window moved along the entire chromosome creating overlapping tiles 10.

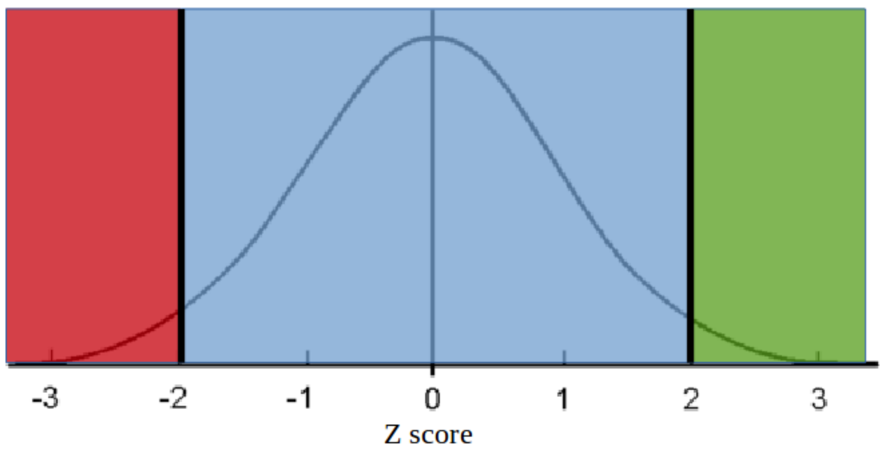


Figure 8: Cutoffs can be aplied to define probes as abnormal or normal. A more extreme threshold will define fewer probes as abnormal.

Trios of shared CNV are then combined in line with routine analysis, combining regions unless separated by 3 or more normal probes (Figure 10).

2.9 creation of true positive feature extraction files

SID underwent three stages of development.

2.9.1 *Training cases*

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

Probes within the reported region were identified and a new FE file was created where the Cy5 signal intensity was replaced by the Cy3 signal intensity. This was repeated replacing the Cy3 signal intensity with the Cy5 signal intensity.

The script created one true positive, with signal intensities representing an aberration in both hybridisation partners and

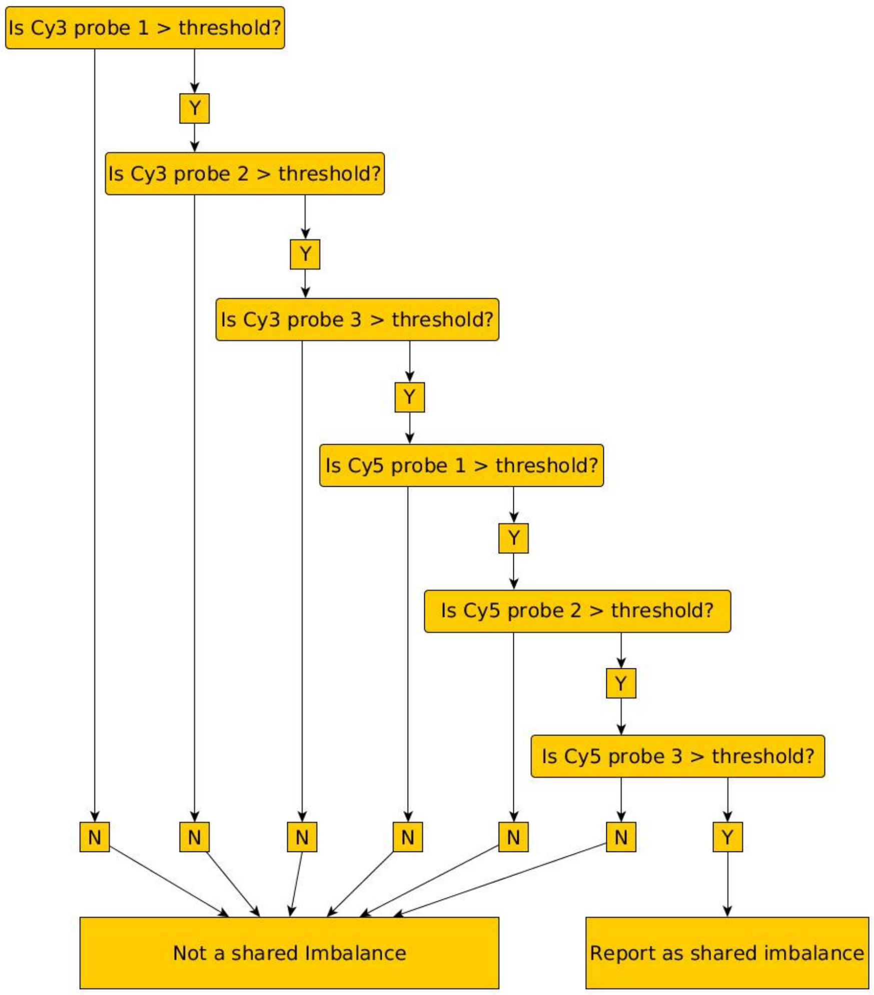

Is Cy3 probe 1 > threshold?
Y
Is Cy3 probe 2 > threshold?
Y
Is Cy3 probe 3 > threshold?
Y
Is Cy5 probe 1 > threshold?
Y
Is Cy5 probe 2 > threshold?
Y
Is Cy5 probe 3 > threshold?
N
N
N
N
N
N
Y
Not a shared Imbalance
Report as shared imbalance

Figure 9: Consecutive probe analysis algorithm. To determine if the sliding window of three probes is abnormal in both samples the Z score of each probe is tested in succession. The threshold is defined in SID python script.
This process is repeated assessing if the Z score is less than the negative threshold.

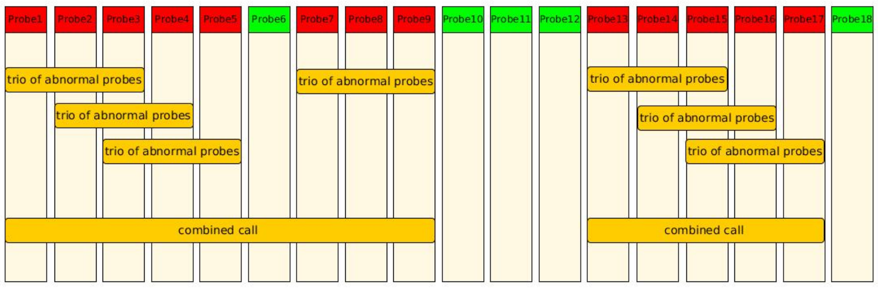


Figure 10: Trios of probes are combined into a single call in line with the analysis strategy. Abnormal probes are represented in red and normal probes represented in green. Trios of shared imbalances are shown. Overlapping and neighbouring calls are combined if they are separated by less than three normal probes (eg probes 10-12).

one true negative file where both partners have normal signal intensities.

This training set was used to investigate and define the parameters required to call a probe as abnormal.

2.9.2 *Test cases (EvE)*

The test set was used to ensure the parameters were not overfitted to the training set. An updated strategy was applied to create the test set. Two samples which had exactly the same reported CNV 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.

EvE, a Python script used for the post-hoc hybridisation of arrays in Preimplantation Genetic Diagnosis (PGD) cases, parses two feature extraction files, extracting all measurements for both samples on both arrays, and creates a new feature extraction file where the Cy3 measurements are populated from the Cy3 sample from the first file and the Cy5 measurements from the Cy5 sample from the second file.

Table 1: Cases with one of the 9 most common CNV were used to create the test set

Aberration	Copies	Number of Probes	Number of arrays
15:22765627-2308509	1	11	9
16:29673953-30198600	1	34	9
16:29673953-30198600	3	34	10
22:18896971-211440514	1	750	8
15:30933080-32514980	1	30	6
16:21806298-22407931	1	15	10
7:72700413-74142327	1	43	6
1:145413387-145747269	3	21	5
6:26440746-26463502	1	3	6

The test cases were reported to have one of the most commonly reported CNVs, representing a range of sizes and loss and gains 1.

2.9.3 *Prospective trial*

Finally a third set of routine array files were used as a prospective trial. 121 files used to create the test set were uploaded without any modification.

2.10 DEFINITION OF TRUE POSITIVE AND FALSE POSITIVE CALLS

A true positive call is defined as any call overlapping with the reported CNV. There may be multiple calls within this region.

A false positive call is any call which does not share a single probe with the reported CNV.

2.10.1 *Unknown true positive calls*

In the testing sets the true positive calls are defined as those which have been reported as clinically significant based on the referral reason however on average an array will have 3-4 regions of CNV [**?**]. These calls can be false positive calls made by the algorithm or real benign calls in regions of common copy number variation.

Therefore the number of false positive calls will be inflated and the true specificity unknown.

3 RESULTS

3.1 TRAINING CASES

3.1.1 *Z score threshold*

Investigating the Z score of probes within regions known to be abnormal suggests a range of Z score thresholds to define probes as abnormal to investigate (Figure 11).

However, reported regions may contain normal probes and whilst sensitivity is favoured over specificity a balance must be found to prevent many false positive calls.

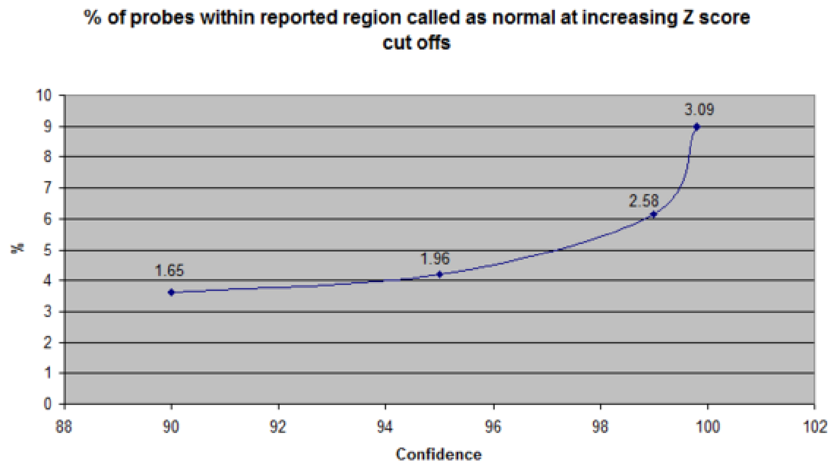


Figure 11: The percentage of probes called as normal within known abnormal regions increases with the Z score threshold. Abnormal regions can include normal probes so even at a Z score threshold of 1.65 over 3% of probes are called as normal. The Z score is shown above each point

Table 2: A very high number of false positive calls was seen in the 86 training cases using a range of Z score thresholds.

	Z Score						
	2.374	3	3.5	3.55	3.75	4	4.25
% True Positives (n)	92 (79)	92 (79)	92 (79)	92 (79)	92 (79)	92 (79)	88 (76)
% Arrays with false positives (n)	67 (58)	45 (39)	37 (32)	36 (31)	33 (28)	30 (26)	30 (26)
Number of false positive calls	10228	7681	6177	6044	5538	4944	4394
Average number of false positive calls per array (max)	192.98 (2128)	225.91 (1733)	228.78 (1468)	232.46 (1443)	240.78 (1351)	235.43 (1230)	209.24 (1108)

3.1.2 *Removal of cases with high false positive calls*

A range of Z score thresholds were applied, starting at $\pm$ 2.374, defining abnormal probes as the upper 1% and lower 1% of the reference range, and increasing up to 4.25 (Table 2).

At the lowest Z score threshold (2.374) the false negative rate was 8% with 67% of cases containing a false positive call.

The same Z score threshold resulted in 10228 false positive calls, of which 94% came from just five arrays (range 1612-2128). The average number of false positive calls in the other 81 arrays was 10.8 (max 104) (Table 3).

Visual inspection of the array traces and the raw data from these 5 arrays gave no indication why so many false positives calls were made or how this could be mediated/predicted. Therefore these five arrays were considered a different population and excluded from further analysis.

3.1.3 *Comparison of true positive and false positive calls*

The remaining 81 arrays show that at least one call was detected within the abnormal region in 94% of cases with a Z score up to 4 (Table 3).
Using a Z score of 2.374 nearly 2 in 3 arrays had a false positive call. A Z score threshold of 4 reduced the frequency of false positive calls to 1 in 4 arrays, with an average 1.66 and maximum 6 false positive calls per array.

Table 3: The calls made in 81 training cases using a range of Z score thresholds. The true positive call was missed in at least 5 cases across all thresholds, with a false positive call made in at least 1 in 4 cases. The number of false positive calls, and the average number of calls made decreased as the Z score threshold increased.

	Z Score						
	2.374	3	3.5	3.55	3.75	4	4.25
% True Positives (n)	94 (76)	94 (76)	94 (76)	94 (76)	94 (76)	94 (76)	90 (73)
% Arrays with false positives (n)	65 (53)	42 (34)	33 (27)	32 (26)	28 (23)	26 (21)	26 (21)
Number of false positive calls	571	114	50	48	41	35	31
Average number of false positive calls per array (max)	10.8 (104)	3.35 (20)	1.85 (7)	1.84 (7)	1.78 (7)	1.66 (6)	1.48 (5)

3.1.3.1 *Number of probes within a call*

The length of false positive calls are shorter than true positive calls. Increasing the minimum number of probes within a call would remove these false positive calls but would also remove true positive calls and fail to meet the functional requirement that a call is a minimum of 3 consecutive probes (Figure 12).

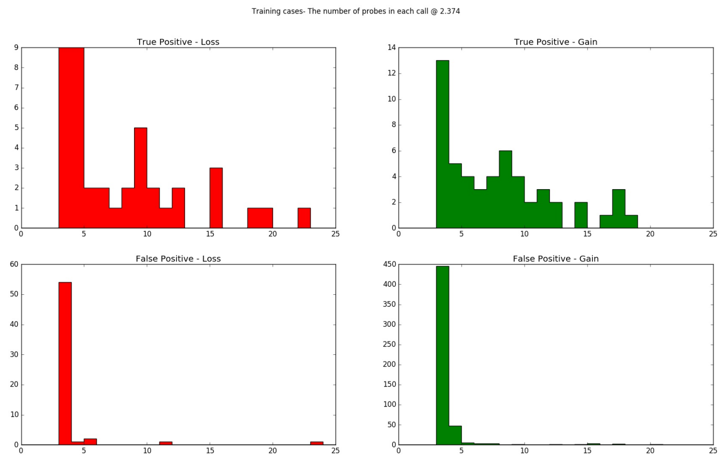


Figure 12: Training cases: False positive calls are smaller than true positive calls.

3.1.3.2 *Difference between true and false positive Z scores using a threshold of 2.374*

Alternatively increasing the Z score threshold to above the lowest scoring probe in abnormal regions may prevent the region being called 9.

Examining the minimum (Figure 13) and average (Figure 14) Z scores for each true and false positive calls show that the two populations are not disparate therefore increasing the Z score threshold may also remove true positives.

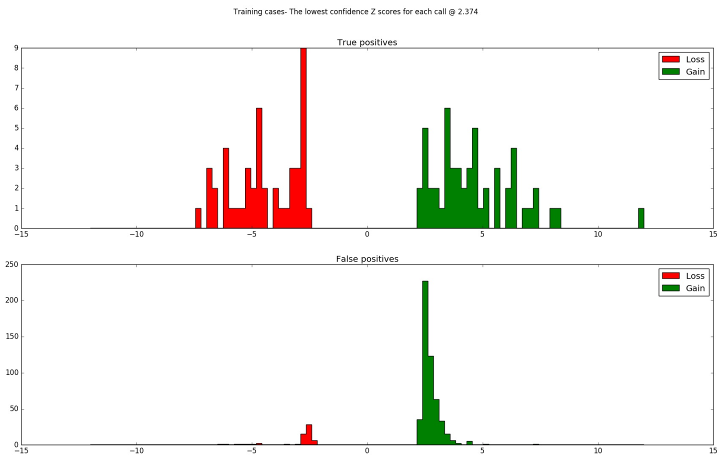


Figure 13: Training cases: The lowest confidence Z score within each call at a threshold of 2.374. There is an overlap between the confidence of the lowest probe within true and false positive calls

However the increased number of probes within a true positive region (Figure 12) and the higher average Z score (Figure 14) may mean the true positive regions remain.

3.1.3.3 *Difference between true and false positive Z scores using a threshold of 3.55*

Increasing the Z score threshold to 3.55 showed that the lowest confidence Z scores in remaining false positive calls were similar to that of true positives (Figure 15).

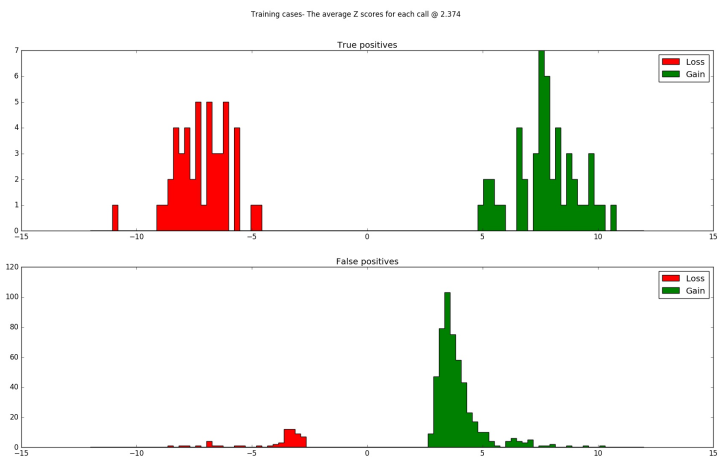


Figure 14: Training cases: The average Z score of probes within calls at a threshold of 2.374. The average Z score of true positive calls are further from the mean than the average Z score of false positive calls.

Similarly the distribution of average Z scores were very similar between true and false positives calls (Figure 16).

3.1.3.4 *Suitability of reference range*

There were no recurring false positives calls to suggest the reference ranges were unsuitable.

3.2 TEST CASES

The training cases are not truly representative of an array where both hybridisation partners have the same CNV. Cy3 and Cy5 dyes are incorporated with varying efficacy so the reference value will vary. When calculating the Z score for these cases comparing the signal intensity of one dye to the reference value of the other dye may introduce error.

The same thresholds were applied to 69 test cases 1.

There were no recurring false positive calls.

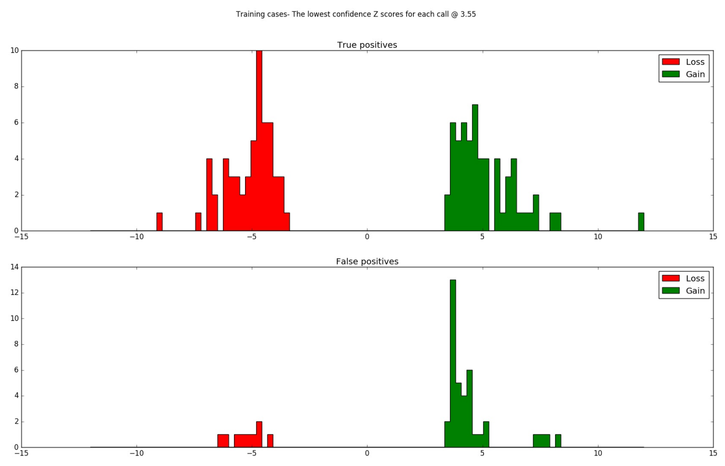


Figure 15: Training cases: At a threshold of 3.55 the lowest confidence probes within true and false calls are very similar

Table 4: Test cases: A call was made in the expected CNV region in all cases with a threshold below 4. There were no false positive calls made with a threshold above 3.55

	Z score						
	2.375	3	3.5	3.55	3.75	4	4.25
% True Positive arrays (n)	100 (69)	100 (69)	100 (69)	100 (69)	100 (69)	100 (69)	97 (67)
% Arrays with false positives (n)	13 (9)	8.7 (6)	1.4 (1)	0	0	0	0
Number of false positive calls (max)	48 (28)	9 (3)	1 (1)	0	0	0	0

3.2.1 *A greater sensitivity was seen using more representative feature extraction files*

There was 100% sensitivity using a Z score below 4 and there were no false positive calls using a Z score above 3.55 1.

3.3 PROSPECTIVE CASES

121 routine arrays were processed. The expected results are unknown but the number of calls is indicative of the number of calls required to be assessed during analysis.

Using the Z score threshold of 2.374 there were a significant number of calls, with more than 1 in 3 cases having calls to investigate. A threshold of 3.55 resulted in a call in at least 1 in 9 arrays (Table 5).

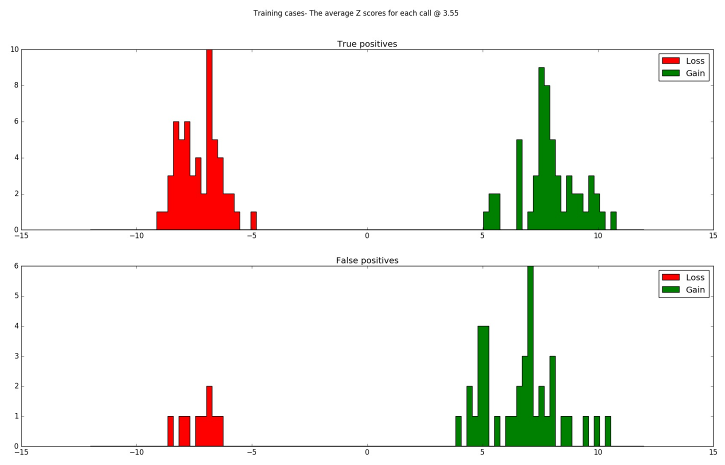


Figure 16: Training cases: The average Z score for probes within true and false positive calls with a threshold of 3.55 are very similar.

Table 5: My caption

Z score	% of arrays with a call (n)	Total number of calls (max calls in a singe array)
2.374	37 (45)	1760 (555)
3	16 (19)	591 (219)
3.55	12 (14)	246 (119)
3.75	12(14)	186 (98)
4	10 (12)	125 (66)
4.25	7 (9)	90 (50)

It is impossible to know if any of these are actually true positives.

The average size of each call is just over three probes (Table 6)

3.4 PERFORMANCE OF SID

When uploading a run of 96 arrays, initially each array is processed in approximately two minutes. However, due to the increasingly large features table, the time taken to process an array can increase to around 15 minutes.

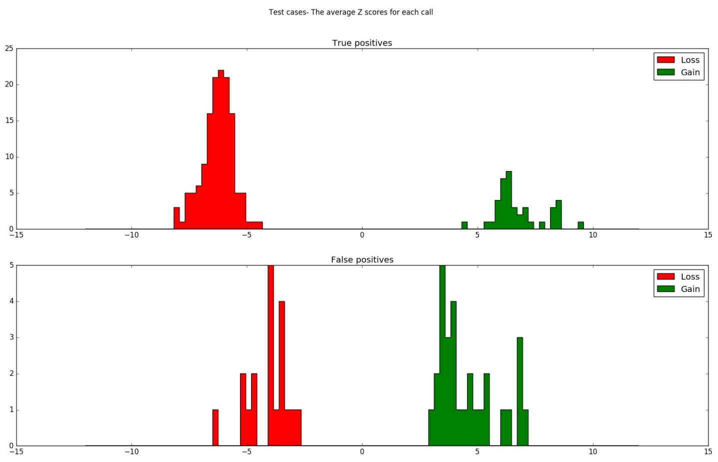


Figure 17: Test cases: The average Z score in true positive calls made at a threshold of 2.374 are more significant in true positive calls than false positive calls however there is overlap between the two groups

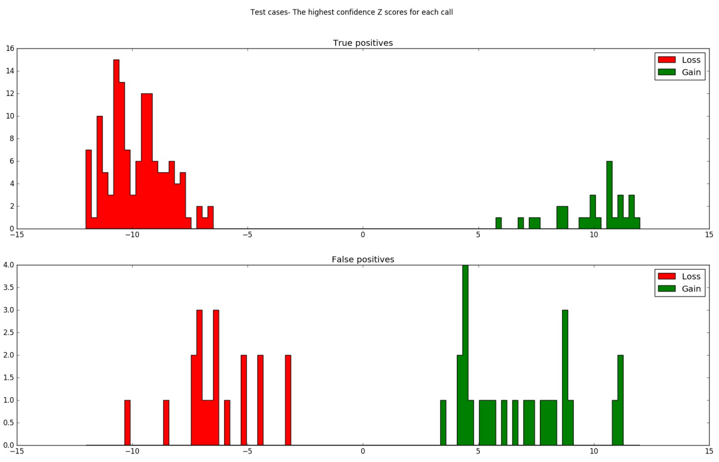


Figure 18: Test cases: At a Z score threshold of 2.374 there is an overlap between the highest confidence probe within true and false positive calls.

Table 6: Prospective cases: The average and maximum number of probes within a call during analysis of 121 prospective arrays.

Z score	Average probes in call	Maximum probes in call	Number of calls
2.374	3.24	8	1760
3	3.19	7	591
3.55	3.14	6	246
3.75	3.13	6	180
4	3.1	5	125
4.25	3.09	5	90

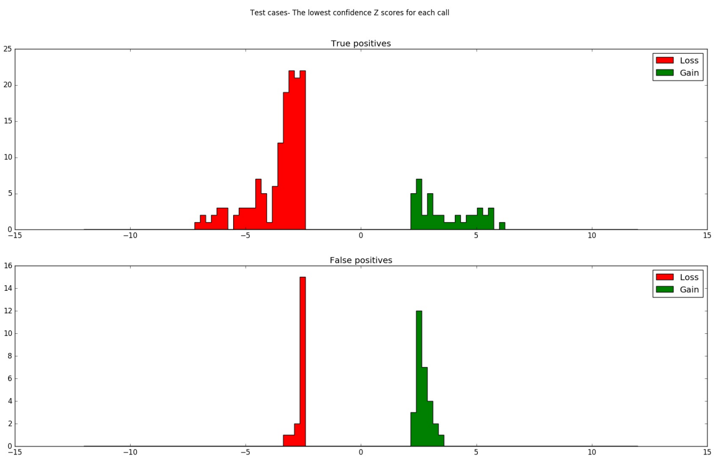


Figure 19: Test cases: There is overlap between the lowest confidence probe within true and false positive calls made at a threshold of 2.374

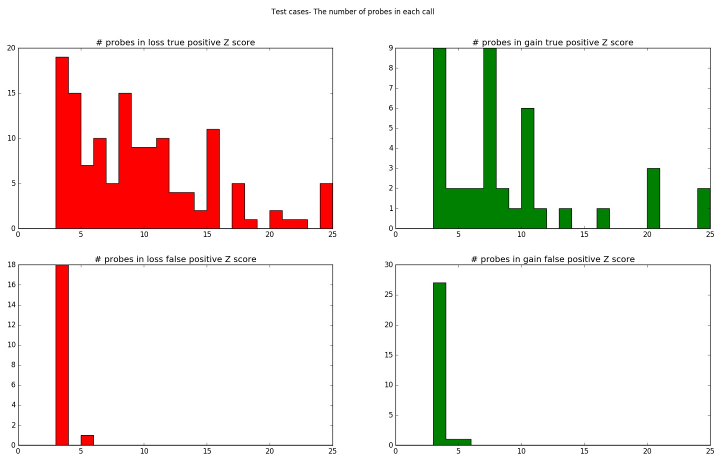


Figure 20: Test cases: At a threshold 2.374 true positive calls contain more probes than false positive calls.

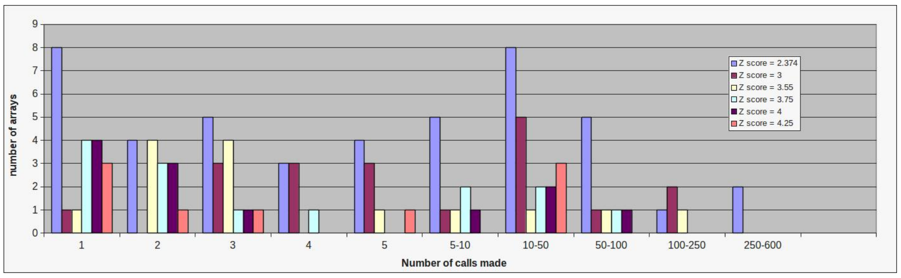


Figure 21: Prospective cases: The number of calls per array across a range of threshold in 121 prospective cases

4 DISCUSSION

4.1 SUITABILITY OF SID

The Shared Imbalance Detection (SID)) tool, written in Python and utilising a MS-SQL database capable of taking a feature extraction file and by applying a Z score algorithm can detect regions of shared CNV from manually created true positive cases with a sensitivity of 100% in a timely fashion.

4.2 Z SCORE THRESHOLD TO DEFINE A PROBE AS ABNORMAL

Analysis of the test cases suggest that using a Z score of between 3.55 and 4 to define a probe as abnormal would detect shared copy number variation without any false positive calls.

Conservatively applying the lower threshold to prospective arrays resulted in calls in 1 in 9 cases.

4.3 COMBINING OVERLAPPING TRIOS OF ABNORMAL PROBES

Trios of shared abnormal probes are combined in line with existing analysis protocols.

4.4 BENIGN CNVS

It is expected that hybridisation partners would share one or more of the many known, common benign CNV regions. SID would correctly identify these regions as abnormal, however the above analysis would classify these calls as false positives.

It may be worth excluding these regions from future specificity and sensitivity calculations

Should SID be adopted by the clinical service stakeholder should be consulted to determine if calls within known benign CNV regions should be filtered out, or if the experience of clinical scientists would make analysis of common calls trivial.

4.5 CALLS SMALLER THAN 5 CALLS

The vast majority of false positive calls in the test set and prospective arrays contain less than 5 probes (Table 6 and Figure 20).

The average signal intensities of probes within a true positive call is higher than that of a false positive call 17 so a secondary analysis, could be applied to further reduce false positive calls.

4.6 DIFFERENT THRESHOLDS FOR GAIN/LOSS

The proportional signal intensity change between diploid and triploid and diploid and monosomy is the same (±50%). Therefore the Z score would be the same for monosomy and trisomy. Similar numbers of false positive gains and losses suggest using the same threshold for losses and gains does not introduce bias however should further work identify a lack of sensitivity for either gain or loss separate thresholds could be applied.

4.7 ALTERNATIVE APPROACHES TO CONSECUTIVE PROBES ANALYSIS

4.7.1 *Larger sliding windows*

The Z score algorithm implemented differs to that suggested by Agilent [11]. Agilent suggest a sliding window and the proportion of probes outside a threshold used to define a region as abnormal.

In order to meet the functional specification of detection of three consecutive probes the sliding window must be very small, and the proportion of abnormal probes within a region would be very high. This may however result in a call which does not meet the specification (Figure 22a)

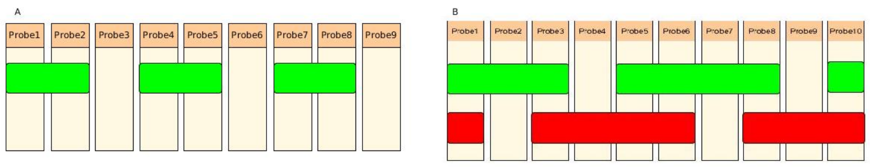


Figure 22: Classifying overlapping shared CNV. a. Using a sliding window approach and measuring the proportion of probes outside a threshold may result in calls which do not meet the functional specification of three consecutive probes.
b. The existing algorithm may miss clearly overlapping CNV where three consecutive abnormal probes are not shared

This approach would however prevent aberrations such as those in Figure 22b being missed.

The consecutive probe approach is designed to match the routine analysis approach however a sliding window approach may be worth investigating. Small overlapping windows can be applied and combined similar to the existing tiling strategy.

4.7.2 *Sliding window - Regions of interest*

A variation of the sliding window approach was investigated, using defined regions of interest (every CNV which has been reported more than twice) as a window. The same Z score analysis was performed and the proportion of abnormal probes within the region used to determine if a region is abnormal or not.

Despite this approach being targeted to known clinically relevant CNV and the potential to expand the list of regions this approach is more limited than the consecutive probe approach so was not pursued.

4.8 FURTHER WORK

4.8.1 *Sex Chromosomes*

Further work is required to include sex chromosomes in the analysis. The gender of the sample affects the signal intensities of probes on the sex chromosomes. To identify CNV on the sex chromosomes a gender specific reference range is required. The average signal intensities across non pseudoautosomal regions of the X and Y chromosomes could determine gender and therefore which reference range should be applied.

4.8.2 *Sample Type*

This project has only assessed postnatal blood samples. The array service also processes prenatal, solid tissue and PGD samples, which have variable resolution and performance due to the quality of the sample.

Differing analysis strategies and Z score thresholds would be required for each sample type in line with existing analysis approaches in addition to a method for specifying the sample type of each array.

4.8.3 *What happens when shared CNV is detected?*

Once shared CNV is detected further investigations is required. The level of investigation required would dictate the acceptable levels of specificity of SID. If a thorough investigation is required the time and cost incurred would negate any savings brought by replacing phenotype mismatching with SID.

The signal intensities can be examined within the analysis software which would confirm or deny the presence of CNV. This practise is in line with current analysis methods when identifying if an aberration is a duplication in one hybridisation partner or a deletion in the other. However having >100 calls on a single array would make this approach unfeasible.

Another potential approach is EvE, a tool which allows post hoc hybridisation of samples. Each sample could be independently compared against another sample, such as a reference control, to see if CNV is present.

EvE takes about one minute to run and the analysis could be performed in less than 10 minutes. However should this be required routinely any time savings would be diminished. EvE has been validated for use with the PGD service but has not yet been validated for use with postnatal blood samples.

An extreme approach is to re-run the samples against different hybridisation partners. If this is the case the specificity must be well understood to avoid a large number of samples being repeated, removing any cost savings.

The approach taken would require a thorough consultation with clinical scientists and bioinformaticians,

4.8.4 *Exporting results to LIMS*

The SID database will be a separate database to the departmental LIMS database, hosted on the same MS-SQL server.

A function could therefore be added to the end of SID to connect to the LIMS system and export the results directly to the LIMS system. The filename can be used to link to the two patient records using an existing query.

The LIMS system would require an extra table to hold the shared CNV. Clinical scientists should be consulted to determine where and how this information should be displayed.

Confirmation that analysis has completed successfully should be included for samples with no shared CNV.

4.8.5 *Initiation of SID*

Array runs occur at scheduled times during the week. SID scans a directory and compares the files in the directory to those already processed (stored in the feparams table) which would enable the script to be scheduled to run at specific times.

Alternatively the script could be set off manually by the array technician once the feature extraction process has completed. An SOP and training would be required.

4.8.6 *Migration of database to the server*

The database has not yet been migrated to the SQL server on the server. Interacting across a network into a remote server may increase the time taken to process each sample and the stability of this connection tested.

If run overnight an increase in time may be tolerable however a process to identify and remedy interrupted or failed processing will be required.

4.9 LONG TERM MANAGEMENT OF SID

4.9.1 *Unit test*

After any changes are made to the script or database a unit test, utilising a truth set from a subset of the test set, should be performed to ensure no loss of function.

4.9.2 *Recalculation of the reference range*

The measured signal intensity can be affected following any maintenance or change to the microarray scanner. Following any maintenance a new reference range should be calculated, possibly using the first array run following maintenance.

This reference range should be created in a new table and the SID python script updated to apply the new reference and record the reference range applied in the feparams table.

4.9.3 *Truncating the database to maintain performance*

Each array run consists of 48 or 96 arrays. As the run is processed and the Features table grows in size the time taken to process each array increases up to 7 fold.

Approaches to control SQL transactions failed to improve this situation however should this processing time become prohibitive alternative insert approaches can be considered, such as creating and inserting the features from a comma separated values (CSV) file. This approach proved slower when uploading single arrays.

Truncation of the features table, following analysis of each array or array run would maintain performance.

Alternatively a features table could be created for each run, however a table holding 96 samples would be around 800MB which, with at least two runs per week, would quickly become prohibitively large. Given the relative short processing time, high throughput of arrays and long term availability of feature extraction files if a re-analysis is required it would be sensible to re-run SID than to return to stored values.

The probeorder, feparam and referencevalues tables should not be truncated. The feparam table holds the filenames used by the python script to prevent feature extraction files being reprocessed. The referencevalues table enables re-analysis of a sample in the future after a new reference range has been created and the probeorder table records the array design.

The consecutive_probe_analysis table could be truncated once the shared CNVs exported to the LIMS.

4.9.4 *Contingency plan*

Should SID be implemented and then become unavailable contingency plans include temporarily suspending the array service or reverting to the current strategy of manually phenotype mismatching hybridisation partners.

4.9.5 *Database backup*

SID will be incorporated to Trust IT's existing database backup strategy, utilising a 3rd party software to perform daily backups, keeping the last three backups.

4.9.6 *Validation*

ISO laboratory quality accreditation requires thorough validation of all tests and equipment. To validate SID 100 arrays can be analysed and 3 common CNVs assessed. The predictions made by SID can be compared to the signal intensities inspected manually on the array trace. The workload of manual assessing 300 regions is inline with validation of other tests within the department.

4.9.7 *Documentation*

Three documents are required to accompany SID:

ISO accreditation requires a validation document which details the steps taken to confirm SID performs as expected..

A SOP or user manual describes how SID is run and explaining some common error messages.

A development manual would be aimed at members of the bioinformatics department who will be responsible for developing and maintaining the script and database.. This will describe in depth how SID works, dependencies and a change log.
This would also contain instructions to perform tasks which the user cannot perform, for example completely removing arrays to enable the array to be reprocessed and creation and adoption of new reference ranges.

These documents would be stored in the controlled document management system (Q-Pulse) and on the laboratory wiki which hosts the current standard operating procedures for the microarray service.

4.10 FUTURE CONSIDERATIONS

4.10.1 *Array redesign/multiple array platforms*

The department currently has three different array designs, the routine design, one high resolution array used only for specific cases and a legacy design.

Should SID replace phenotype mismatching SID should also be required to process these array files. Each array design would require a probeorder table and a reference range. The array design is contained within the filename. SID would need updating to extract this to apply the correct tables.

Similarly any future changes to the routine array design would require the same measures.

5 CONCLUSION

SID is capable of processing raw feature extraction files, and by analysing each hybridisation partner independently identify CNV present in both hybridisation partners.

A call was identified within the known CNV in 100% of the 69 cases manually created true positive cases.

A number of calls outside these regions were also identified which may be false positive calls or true shared CNV within a common CNV region. An indication for the prevalence of these calls was shown in 121 prospective arrays.

The long term management of SID and further work required before SID can be implemented in clinical service has been detailed, including a strategy for analysing the sex chromosomes and migrating SID to the departmental server. The departmental LIMS system must be modified and link to SID created to capture shared CNV.

BIBLIOGRAPHY

[1] Joo Wook Ahn, Kathy Mann, Sally Walsh, Marwa Shehab, Sarah Hoang, Zoe Docherty, Shehla Mohammed, and Caroline Mackie Ogilvie. Validation and implementation of array comparative genomic hybridisation as a first line test in place of postnatal karyotyping for genome imbalance. *Molecular Cytogenetics*, 3:9, April 2010. PMID: 20398301 PMCID: PMC2885406.

[2] Phillipe Hupe. GLAD-Introduction. http://bioinfo-out.curie.fr/projects/glad/node2.html, November 2004.

[3] Siddharth Roy and Alison Motsinger Reif. Evaluation of calling algorithms for array-CGH. *Frontiers in Genetics*, 4, October 2013. PMID: 24298279 PMCID: PMC3829466.

[4] Feng Zhang, Wenli Gu, Matthew E. Hurles, and James R. Lupski. Copy number variation in human health, disease, and evolution. *Annual Review of Genomics and Human Genetics*, 10(1):451–481, 2009. PMID: 19715442.

[5] Joo Wook Ahn, Susan Bint, Anne Bergbaum, Kathy Mann, Richard P Hall, and Caroline Mackie Ogilvie. 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*, 6:16, April 2013. PMID: 23560982 PMCID: PMC3632487.

[6] Bertrand Jordan, editor. *Microarrays in diagnostics and biomarker development: current and future applications*. Springer, Heidelberg ; New York, 2012.

[7] John Dunlop and Jake Miller. Why haven't you adopted a patient to patient approach?, July 2015.

[8] JooWook Ahn. The frequency of abnormalities suggest a 1 in 6000 chance of two patients randomly having the same imbalance, June 2015.

[9] Agilent Technologies. CGH & CGH+SNP microarrays. http://www.genomics.agilent.com/en/CGH-CGH-SNP-Microarrays/;jsessionid=Hz91Jt8CfVWDlHljTS2ytPCJG1R3P9xvNZqtS8JrtxPowrT
PG-9, 2016.

[10] Agilent Technologies. eArray. https://earray.chem.agilent.com/earray/, 2016.

[11] Agilent Technologies. Agilent CytoGenomics 2.0 Reference Guide, 2011.

[12] E. S. Venkatraman and Adam B. Olshen. A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics (Oxford, England)*, 23(6):657–663, March 2007. PMID: 17234643.

[13] Hanni Willenbrock and Jane Fridlyand. A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics*, 21(22):4084–4091, November 2005. PMID: 16159913.

[14] Weil R. Lai, Mark D. Johnson, Raju Kucherlapati, and Peter J. Park. Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics (Oxford, England)*, 21(19):3763, October 2005. PMID: 16081473 PMCID: PMC2819184.

[15] Yue S. Niu and Heping Zhang. THE SCREENING AND RANKING ALGORITHM TO DETECT DNA COPY NUMBER VARIATIONS. *The annals of applied statistics*, 6(3):1306–1326, September 2012.

[16] MongoDB, Inc. MongoDB.

[17] S. van der Walt, S.C. Colbert, and G. Varoquaux. The NumPy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30, March 2011.

[18] SciPy. Scipy.org.

[19] Andy Dustman. MySQLdb1. https://github.com/farcepest/MySQLdb1, January 2014.

[20] Michael Bayer. SQLAlchemy.

[21] pyodbc. mkleehammer/pyodbc.

[22] PyMongo. Python Driver (PyMongo) Getting Started With MongoDB 3.0.4.

[23] R Foundation for Statistical Computing. R: A language and environment for statistical computing, 2014.

[24] Wolfgang Huber, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, Raphael Gottardo, Florian Hahne, Kasper D Hansen, Rafael A Irizarry, Michael Lawrence, Michael I Love, James MacDonald, Valerie Obenchain, Andrzej K Oles, Herve Pages, Alejandro Reyes, Paul Shannon, Gordon K Smyth, Dan Tenenbaum, Levi Waldron, and Martin Morgan. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, 12(2):115–121, January 2015.

[25] Venkatraman E. Sechan and Adam Olshen. Dnacopy : Dna copy number data analysis.

[26] MySQL. MySQL Community Server.

[27] SQL Server Express Edition Microsoft.

[28] Microsoft SQL Server Migration Assistant v5.3 for MySQL.

[29] HeidiSQL. HeidiSQL - MySQL, MSSQL and PostgreSQL made easy, 2016.

[30] Python Software Foundation. Python Language Reference, version 2.7, July 2010.

[31] Continuum. Why Anaconda?, April 2016.

[32] The Eclipse Foundation. Eclipse IDE.

[33] PyDev. Python IDE for Eclipse.

[34] Pylint. Code analysis for Python.

[35] Agilent Technologies. Aglient workbench - CGH interactive analysis user guide, April 2012.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

`http://code.google.com/p/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`