Visualising Errors in Animal Pedigree Genotype Data

,

Martin Graham[1], Jessie Kennedy[1], Trevor Paterson[2] and Andy Law[2]

[1]School of Computing, Edinburgh Napier University, UK
[2]The Roslin Institute, University of Edinburgh, UK

Abstract

Genetic analysis of a breeding animal population involves determining the inheritance pattern of genotypes for multiple genetic markers across the individuals in the population pedigree structure. However, experimental pedigree genotype data invariably contains errors in both the pedigree structure and in the associated individual genotypes, which introduce inconsistencies into the dataset, rendering them useless for further analysis. The resolution of these errors requires consideration of the genotype inheritance patterns in the context of the pedigree structure. Existing visualisations of pedigree structures are typically more suited to human pedigrees and are less suitable for large complex animal pedigrees which may exhibit cross generational inbreeding. Similarly, current table-based viewers of genotype marker information can highlight where errors become apparent but lack the functionality and interactive visual feedback to enable users to locate the underlying source of errors within the pedigree.

In this paper, we detail a design study steered by biologists who work with pedigree data, and describe successive iterations through approaches and prototypes for viewing genotyping errors in the context of a displayed pedigree. We describe how each approach performs with real pedigree genotype data and why eventually we deemed them unsuitable. Finally, a novel prototype visualisation for pedigrees, which we term the 'sandwich view', is detailed and we demonstrate how the approach effectively communicates errors in the pedigree context, supporting the biologist in the error identification task.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces – *graphical user interfaces;* J.3 [Life and Medical Sciences]: Biology and Genetics.

1. Introduction

Animal breeders and biologists study the inheritance of genetic markers in animal pedigrees in order to understand the genetic architecture, to identify regions of the genome that contain genes controlling traits of economic and welfare benefit, and ultimately to improve the quality and health of animal stock.

However, in practical terms, sample handling, pedigree recording and technical errors all conspire to result in data sets that are apparently inconsistent with the Mendelian laws of inheritance. Our paper describes the design of a visualisation tool that will allow biologists to investigate and repair errors within these particular sets of animal data - known as "pedigree genotypes".

In their simplest terms a pedigree is the inheritance structure of a group of animals, and the genotype is the genetic makeup of a particular individual. In diploid, sexually reproducing organisms, each individual will inherit half its genetic material from each parent and thus the genotype observed in any individual is constrained by the genotype of its parents. Because of the numerous possible causes of error, the size of the data sets and the complexity of the pedigrees with which animal breeders and biologists work, the cleaning of such data is a complex task.

We describe this central problem in greater depth in the following section, followed by a summary of previous work into pedigree visualisation. We then describe the generic data structure we are processing, and move to the main part of our work: the description of an iterative design cycle to produce a visualisation that can display and interact with pedigree genotype data and associated errors. Finally we draw some conclusions and detail future work.

2. Problem Description

For each genetic marker the genotype inherited by each individual within a pedigree is constrained by the Mendelian laws of inheritance describing the transmission of alleles (specific measurable variants for the marker) from parents to offspring. In sexually reproducing diploid organisms such as vertebrates, an individual has two sets of chromosomes: one inherited from each parent. Thus each individual's genotype for a given marker must consist of one allele from each parent. An observation of any genotype that is inconsistent with these rules is indicative either of an error in the pedigree information (or sample identification), an error in the genotype assignment, or more rarely, the result of a novel mutation.

Numerous genetic analysis algorithms, for example those for determining "linkage" between markers, require *completely* consistent pedigree genotypes as their input data. In experimental breeding studies however, discrepancies between the pedigree structure and the

observed inheritance patterns are relatively common, typically in the order of 1-3% of all collected data points [BA03]. Inheritance-checking algorithms can be employed to identify individuals with an apparent marker genotype that could not possibly have been inherited from their asserted parents, as demonstrated in Figure 1. In this example, there is an obvious discrepancy between Junior and Dad. Junior has failed to inherit an allele from Dad and also has a novel allele that could not have been inherited from either parent. But from this data alone, it is not possible to determine the nature of the error, as there are two categories of possible causes: pedigree errors (Dad and Junior are not parent and child) or genotyping errors (either Dad or Junior's genotype has been wrongly assigned).

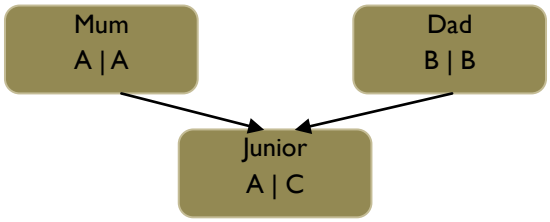

Figure 1: *A Simple Pedigree Error for a Single Marker.*

Pedigree errors (or apparent pedigree errors) occur either through deficiencies in record keeping, errors in sample handling (genotyping the wrong sample is functionally equivalent to a pedigree recording error) or "unplanned" mating events. Genotyping errors generally arise through technical problems in the assay procedure but may also be the result of failures in data handling processes.

Obviously, analyses conducted using inconsistent genotype datasets – if they can be run at all – are likely to result in erroneous conclusions, an example being that of chimpanzee paternity given in [PBBT05]. However, as previously noted, the identification of the precise nature of the error is non-trivial. The checking algorithms will highlight inconsistent transmission patterns but can distinguish neither the nature of the error (pedigree versus genotyping error), nor the source. In Figure 1's minimal example, the genotype error may be reported on Junior or even on the transmission relationship between Dad and Junior, but the actual error could be with Dad's genotype. An additional complexity is that experimental genotype data is often incomplete, with genotypes not determined for all individuals in the pedigree. Although the checking algorithms can infer possible genotypes in place of missing data points as it applies the Mendelian transmission rules across the pedigree, this can make the true location of genotype errors even more ambiguous as errors can easily be propagated. Therefore, inheritance checking algorithms cannot unambiguously pinpoint erroneous data points, only genotypes where the consequences of an erroneous data point are revealed as an inheritance inconsistency in the asserted or inferred genotypes of related individuals.

Biologists lack adequate tools with which to investigate the inconsistencies in genotype datasets. This exploratory task requires the data – in particular the errors – to be reviewed in the context of the full set of markers and the entire pedigree and is thus complex. This complexity is compounded by the study of large pedigrees of hundreds or thousands of individuals, genotyped for tens or hundreds *of* thousands of different genetic markers. It will only be tractable – even for an expert – with the assistance of software tools that allow visualisation of the errors in a biologically meaningful manner. The key challenge is to present the complex pedigrees extant in animal breeding experiments and populations in a manageable and understandable fashion and overlay that visualisation with the apparent errors within the dense genotyping data sets. If we have 10,000 markers and a given node/edge in the pedigree has errors over 500 of them, we might infer more than if the same node/edge only had errors for 1 marker. Therefore we need to address the double difficulty of not only clearly showing the pedigree structure but also communicating error information on top.

3. Previous Work

Most visualisations that display pedigree structure for genetic research have a standard layout method, influenced by an attempt to standardize human-pedigree nomenclature [BSU*95] and have originated from within the biology community. Generally, a top-down layered graph presentation is used to display the pedigree structure [ABH*98; BGP*99; HL07], and related genetic marker data, such as SNP data, is combined with the pedigree display as in HaploPainter [TN05] or the nodes of the pedigree graph used to display information such as gender and other selected attributes.

As these approaches use a familiar node-link representation to convey the pedigree structure they are limited to clearly displaying only pedigrees of a hundred or so individuals on-screen (and often less) as they fill screen space quickly. Many packages [GD04; MPW*05; SHP98; Won00] are designed to produce hard-copy outputs for printers rather than provide an on-screen interface to pedigree data, and hence user interaction is limited in these applications. Others [Col07; Zha06] produce static on-screen snapshots using third-party graphing software. The packages designed for human pedigrees assume that breeding almost always takes place between members of distinct generations, thereby making the standard layout less suitable due to the cross-generational links and sheer number of offspring in any one generation of a typical animal pedigree.

Some pedigree viewers aim to keep pedigrees on the screen rather than resort to hard-copy. One approach used to overcome space limitations is to show a 3D model of the pedigree data. PedVizApi [FFFP08] provides a Java-based widget that supplies a 2D and a 2.5D (fixed 3D perspective) view of pedigrees that can contain thousands of nodes. Celestial3D [LWE*08] offers a fully translatable 3D view of pedigree data and can extend to displaying multiple pedigrees. However these approaches suffer from problems common to 3D displays, namely occlusion and depth perception issues.

2D applications that try to visualize whole pedigrees include PViN [WL05] which uses overview and zoom windows to allow exploration of large pedigrees of tens of thousands of nodes. PPPA [MPB06] adopts a modified force-directed layout algorithm to display entire human genealogies; their particular modification aims to

differentiate individuals and families marked as carrying a particular genetic disease.

Other approaches with distinct InfoVis, as opposed to bio-science, roots concentrate on reducing the complexity of the visualisation and improving the interactive features available to a user [FZ95; MB05]. The rationale here is that deciding to draw a large pedigree in its entirety invariably generates an incomprehensible, user-unfriendly structure. Instead, these approaches allow a user to select a small number of focal individuals and then display ancestor and descendant trees for each of these individuals. The user is free to navigate to other individuals and [MB05] introduces techniques such as mouse gestures for interactive tasks and animation to clarify any changes in viewpoint.

An attempt to combine both pedigree-wide and individual-centric representations can be found in the recent GeneaQuilts prototype [BDF*10]. Their representation uses a sequence of cascaded lists, one per generation, with matings and consequent offspring indicated by pathways that emanate out from strips of family nodes placed orthogonally between successive generations. The overall visual effect is of a matrix, heavily weighted along its diagonal, with row and column labels placed strategically within the matrix rather than along the edges. Here the user can zoom out to see an overview of the entire pedigree, or use highlighting or filtering techniques to show or reduce the structure to the ancestors and descendants of a few selected individuals.

Focusing on an individual also allows simpler tree visualisation techniques to be used to show ancestor and descendant profiles for an individual. This is practical when considering human family trees as the amount of inbreeding (i.e. non-tree structure) is considerably less than in animal pedigrees, and any multiple paths that do occur can be resolved by displaying a small number of duplicate nodes in multiple locations. Several examples exist of using tree-based visualisations on human genealogy data in tandem with the individual-centric approach [DR08; MBS*07; WdPO04]. A recent H-Tree representation of pedigrees [TNS10] also uses this approach, though with the caveat that only ancestors of a single individual can be displayed. The descendant tree structure, being more varied than a simple binary tree, would not fit within an H-Tree layout. Returning to animal pedigrees, Peditree [vBH05] also simplifies the complexity of a pedigree structure so it can use a Windows Explorer-style tree widget.

Lastly, Hart & Ross [HR00] combine a display of inheritance characteristics (of a type) with an associated pedigree display for a visualisation of a simulated genetic algorithm. Here though, as there are no possibilities for 'alleles' to be mislabelled or individuals to have the wrong 'parents', their work does not explore the scenario of possible errors in their data and how they could be communicated and resolved.

In summary, the approaches which aim to alleviate layout difficulties and improve user interaction focus on communicating the structural aspects of a pedigree, with associated data being restricted to simple node labeling. In real pedigree data, there can be thousands of marker alleles to match between individuals and along with that comes the need to communicate through the visualisation which of these markers are inconclusive or in error; an aspect which is not supported by any of these existing visualisations – all of which assume a total correctness of the visualised data.

4. Pedigree and Genotype Data

Together, the pedigree and associated genotype data constitute a graph model where the pedigree forms the structure of the graph, with the genotype marker data giving each node in the graph a set of associated data.

The pedigree itself forms a restricted type of Directed Acyclic Graph (DAG) structure, where each individual can have zero or more children, and never more than two parents (in practice some parent(s) may be unknown, so there may be less than two in the data.) Having two parents per individual helps explain why we shy away from the 'family tree' label; pedigrees, especially animal pedigrees, are usually heavily interlinked and even when just considering either the offspring or ancestors of a single individual the structure is never guaranteed to be a true tree. Many pedigrees can also be roughly delineated into layers or generations, especially in the case of pedigrees resulting from controlled animal breeding. Generally the individuals of one generation produce offspring that form the next generation and so on, though even in controlled environments the distinctions can become blurred.

The product of associating extra data with a graph structure is known as a multivariate graph, though the term 'multivariate graph' (or network) has itself been used to cover several different types of graph data; essentially 'multivariate' has been prefixed to any type of graph where multiple variables have been shown to be present, either connected to the edges or the nodes, or both. Wattenberg [Wat06] took the idea of multivariate nodes, i.e. nodes with sets of categorical attributes, to produce a summarizing *pivot* graph of relationships between nodes with given attributes. An example case could be to show the linkages in a social network based on the attributes of home city and gender. Our definition of multivariate graph is closer to this use of the term, as we have multiple attributes associated with each node of the structure, e.g. gender, generation and genotype data, but do not make distinctions on the type of relationship between the nodes – all relationships are inheritance-based, and all other relational information such as sibling and mate is derived from this.

5. Prototype Design Cycle

Exploring large, complex data sets to source and correct errors is challenging, especially when the errors may have non-obvious causation and only show their effects at a distance removed from their source (in this respect, tracing errors in genotyped pedigrees can be considered analogous to bug-hunting in computer programming.) The challenge is to develop a suitable visual representation for the pedigree genotype data which would communicate the source of any discovered errors. What follows is a description of the design process and iterations we proceeded through that resulted in the current visual representation and interaction mechanisms of the pedigree genotype viewer.

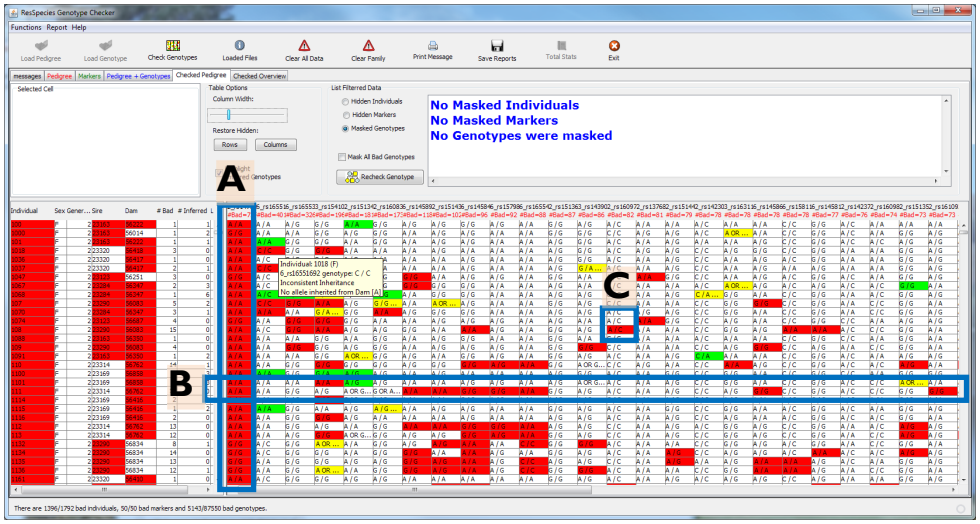

Figure 2. *The GenotypeChecker visualisation, with individual animals organised along rows and markers along columns.Colour coding is used to indicate errors and uncertainties No account however is made for pedigree structure making it impossible to track the cascading of errors inferred through the inheritance hierarchy.*

In order to clean a genotype dataset, a biologist needs to be able to locate and identify the actual cause of all reported genotype inconsistencies and remove the source error. Typically this is a process of data (genotype) removal, although it might be possible to resolve wrongly labelled samples or pedigree links, after further experimental confirmation. Problematic branches of a pedigree may be deleted, but individuals with descendants cannot be removed from the pedigree, instead they may be treated as 'unknown' by deleting all of their asserted genotypes. The key to determining the steps for data cleaning is to allow the biologists to visualise the errors identified in the full context of the pedigree.

To this end we have designed, developed and evaluated a number of exploratory visualisation prototypes by means of a collaborative formative evaluation between visualisation and biological experts. Each prototype reached varying levels of maturity before it was either discontinued or finished; usually the speed at which a prototype was discarded reflected its inability to convey basic information clearly and efficiently such as families within a pedigree. The prototypes were implemented and tested with real data sets to ensure the visualisation could represent the typical pedigrees and genotype information in a meaningful way for the biologists, and each produced findings that fed directly into the next iteration of prototype development. Evaluation consisted of monthly meetings between the biologists and prototype developers where ideas were mocked-up, discussed and examined, and for concrete prototypes the biologists would examine the range of interaction each provided in terms of displaying pedigree information and the underlying problem of identifying errors within their data sets. The major iterations in this sequence of visualisation design and development are discussed here, with accompanying screenshots showing how each displayed a currently representative real data set of 1,792 animals genotyped across 281 markers.

5.1 Table-based Visualisation

We had previously developed a prototype software tool, 'GenotypeChecker' [PL11], for the purpose of cleaning pedigree genotype data sets, and we took this as the starting point for the development of a more pedigree-aware interface. The application is built on the ResSpecies Model and API for population genetics [Res06] which includes a genotype checking algorithm. For each genotyped marker this algorithm applies the rules of Mendelian inheritance across the population pedigree and both infers any missing genotypes that can be resolved from the asserted genotypes and reports back any inconsistencies i.e. any individuals for which the asserted genotype cannot be true when the rules are applied to the data.

The data is presented as a table of individuals (rows) by markers (columns), with individual genotype data points filling the cells. After running the checking algorithm, colour-coding is used to highlight inconsistent genotypes and individuals in red and inferred genotypes in yellow and green. Especially problematic markers or individuals will be noticeable as columns or rows with large amounts of colouring. In order to trace the root causes of reported errors the user explores the results by sorting on a variety of metrics (individuals, markers, error frequency *etc.*) and can generate sub-tables of selected markers of interest, or views of the immediate family members of a particular individual. The user can interactively test hypotheses for which data points are erroneous and should be excluded from the dataset by 'masking' genotypes or individuals and reapplying the checking algorithm, successively removing bad data until no errors are reported.

Through observing use of this prototype application by the biologists we saw that users quickly filtered out data for both completely consistent and very problematic markers and then proceeded to mask (remove) genotype data from very troublesome individuals (i.e. pedigree /identification errors). Then by exploring the pedigree inheritance pattern

on a marker by marker basis, they attempted to identify and mask likely candidate erroneous genotypes, so that when the checking algorithm is reapplied the reported inconsistencies in close relatives also disappeared.

Figure 2 illustrates part of the analysis of a real anonymised data set containing 1,792 individuals, genotyped for 281 markers over 3 generations. One marker (Figure 2 - box A) is clearly the most problematic, reporting multiple inconsistent genotypes, and probably reveals a problem with that genotype assay (or some systematic recording error). Most individuals report only sporadic inconsistent genotypes (0 or rarely 1, 2 or 3 errors) but a single individual reports 8 inconsistencies (Figure 2 - box B), possibly indicating a mis-identified individual or DNA sample or incorrect pedigree information for this individual. The more sporadic errors (such as Figure 2 - box C) require more careful analysis in the context of close family members to pinpoint the likely source error for genotypes of that marker. For errors where the problem is not an evidently universally corrupt marker, the lack of pedigree-aware visual context becomes crucial.

This error identification, confirmation and removal can still be a laborious iterative process, but the major limitation reported with the GenotypeChecker prototype was the tabular display of genotype information, whereas the user needs to explore the information in the context of the pedigree structure, The ability to view errors within the genotyping data in the context of the pedigree is vital in a genotype wrangling tool. For example, if multiple siblings appear to have genotypes inconsistent with one or other of their parents, then the biologists' thoughts will shift towards the error lying in the identity of the parent/the parental sample. If a single animal within a large nuclear family shows multiple inconsistencies, then the focus will shift to the offspring sample.

Given the restrictions imposed by the table view we then began to iterate through a succession of prototypes designed to communicate the pedigree structure of the data set under examination.

5.2 Node-link Visualisations

Using the data sets available for the GenotypeChecker prototype, we explored two different graph layout techniques, one a general graph layout and the other for DAG (Directed Acyclic Graph) drawing.

The general graph visualisation of the same data set seen in Figure 2 can be seen in Figure 3, and immediately shows the shortcomings of such an approach. Rather than free the data to use the display in a less rigid manner than traditional pedigree layouts, what appears is the standard multiple edge crossings that plagues force-directed graphs, the only discernible information being the lop-sidedness of the breeding in terms of gender, with a few males shown in blue mating with many more females (in pink). On another data set, the force-directed layout did reveal that one 'pedigree' was in fact made of two unconnected structures which moved apart. The biologists were negative with their feedback to this approach, influenced by their previous experience with such representations, and taking lessons from past examples where the ability to trace direction within a structure was of vital importance in representation choice [GKH00], it was agreed that the general graph layout representation was unsuitable even before we reached the stage of how to layer error information on top of the graph visualisation.

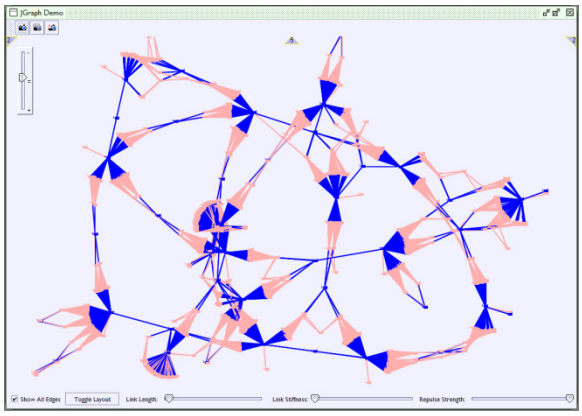

Figure 3: *A general graph display of a pedigree with 1,792 individuals. The graph concentrates around a few males and the sense of 'direction' in the pedigree is lost.*

We then developed a prototype that displayed the pedigree as a DAG, using a barycenter heuristic for node positioning. Barth et al's [BMJ04] fast edge-crossing counting algorithm was used to detect when the number of edge crossings reached a limit. Edge crossings can be overwhelming in a full pedigree diagram so our default setting was to show only the relationships of selected individuals in the pedigree, though the option remains to show all the edges at once, as in the screenshot in Figure 4.

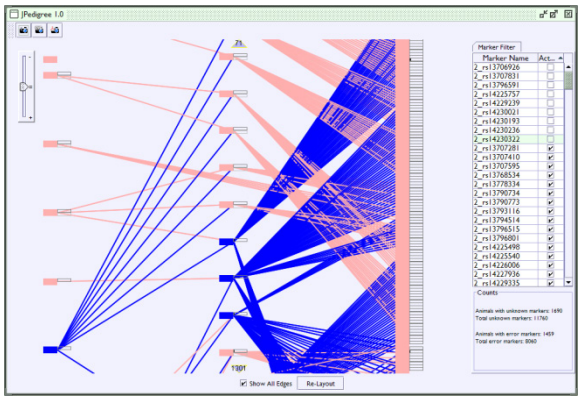

Figure 4: *A restricted graph layout of a pedigree as seen in many traditional pedigree representations.*

We used the same simple node display as in the general graph, plus a basic error representation: marker errors and uncertainties for individuals were shown as black and grey bars adjacent to the respective nodes, the length of each bar proportional to the number of errors and uncertainties. Simple pop-up menu driven navigation such as moving to a child or parent of the node under the pointer, and zooming out to see all its immediate relations was also developed, as was a mouse-wheel driven zooming action and indicators of where and how much of the structure was off-screen.

These however were essentially techniques to combat the limitations of this style of graph representation for rendering this type of data set. Following edges and especially crossing edges is known to be a cognitively demanding task [HE05] in graph visualisations. Trying to follow sequences of paths to reconcile relationships that are once or twice removed is predictably more difficult.

Following a demonstration to and further discussions with the biologists it was decided to explore an alternative to network displays for the pedigree data, namely matrix visualisations.

5.3 Matrix Visualisation

Matrix visualisation is the major alternative visualisation technique to node-link displays for graph/network data; source and destination nodes form the rows and columns and edges are placed in the cells of the matrix to indicate relationships between source and destination nodes.

The naïve solution of developing one single matrix visualisation for an entire data set was considered but dismissed without prototyping. Calculation showed that such a matrix would always be 90%+ empty space. Unlike software call graphs [Ham03] or social networks [HF07] that regularly use matrix visualisations, our data is bounded by a density limit – there cannot be more then $2N$ edges in a pedigree of N individuals. It is also, with a few exceptions, separable into discrete sub-graphs, one per generation, where the destination nodes of one generation form the source nodes for the next. The very features of the data structure that made it conducive to visualisation using a DAG layout, made it distinctly unsuitable to rendering as a single matrix. Essentially we would see a series of sparse sub-matrices embedded along the diagonal of an otherwise empty, and much larger, matrix save for a few outliers representing cross-generational breeding. While this might be revealing for a data set of unknown structure, we already know this pattern exists in pedigree data. Further, path routing and tracking within matrix visualisations has been shown to be the most significant drawback with this style of representation [GFC04].

We then explored whether any available IV applications could support the visualisation of our pedigree data and could possibly be adapted to work with error information layered on top. Of the existing techniques we surveyed, GeneaQuilts was the only one that could support the size of datasets we were using. GeneaQuilts offers a novel view of pedigrees, with each generation forming a staggered sub-matrix of individuals by families and the pedigree relationships indicated via paths and intersections of these rows and columns. Successive generations are linked to the previous generations by sharing family columns. Interaction is fluid with path highlighting, smooth pan and zooming, and focus and context mechanisms on selected nodes.

However, the GeneaQuilts visualisation does not provide a consistent focal point between offspring and parents or between full and half-siblings which the biologists recognised was vital for the potential assessment of errors. Viewing two adjacent generations together can only be done if the generations in question are small or zoomed out until most detail is lost. Even with coloured selections to ease path following through the braids of alternative paths, the operator must track back along lines composed of multiple orthogonal elements. The indication of family context is remote from both the parent and the offspring labels. To address this Geneaquilts includes a fluid navigation device for moving directly to parents or children of an individual that works well for human-scale offspring, and the ability to filter on a few selected individuals and their relatives. Figure 5 shows the same data set as displayed in the previous figures in GeneaQuilts, highlighting the layout issue and also a problem with the navigation guide which cannot cope with the number of offspring for the selected male.

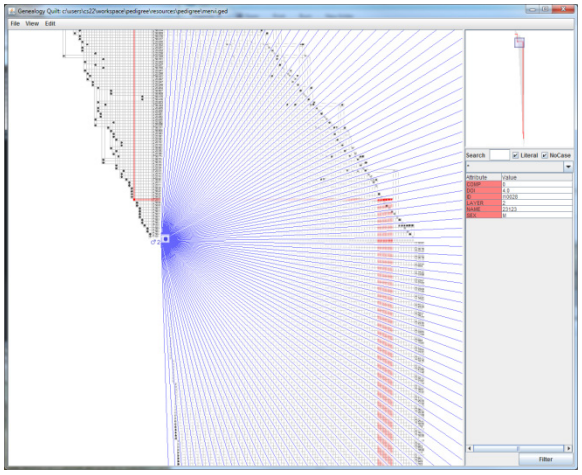

Figure 5: *The wide fan-out of animal pedigrees causes problems for many visualisations that handle human pedigrees well, e.g in the GeneaQuilts system, the number of offspring for some males overwhelms the navigation aid.*

We took lessons from these findings and developed our next visualisation prototype as a series of matrices, one per generation, which would convey the matings between individuals in one generation: the males and females along the axes with the offspring in the 'cells'. This differs from the standard semantics of a graph structure represented in a matrix, where one axis represents a set of sources and the other a set of sinks (often the sets overlap) with the intersections of rows and columns used to show the presence and properties of edges. Here the axes represent two categories of a distinct property (gender) and the objects at the column/row intersections represent a collection of one or more offspring of the two individuals at the axes of the particular row and column. The edges from these offspring groups are implicit up and across to the parent nodes on the axes.

For most human pedigrees this again would generate a series of matrices with >90% empty space, especially where polygamy (>1 partner) is not evident. However, animal breeding experiments typically take the form of a few males mating with a much larger number of females, and thus we were confident there would be rows/columns with multiple entries in the visualised matrices.

A screenshot of the multiple-matrix prototype is shown in Figure 6. It was revealing that even with multiple

matings, much of the matrix was still empty – most of which can be explained by realising that while males may mate with more than one female in a given generation, the opposite situation very rarely occurs, thus the row/columns for females tended to have only one entry each. On the positive side it was also noted that separating the males and females from each other within this style of visualisation, as opposed to the graph or other matrix representations, allowed simple yet powerful sorting methods to be performed. The rows and columns could be ordered by the individuals with most children or most partners, or by properties of their own parents. Ordering the animals by something as simple as name or ID also made locating known pre-existing individuals or examples much easier than in the node-link visualisations.

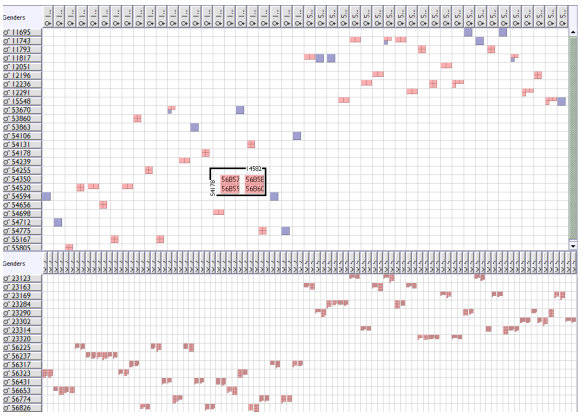

Figure 6: *The multiple matrix visualisation was still composed of mainly empty cells.*

At this stage error data was not overlaid on the display, the colour coding used was essentially a classic "pink for girls" and "blue for boys" for the offspring – colour coding was not needed for the parents as males and females were already separated by axis.

Discussion with the biologists revealed that as the main point of the interface is to track/visualise errors then actually showing the gender of offspring was not important. They reinforced after viewing the multiple matrix visualisation and GeneaQuilts that the idea of families were central to error detection; viewing an individual's errors in the context of its siblings would give a good indication of error type and errors as a whole for an offspring set would signify certain procedural errors.

The issue of unused space was still an aggravating feature of the display, and we realised that if either the males or females in a generation were monogamous (i.e. mated with only one partner) then a matrix was not necessary, but simply two parallel lists of females and males with their offspring in a third list in between would suffice. This would collapse the matrix down to a three row table for each generation. Bearing these findings from the matrix visualisation in mind we moved onto development of a new technique – the sandwich visualisation.

5.4 Sandwich Visualisation

The Sandwich visualisation was the result of this recalibration from a matrix per generation to a simpler three-row table per generation. In essence, if the relationship in matings in one generation between the two genders was one-to-one or one-to-many (usually single males to many females) then the resulting relationships and offspring could be represented in this style. This works as the individuals in each generation are divisible into one of two types, male or female, and offspring have exactly one relationship each in the pedigree to a parent of each gender. More formally, each generation's relationship with its offspring forms a one-sided regular bipartite graph of degree 2. Using this technique to represent a general bipartite graph where there is the possibility of nodes in both layers having a higher edge degree than two would not be possible without dummy offspring representations.

Where an individual (usually male) has multiple matings, this is communicated by simply repeating the entry for that individual so it is positioned in all the columns that correlate to its partners. In a table ordered by male properties this leads to such individuals forming larger contiguous multi-column cells in the table as seen in Figure 7. If the ordering is such that a node is fragmented into several disjoint places, connecting arcs are used to indicate the relationship between those parts. This technique is also used when both genders have individuals who have mated with multiple partners i.e. the mating pattern within generations is many-to-many.

This representation finally gives us the family-centric visualisation that is necessary to view and assess errors in the context of a pedigree structure. Into this representation, we then introduced the notion of communicating error size and type. A simple discrete four-level colour-coding is used to indicate the proportion of erroneous markers associated with an individual: white means no errors in the marker set with an individual, a lighter shading indicates few errors (0-5%), a mid-shading indicates moderate error levels, and a heavy shading indicates many (>20%) errors.

These errors are divided into three types: markers where nothing traceable was found from the father, nothing from the mother, or where a new allele was seen - and often errors could occur in more than one of these categories for any given combination of marker and individual. The three categories of error were then represented in the offspring

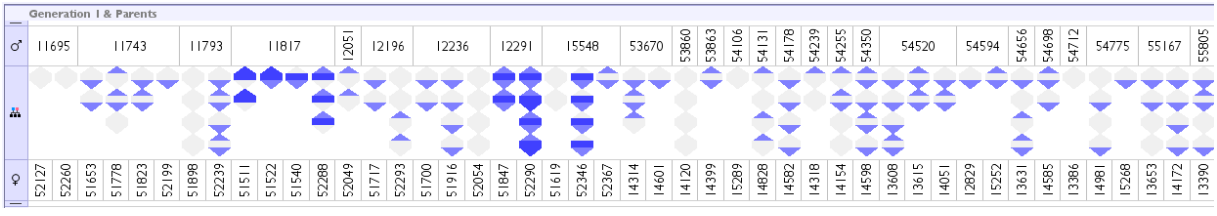

Figure 7: A *sandwich visualisation of two adjacent generations. Parents are assigned to rows by gender with offspring in-between. Some of the cells in the top row have extended to cover multiple matings with partners. A progressively darker shading of blue indicates individuals with more errors.*

row as the component parts of a hexagonal glyph, with the tips acting as stylised arrows oriented either up or down. These tips point to the sire and dam rows with the colour coding for the sire or dam errors, and the 'mid-stripe' of the hexagon is coloured for novel allele errors. In the sire and dam rows, the combined error count for the sire or dam is used to colour the representation of an individual. Figure 8 is a graphical key to the basic representation described above.

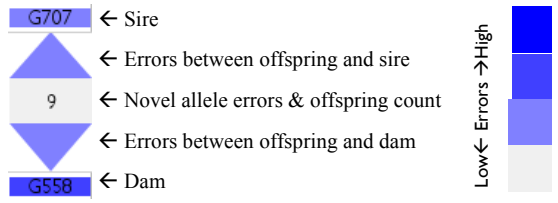

Figure 8: A hexagonal glyph and its colour-coded sections.

What we had now was a simpler visualisation when compared to the earlier graph-based visualisations and certainly more space-efficient than either the graph or matrix-based visualisations, with the added bonus that offspring are always directly adjacent to their parents in the sandwich view for their generation

Evaluation of the sandwich visualisation with the biologists highlighted some improvements. One of our findings from the previous matrix prototype was the identification of matings as vital, whereas individual offspring were not so crucial. All that was required for a high-level overview was a representation of the degree of error resulting from that mating. This would then allow us to reduce the number of individual objects to be rendered per generation and the technique could be repeated for each generation. If a mating was found to be worthy of further investigation a standard graph layout display or reverting to the individual offspring display could be used for drill down to see all relevant individuals in more detail. We therefore decided to make the default representation for any group of offspring a combined stylization, with the actual count represented textually.

In Figure 9, three generations of a pedigree are displayed, (the relationships between two of these generations were shown in Figure 7), but with aggregated representations for the children. Here, a group of offspring between two parents is represented as a single hexagonal glyph, displayed with a label indicating the number of individuals this aggregate represents. The error in each aggregate representation for each type (to sire, dam, or novel allele) can be calculated either as a mean average of the individual error types, or by taking the maximum values of those found in the offspring set. This choice, along with the option to revert back to the individual child view, can be made by the user and the visualisation simply uses a new renderer for the offspring row.

Two selection sliders allow the user both to filter out from the analysis markers above the selected error threshold and to alter the colour level thresholds for error display in the sandwich view. This filtering allows the biologist to home in on problematic mating pairs (families). In the example shown in Figure 9 three males of generation 0 sire families with severe levels of inheritance inconsistencies, indicating closer investigation of these matings is warranted. Individuals or families in the sandwich view can be selected, highlighting descendants and ancestors in yellow, to reveal any consistent patterns in error transmission. For example both families fathered by male 12291 exhibit high levels of inheritence inconsistency. However, following daughters of 12291 mated in the following generation reveals much lower rates of error transmission, suggesting that the original data error may lie with the paternity of

Figure 9: *Aggregated error indicators across a three-generation pedigree with a cut-off that can be controlled through histograms with sliders. Here, with the few most troublesome markers excluded, three sires' offspring are revealed as particularly problematic, and the descendants of one sire (12291) selected and highlighted in yellow.*

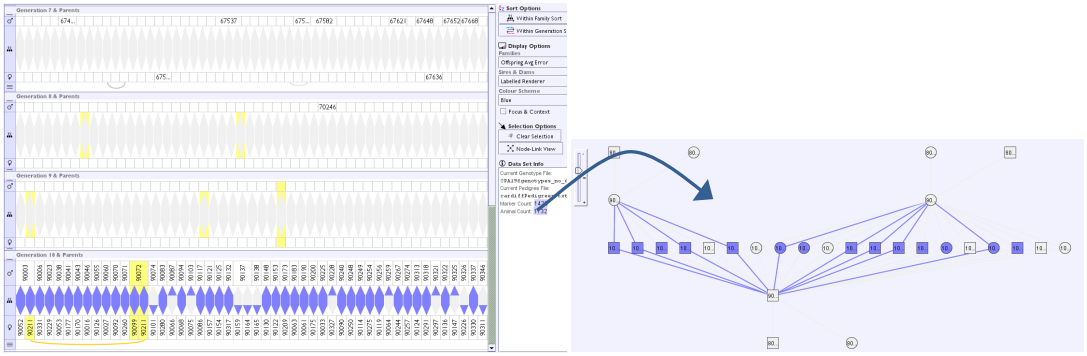

Figure 10: *An almost error-free multi-generation data set, apart from what appears to be a systematic series of errors between the bottom two generations. Selected individuals can also be viewed in the more traditional graph view on the right.*

these two families.

We now have an overview presentation which summarizes the aggregate error metrics in the context of the pedigree, and the detail in this overview can be compressed by providing a summary based on families (mating pairs) belonging to each generation. Situations such as assay-wide marker errors can be revealed as in Figure 10 where many of the bottom generation children are reporting errors reconciling their genotypes with their parents. This may be a record-keeping or serious sampling error, a notion strengthened by the noticeable lack of error elsewhere in the pedigree.

6. Conclusions

In conclusion, we have developed a prototype visualisation tool for exploring inconsistencies in the inheritance of genotypes within pedigree datasets with the aim of expediting the identification and repair of data errors. The design has been derived from analysis of the requirements and working practices of experienced biologists and consideration of the pros and cons of existing pedigree, graph and matrix visualisation techniques.

The visualisation has been designed to accommodate large pedigree and genotype data structures, maximizing the space efficiency of the pedigree layout whilst retaining aspects of a top-down layered node graph familiar to biologists. By collapsing individual nodes to a family-based representation we retain the level of detail required for a meaningful interpretation of the inheritance patterns exhibited at an overview level. The identification of problematic families is thus simplified and they can be explored at greater detail. The detailed drill-down exposition of problematic genotypes within families is supported by a linked display based on a more traditional pedigree layout which is effective for small graphs.

For pedigree data sets with more than approximately 200 matings per generation, the linear layout of each generation may cause either overrun of the screen bounds if the cells are set to a fixed width, or reduction to such a size that individual labels cannot be displayed. However, this can be dealt with using a focus and context technique which we are currently implementing. In tandem with the sorting on individual sires, dams, and error concentration already provided we provide attention clues by giving more space to items on the left-hand side of the screen and less to the right. Thus the items of interest, as chosen by the particular sort metric in use, are brought into focus.

7. Future Work

Currently the visualisation and the genotype checker are loosely coupled and do not support the interactivity required for hypothesis testing and data cleaning. Following the approach of our earlier GenotypeChecker application [PL11] the user will be able to test the effect of removing candidate errors by selecting problematic datapoints and temporarily removing or masking these genotypes, and then reapplying the inheritance checking algorithm. By these means the user will be able to incrementally identify and remove the minimal set of bad datapoints that must be removed to create a completely consistent dataset.

8. References

[ABH*98] AGARWALA, R., BIESECKER, L. G., HOPKINS, K. A., FRANCOMANO, C. A., and SCHAFFER, A. A.: Software for Constructing and Verifying Pedigrees within Large Genealogies and an Application to the Old order Amish of Lancaster County. *Genome Research 8*, 3 (March 1998) 211.

[BA03] BRUSH, G. and ALMASY, L.: Pedigree and genotype errors in the Framingham Heart Study. *BMC Genetics 4*, Suppl 1 (December 2003). doi:10.1186/1471-2156-4-S1-S41.

[BDF*10] BEZERIANOS, A., DRAGICEVIC, P., FEKETE, J.-D., BAE, J., and WATSON, B.: GeneaQuilts: A System for Exploring Large Genealogies. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (Nov/Dec 2010) 1073-1081. doi:10.1109/TVCG.2010.159.

[BGP*99] BRUN-SAMARCQ, L., GALLINA, S., PHILIPPI, A., DEMENAIS, F., VAYSSEIX, G., and BARILLOT, E.: CoPE: a collaborative pedigree drawing environment. *Bioinformatics 15*, 4 (April 1999) 345-346.

[BMJ04] BARTH, W., MUTZEL, P., and JÜNGER, M.: Simple and Efficient Bilayer Cross Counting. *Journal of Graph Algorithms and Applications 8*, 2 (2004) 179-194.

[BSU*95] BENNETT, R. L., STEINHAUS, K. A., UHRICH, S. B., O'SULLIVAN, C. K., ROBERT G. RESTA, LOCHNER-DOYLE, D., et al.: Recommendations for Standardized

Human Pedigree Nomenclature. *American Journal of Human Genetics 56*, 3 (1995) 745-752.

[Col07] COLE, J. B.: PyPedal: A computer program for pedigree analysis. *Computers and Electronics in Agriculture 57*, 1 (2007) 107-113.

[DR08] DRAPER, G. M. and RIESENFELD, R. F.: Interactive Fan Charts: A Space-saving Technique for Genealogical Graph Exploration. In 8th Family History Technology Workshop (2008). Retrieved 21 August from http://fht.byu.edu/prev_workshops/workshop08/papers/1/1-1.pdf.

[FFFP08] FUCHSBERGER, C., FALCHI, M., FORER, L., and PRAMSTALLER, P. P.: PedVizApi: a Java API for the interactive, visual analysis of extended pedigrees. *Bioinformatics 24*, 2 (2008) 279-281.

[FZ95] FURNAS, G. W. and ZACKS, J.: Multitrees: Enriching and Reusing Hierarchical Structure. In Proc. SIGCHI Conference on Human Factors in Computing Systems (1994), pp. 330-336.

[GD04] GARBE, J. R. and DA, Y.: Pedigraph user manual. Department of Animal Science, University of Minnesota, 2004.

[GFC04] GHONIEM, M., FEKETE, J.-D., and CASTAGLIOLA, P.: A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations. In Proc. IEEE InfoVis (2004), pp. 17-24.

[GKH00] GRAHAM, M., KENNEDY, J. B., and HAND, C.: A Comparison of Set-Based and Graph-Based Visualisations of Overlapping Classification Hierarchies. In Proc. ACM AVI (2000), pp. 41-50.

[Ham03] VAN HAM, F.: Using Multilevel Call Matrices in Large Software Projects. In Proc. IEEE InfoVis (2003), pp. 227-232.

[HE05] HUANG, W. and EADES, P.: How People Read Graphs. In Proc. Asia-Pacific Symposium on Information Visualisation (2005), vol. 45, pp. 51-58.

[HF07] HENRY, N. and FEKETE, J.-D.: MatLink: Enhanced Matrix Visualization for Analyzing Social Networks. In Proc. Interact (2007), vol. LNCS 4663, pp. 288-302.

[HL07] HE, M. and LI, W.: PediDraw: A web-based tool for drawing a pedigree in genetic counseling. *BMC Medical Genetics 8* (June 2007) 31. doi:10.1186/1471-2350-8-31.

[HR00] HART, E. and ROSS, P.: Enhancing the Performance of a GA Through Visualisation. In Proc. Genetic and Evolutionary Computation Conference (2000), pp. 347-354.

[LWE*08] LOH, A. M., WILTSHIRE, S., EMERY, J., CARTER, K. W., and PALMER, L. J.: Celestial3D: a novel method for 3D visualization of familial data. *Bioinformatics 24*, 9 (2008) 1210-1211.

[MB05] MCGUFFIN, M. and BALAKRISHNAN, R.: Interactive Visualization of Genealogical Graphs. In Proc. IEEE Symposium on Information Visualization (2005), pp. 17-24.

[MBS*07] MARTEL, J., BUTTERFIELD, J., SKOUSEN, G., LAWYER, D., and RICE, J.: A Highly Interactive Pedigree Viewer. In 7th Family History Technology Workshop (2007). Retrieved 21 August, 2008 from http://fht.byu.edu/prev_workshops/workshop07/papers/4/Pedigree-Viewer.pdf.

[MPB06] MAZEIKA, A., PETERSONS, J., and BÖHLEN, M. H.: PPPA: Push and Pull Pedigree Analyzer for Large and Complex Pedigree Databases. In Proc. 10th East European Conference on Advances in Databases and Information Systems (2006), vol. 4152, pp. 339-352.

[MPW*05] MÄKINEN, V.-P., PARKKONEN, M., WESSMAN, M., GROOP, P.-H., KANNINEN, T., and KASKI, K.: High-throughput pedigree drawing. *European Journal of Human Genetics 13* (2005) 987-989. doi:10.1038/sj.ejhg.5201430.

[PBBT05] POMPANON, F., BONIN, A., BELLEMAIN, E., and TABERLET, P.: Genotyping Errors: Causes, Consequences and Solutions. *Nature Reviews Genetics 6*, 11 (November 2005) 847-859. doi:10.1038/nrg1707.

[PL11] PATERSON, T. and LAW, A.: GenotypeChecker: An interactive tool for checking the inheritance consistency of genotyped pedigrees. *Animal Genetics In Press* (2011).

[Res06] RESSPECIES (2006).ResSpecies - Generic Species Resource Database.(The Roslin Institute

[SHP98] STAJICH, J. E., HAYNES, C., and PERICAK-VANCE, M. A. (1998).PEDPLOT: The Pedigree Plotting Program for the Pedfile Format (Duke University, Duke University Medical Center).

[TN05] THIELE, H. and NÜRNBERG, P.: HaploPainter: a tool for drawing pedigrees with complex haplotypes. *Bioinformatics 21*, 8 (2005) 1730-1732. doi:10.1093/bioinformatics/bth488.

[TNS10] TUTTLE, C., NONATO, L. G., and SILVA, C.: PedVis: A Structured, Space-Efficient Technique for Pedigree Visualization. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (Nov/Dec 2010) 1063-1072. doi:10.1109/TVCG.2010.185.

[vBH05] VAN BERLOO, R. and HUTTEN, R. C. B.: Peditree: Pedigree Database Analysis and Visualization for Breeding and Science. *Journal of Heredity 96*, 4 (2005) 465-468. doi: 10.1093/jhered/esi059.

[Wat06] WATTENBERG, M.: Visual Exploration of Multivariate Graphs. In Proc. SIGCHI Conference on Human Factors in Computing Systems (2006), pp. 811-819.

[WdPO04] WESSON, J., DU PLESSIS, M. C., and OOSTHUIZEN, C.: A ZoomTree Interface for Searching Genealogical Information. In Proc. 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa (2004), pp. 131-136.

[WL05] WERNERT, E. A. and LAKSHMIPATHY, J.: PViN - A Scalable and Flexible System for Visualizing Pedigree Databases. In Proc. ACM Symposium on Applied Computing (2005), pp. 115-122.

[Won00] WONG, L.: Visualization and Manipulation of Pedigree Diagrams. *Genome Informatics 11* (2000) 63-72.

[Zha06] ZHAO, J. H.: Pedigree-drawing with R and graphviz. *Bioinformatics 22*, 8 (15 April 2006) 1013-1014. doi:10.1093/bioinformatics/btl058.