Redeeming Pedigree Data with an Interactive Error Cleaning Visualisation

Martin Graham, Jessie Kennedy
Edinburgh Napier University
10 Colinton Road
Edinburgh, EH10 5DT, UK
+44 (0)131 455 2749
{m.graham, j.kennedy}@napier.ac.uk

Trevor Paterson, Andy Law
Roslin Institute, University of Edinburgh
Easter Bush
Midlothian, EH25 9RG, UK
+44 (0)131 651 9154
{trevor.paterson, andy.law}@roslin.ac.uk

ABSTRACT

We describe a visual data cleansing application for pedigree genotype data, which is used to redeem otherwise unusable pedigree data sets. Biologists and bioinformaticians dynamically and iteratively mask pieces of information from a dirty data set and graduate towards a usable cleaned version of the data, which can then be saved and used in ongoing biological analyses. Cleansing of such data is complicated over and above simple error cleaning in that change or masking of the pedigree structure may shift errors to new parts of the pedigree. Thus a branching history of data manipulations is kept to allow users to restore the data and visualisation to any of the previous states it has travelled through.

Categories and Subject Descriptors

H.5.2. [**Information Interfaces and Presentation**]: User Interfaces – *Graphical user interfaces (GUI)*

General Terms

Algorithms, Design, Human Factors

Keywords

Data wrangling, data cleansing, pedigree visualization

1. INTRODUCTION

Pedigree genotypes are datasets constructed from overlaying genetic marker data collected from a set of individuals onto the genealogy of that same set of individuals. With genetic inheritance, it follows that the values of offspring at any given marker should be inherited from their parents' values for that marker. However, this naïve assumption about data quality is incorrect and errors abound in the data as offspring have values that cannot be traced to one or both of their parents. The reasons for these errors are many and varied, but the end result is that the data is rendered unreliable and invariably useless for downstream analyses. As such, there is a need to clean pedigree genotype data so that biologists can effectively use it in their work.

2. RELATED WORK

A recent direction in visualisation research has been the interactive visual transformation of data, which can include data cleaning as one of the goals. Kandel *et al's* [1] Wrangler is a framework that allows users to visually specify transformations on data sets to improve data quality, which has various advantages over manual cleaning such as re-use of scripts and speed of operation. Bilgic *et al's* [2] more specific application attempts to clean data in social networks i.e. merge instances that refer to the same person (i.e. "j.smith" vs. "John Smith") or, conversely, to mark similar names as distinct entities. Our work is more in the vein of this latter approach as we have a well-specified data domain with a central overarching task – remove errors from the data – with no requirement for any other data transformation.

The concepts of missing and erroneous data, though united under the umbrella term of "uncertainty data" [3], can be seen as having somewhat opposing attributes. Erroneous data is present and definitely wrong, whereas missing data is absent and therefore we cannot know if it is wrong or not. Existing research into specific techniques for uncertainty visualisation [4] tends to concentrate on how to communicate uncertainty parameters and provenance information attached to data rather than data that is clearly wrong.

3. METHOD

3.1 Problem Description

Table 1. A small matrix of individuals vs. markers. Incorrect states are highlighted in red.

	M1	M2	M3	M4	M5	Errors
I1						1
I2						3
I3						1
I4						1
I5						0
Errors	1	1	3	0	1	6

A small example data set is shown in Table 1 as a matrix of individuals against *markers*, where each cell represents a *genotype* - the value an individual for a particular marker. Markers can vary in type, but the industry standard are SNP (Single Nucleotide Polymorphism) markers, each of which consist of a single base-pair represented by a pair of letters from the {G,T,C,A} set. Adding up the errors in the columns and rows gives the errors per individual and per marker for the data set. While this method of representation gives an overview of the data set that is useful for describing the problem, and indeed mirrors the solution found in [5], it falls down when presented with real world data sets as it cannot a) show the pedigree structure of the individuals or b) handle in a visually compact representation the amounts of markers present in real-world data sets. It is worth noting that when this project started, the biologists envisaged future data sets as having up to 10,000 markers – they recently received a data set with 250,000 markers.

AVI '12, May 21-25, 2012, Capri Island, Italy.

3.2 Interface Description

The interface to our data cleansing tool and its main constituent components are shown in Figure 1, and described in more detail in our previous work [6]. Unsurprisingly, given the possible size of these data sets, and specifically the number of markers, the interface uses aggregations over the marker and individual sets. The histograms (a) show the distribution of errors per marker or per individual, either upon load (the initial state as in the first two histograms) or after user applied cleaning (the current state is shown per individual in the third histogram). The histograms tend to display a typical long tail effect, where most individuals and markers have few or no errors, while a handful have many.

In the bottom half of the interface, a pedigree *sandwich view* (b) [6] shows relationships between offspring and their parents in each generation. The two tables (c) to the right of the sandwich view give detailed information on errors for every individual and marker in the data set, sub-dividing errors into types (to sire, dam, and novel alleles). Missing data is subdivided into *incomplete* – missing in the initial data set – and *masked* – data deliberately hidden by the user to remove an error. Selecting a marker in the table switches the application into single marker view, showing only the errors and maskings for that particular marker. Individual markers can also be masked through this table. A branching history widget (d) allows restoration of previous states, essential to an exploratory interaction with the data.

Additionally, at the top of the interface an info bar displays statistics about the data including the count of genotypes currently in error. These views are coordinated in a typical multiple view fashion (the one exception being the initial histogram data view).

Two user-defined hues are used to convey a) erroneous and b) masked/incomplete data - in the figures we consistently use red for error and blue for masked/missing values. As the error cleaning progresses, the amount of error hue in the interface should decrease, to a target of disappearing altogether, whereas the second hue will gradually come to the fore as more and more data points are masked and flagged as unknown values.

3.3 Data Cleaning Operators

Error removal in pedigree genotype data means to mask erroneous data points i.e. to change data from definitely wrong to an unknown state. In essence, we say "we don't know what it is, but it's definitely not that." Whereas out-and-out errors are generally fatal to any further downstream analyses, unknown data points can be effectively handled by inferring across the lack of data; algorithms used in pedigree genotype analysis can skip across the missing points in the pedigree structure and attempt to match up the data on either side. So, the aim isn't to try and deduce what the wrong data should be, but simply to remove the effect it has on analyses. Generally there is enough correct information left after error cleaning that getting the correct data for every point is not necessary, and such an ideal often proves intractable in any case.

Masking can take the form of several operations:

1. Mask a marker
2. Mask an individual
3. Mask a genotype (marker/individual combo)
4. Break a child/parent relationship in the pedigree.

Markers may be removed or masked completely from analyses with no side effect on other markers. However, individuals are obviously related within a pedigree, and masking just one individual can have a range of side-effects, and those side-effects occur for every single marker that is layered on top of the pedigree. In fact, the operation to break a pedigree relationship

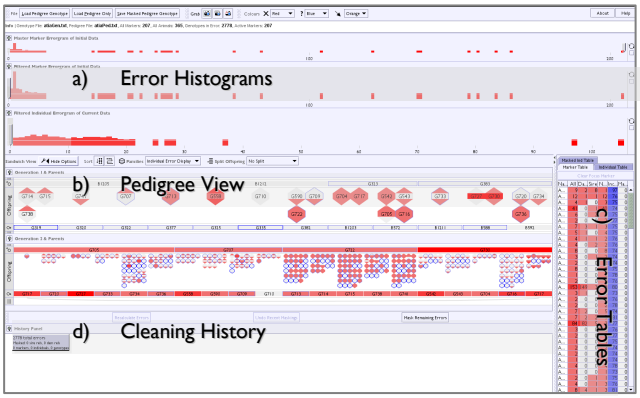

Figure 1. The main components of the pedigree data cleaning application: a) histograms of error distribution, b) a pedigree sandwich view, c) error tables, and d) a history function.

was introduced to nullify pedigree errors which couldn't be solved with the initial three operations. Often, and especially for pedigree errors, masking an individual or genotype just projects erroneous information through the masked individual onto its relations. In an obvious case, if a child is assigned to the wrong parent, then its own children in turn have been linked to the wrong grandparent.

Of these four operators, one and four do not introduce further errors into the system, but two and three do introduce this possibility. All four operators may be carried out multiple times before a recalculation of the current error state is performed.

3.4 Typical Error Removal

The user interface, as seen in Figure 1, is laid out such that the tendency is for the user to first perform operations with the histograms at the top of the application window and move down to the pedigree sandwich component once troublesome markers have been removed. The rationale here is that portions of the data with typically large concentrations of error, such as bad markers, be removed before masking individuals and individual genotypes. After each masking or removal operation, the underlying error checking algorithm can recalculate the new error states, which the visualization then displays. Figure 2 shows a typical path through the interface when cleaning a pedigree genotype, consisting of four sub-tasks: mask bad markers, mask bad individuals (or break their pedigree relationships), iteratively mask genotypes on a per marker basis, and remove final sporadic errors.

3.4.1 Mask bad markers with histograms

The top histogram, showing initial errors across the marker set, has a slider that can be used to mask markers with excessively high levels of error, and this exclusion is reflected in all the components below it. This makes for a logical first step in the data cleaning process – as stated earlier, markers are independent of each other so removal of groups of whole markers does not introduce new errors into the data. This step usually removes a large proportion of the initially reported errors; in Figure 2 the first step of removing just 11 markers from a set of 207 has halved the number of reported errors.

The second histogram now shows the remaining markers from the first histogram. The effect is akin to zooming in on the unfiltered histogram portion from the first histogram. The third histogram shows the distribution of current errors across the set of individuals, and will change according to what markers are currently omitted and which further masking operations are performed in the other components. Here, the slider doesn't mask

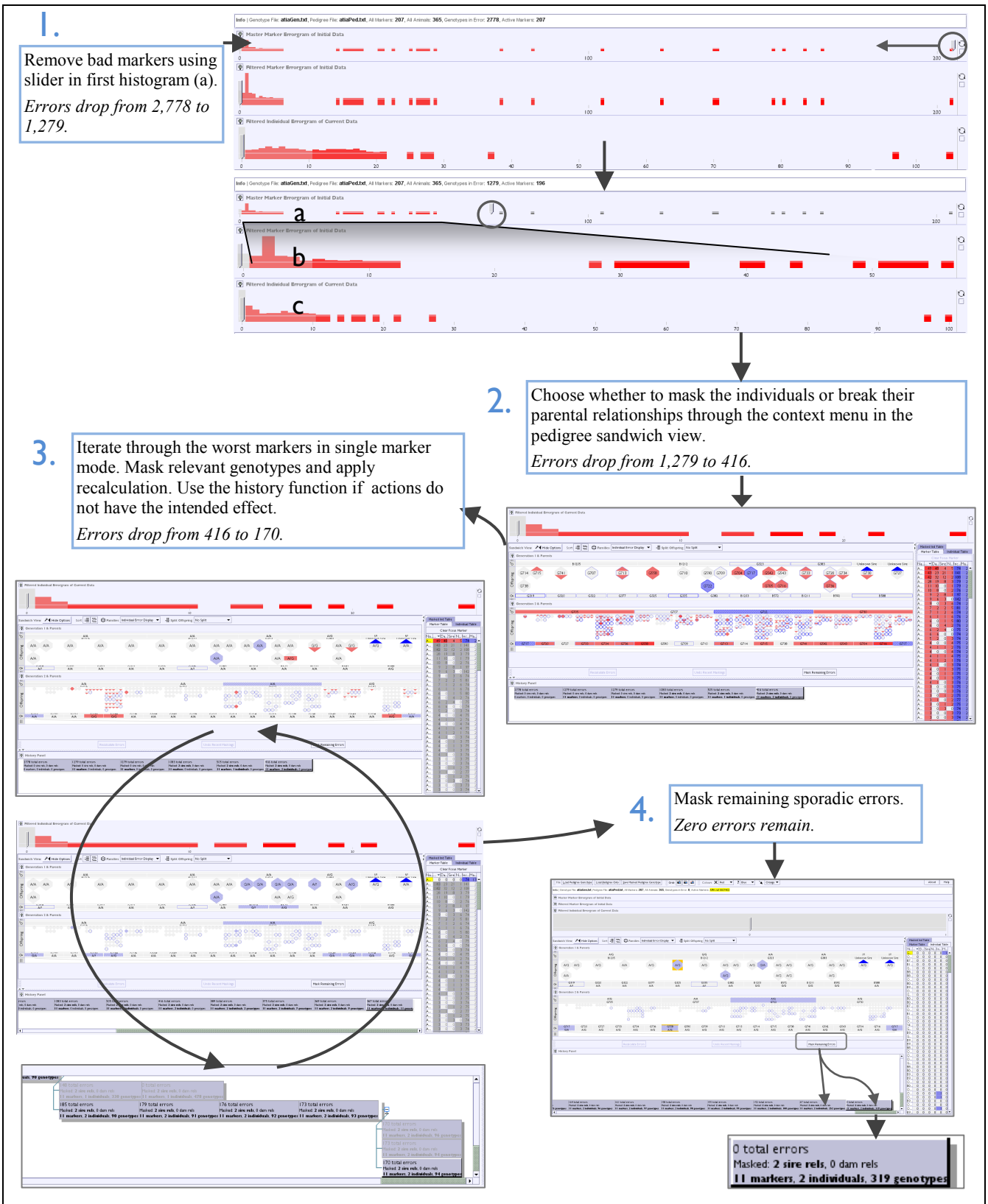

Figure 2. Workflow showing a typical path through the interface to clean an example pedigree genotype data set. (1) Bad markers are removed first, followed by (2) bad individuals via masking and breaking of parental relationships in the pedigree. (3) Remaining problem markers are iteratively dealt with in the single marker view. Finally (4) the "mask all remaining errors" functionality is used (in this case twice) to purge sporadic errors. The result is a data set with zero errors that can then be saved.

but offers a quick visual filter: greying out any individuals with less than the slider's threshold number of errors, and accentuating the most problematic individuals in the pedigree display.

3.4.2 Mask bad individuals in sandwich view

Once the most error-ridden markers have been discarded then a user can start cleaning individuals within the pedigree sandwich view. The most error-ridden individuals in this view will have deeper shades of the hue assigned to indicating error, but can be further identified by either:

a) Using the grey cut-off slider in the individual histogram.
b) Sorting the individual error table by number of errors and then selecting the top-most items in the list. This will highlight the corresponding individuals.
c) Sorting the pedigree view itself by various error count metrics within each generation and then within each family.

Once identified, selecting an individual brings up a context-sensitive menu that allows a user to (un)mask individuals or whole families and also to break/restore an individual's parental relationships. After a selection of pertinent maskings and broken relationships has been made then a recalculation is performed by selecting the option in the bar below the pedigree view. At step 2 in Figure 2, after two individual maskings and two relationship removals, the errors have reduced further from 1,279 to 416.

Previously, recalculation was automatic after every user operation but this caused slowdown when dealing with large data sets. Also, nullifying parental relationships causes the structure of the pedigree itself to change, which entails a change to the pedigree visualisation. This was found to be disconcerting when not specifically expected, so user feedback was to have recalculation happen on demand by the user. In Figure 2, the screenshot at step 2 reveals that the two individuals who had paternity relationships broken have been reassigned to "null sire" as a parent, and the pedigree representation has changed accordingly.

3.4.3 Mask bad genotypes, marker by marker

Once bad markers and individuals have been dealt with, we need to remove errors on a per marker basis. Sorting the marker table by errors per marker allows efficient targeting of the most error-ridden remaining markers. Selecting a marker here now switches the pedigree viewer to "single marker" mode, where only errors for that particular marker are shown, and the individuals can display genotype values for that particular marker. This often reveals blocks of error within the pedigree that can be tackled using minimal and judicious genotype masking – for instance a group of offspring all reporting errors can often all be solved by masking just one of the parent genotypes. It is notable that the masking is often not on data that is itself in error, but on a value that causes error in offspring. Iteratively tackling the worst remaining markers in this fashion gradually erodes the number of remaining errors; in the example in Figure 3 we reduce the count to around 170 by the end of this third step.

3.4.4 Final clean

There will usually remain a long tail of many markers with a few errors each, and a "Mask Remaining Errors" function is available to automatically mask these sporadic genotype errors. As displacement of error is often caused by genotype masking, a new set of errors is often the result of this function, so it may take multiple invocations to purge the data set completely of errors.

3.4.5 Export

The final step in the data cleaning process is to save the error-free versions of the data as separate pedigree and genotype data files. A log file of markers, individuals, genotypes and relationships that have been masked is also saved to the same directory.

4. CONCLUSION

Pedigree genotype data sets are often riddled with error, many of which depend on and are propagated by inheritance patterns within the pedigree structure. However, data cleaning solutions for these data sets have not focused on using a pedigree-centric resolution nor used aggregate views such as histograms to ease data cleaning efforts on the part of the user. Our system allows users to methodically remove erroneous data from such data sets starting with large blocks of error down to individual genotypes.

The pedigree view provides users the ability to see errors in the context of the pedigree structure, enabling them to make informed decisions about which masking operations would be best suited to remove given areas of error. A history function and associated view allows actions to be undone, encouraging an exploratory "what-if?" approach to error removal within these datasets.

The obvious question is why attempt interactive cleaning if we can clean the data set automatically? The answer is that an expert biologist can often spot places in the pedigree where a single operation can carry out the effect of many operations elsewhere in the structure, and also decide which operation makes sense in a given context and will not propagate errors. Also, if someone is to fundamentally change a dataset then they really need to know and understand what they've done to it beyond press a button. All of which is beyond a current automatic masking function.

5. ACKNOWLEDGMENTS

We would like to thank the UK BBSRC (Biotechnology and Biological Sciences Research Council) for funding this project.

6. REFERENCES

[1] Kandel, S., Paepcke, A., Hellerstein, J. and Heer, J. 2011. Wrangler: Interactive Visual Specification of Data Transformation Scripts. In *Proceedings of the ACM Human Factors in Computing Systems* (Vancouver, Canada, May 07-12, 2011). CHI '11. ACM, 3363-3372. DOI= http://doi.acm.org/10.1145/1978942.1979444.

[2] Bilgic, M., Licamele, L., Getoor, L. and Shneiderman, B. 2006. D-Dupe: An Interactive Tool for Entity Resolution in Social Networks. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology* (Baltimore, USA, October 31 - November 02, 2006). VAST '06. IEEE, 43-50. DOI= http://dx.doi.org/10.1109/VAST.2006.261429.

[3] Griethe, H. and Schumann, H. 2006. The Visualization of Uncertain Data: Methods and Problems. In *Proceedings of the Simulation und Visualisierung 2006* (Magdeburg, Germany, March 02-03, 2006). SimVis '06. SCS, 143-156.

[4] Sanyal, J., Zhang, S., Bhattacharya, G., Amburn, P. and Moorhead, R. J. 2009. A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasets. *IEEE T. Vis. Comput. Gr.* 15, 6 (November 2009), 1209-1218. DOI= http://dx.doi.org/10.1109/TVCG.2009.114

[5] Paterson, T. and Law, A. 2011. GenotypeChecker: An interactive tool for checking the inheritance consistency of genotyped pedigrees. *Animal Genetics*. 42, 5(2011), 560-562. DOI= http://dx.doi.org/10.1111/j.1365-2052.2011.02183.x.

[6] Graham, M., Kennedy, J., Paterson, T. and Law, A. 2011. Visualising Errors in Animal Pedigree Genotype Data. *Comput. Graph. Forum*. 30, 3 (June 2011), 1011-1020. DOI= http://dx.doi.org/10.1111/j.1467-8659.2011.01950.x.