An Iterative Approach to Unsupervised Pedigree Layout

Curran Kelleher, Dr. Brian Drohan,
Dr. Kevin Hughes, Dr. Georges Grinstein

Abstract

We introduce a novel unsupervised pedigree layout algorithm.

1 Introduction

The problem of unsupervised pedigree layout has been a long standing computational challenge. Attempts have been made to automate pedigree layout, but most of them are limited in scope, and are not comprehensive general solutions. We introduce an iterative algorithm for deriving aesthetically optimal pedigree layouts. Our hope is to define algorithm which is guaranteed to layout all legal pedigrees correctly, and which can be extended with manual intervention to resolve impossible pedigrees.

1.1 Our Algorithm

An *individual* is a representation of a person. Given an individual $i \in I$, the parents of i are denoted by $mother[i]$ and $father[i]$ (these return NIL when no parent is known). The input to our agorithm is a set I of individuals. Our algorithm computes a pedigree layout based only on this information. The output of our algorithm is a mapping from each individual $i \in I$ to an (x, y) location. Once the locations are in place, drawing the pedigree is trivial. In summary, the input to our algorithm is a collection of individuals, and the output of our algorithm is a collection of (x, y) coordinates associated with each individual, which can be used to draw the pedigree.

Our algorithm begins with a step which derives the *couples graph* from the input individuals. A *couple* is a pair of individuals who are parents of a sibship. A *sibship* is a collection of siblings who are children of a common couple. We define the *couples graph* as the undirected graph connecting all couples which are connected by a common individual. An edge between two couples c_1 and c_2 is added to the couples graph in two cases: 1. when a couple c_1 has a child which is part of another couple c_2 (this rule adds edges between grandparents and parents), and 2. when and individual i participates in multiple different couples c_1 and c_2 (in this case, a child from c_1 and a child from c_2 are half siblings whose common biological parent is the individual i).

The remaining tasks of our algorithm are as follows: 1. assign initial (x, y) coordinates and (dx, dy) velocity vectors to all individuals as well as couples; 2. generate a list of *layout steps* based on the individuals and couples; and 3. execute those steps repeatedly until the layout converges.

A *layout step* is defined as a computation which takes as input the state of the layout, and provides as output the next state of the layout. The *state of the layout* is defined as the set of (x, y, dx, dy) vectors associated with each individual and each couple, as well as the list of currently active layout steps. Layout steps are chained together such that the output of one step is the input for the next step.

After introducing the concepts of *individuals*, the *couples graph*, and iterative *layout steps*, the task before us was to design a set of layout steps which will correctly lay out a pedigree.

All correctly drawn pedigrees have planar embeddings of the couples graph. This can be observed by taking any pedigree diagram, drawing dots in the middle of each couple (directly in the middle of the mother-father pairs), and drawing lines between couples where an edge would be added to the couples graph (either between parents and grandparents or in the case of half siblings). No edges cross. Because of this observation, our first objective was to ensure a planar embedding of the couples graph. We accomplished this by generating a set of layout steps which execute a force-directed layout of the couples graph.

A planar embedding of the couples graph is only half the battle. Pedigrees demand a couples graph layout that is not only planar, but one in which couples belonging to the same generation are on the same horizontal line. Luckily, the generational levels for each couple can be computed from the couples graph. Using these levels, we address the issue at hand by adding layout steps which restrict the movement of couples to stay within the correct levels. At the beginning of the algorithm (when the planar arrangement is being established by the force-directed layout steps), these restrictive rules are turned off to allow couples to roam freely. Over time (after the planar layout has been reached) these restrictive rules are slowly put into effect, pulling the planar layout into the correct levels.

Once we have a set of layout steps which converges to a planar embedding of the couples graph, we can derive from that layout state a set of layout steps for appropriately placing individuals.

The first critical observation we made is that the embedding of the couples graph in the plane must be planar in order for the pedigree to be correct.

2 Raw material for the paper

We denote the couples graph $C = (V_C, E_C)$ where V_C is the vertex set and E_C is the edge set. Each vertex in the couples graph represents a couple defined by a unique $(motherId, fatherId)$ pair. Accessing a couple in $V[C]$ is denoted by $V[C][motherId, fatherId]$. Accessing the mother and father ids for a given couple $c \in V[C]$ is denoted as $motherId[c]$ and $fatherId[c]$. The input individuals are represented as a map I where $I[id]$ is the individual with id *id*. The following pseudocode derives the couples graph from the input individuals map:

```
DERIVE-COUPLES-GRAPH($I$)
1  $V[C] = \emptyset$, $E[C] = \emptyset$
2  for each individual $child \in I$
3        if $V[C][motherId[child], fatherId[child]]$ doesn't exist
4              $V[C]$
```

One advantage of our approach is that one can watch the layout incrementally occur. This provides a dynamicity of layout not found in existing pedigree visualization systems which allows our system to be extended to support interactive data manipulation, such as adding or removing people in the pedigree, while the layout is occurring.

In order to use a consistent vocabulary throughout this paper, here we define our terms:

- id – A unique identifier associated with an individual.

- individual – A node in the pedigree with the an id, mother id (mid), father id (fid), and gender. An individual is associated with the following attributes at various stages of the layout algorithm: (x, y) location, level number, spouse, and sibling set.
- level – A horizontal level of the pedigree. Levels are numbered such that the highest level number is in the southernmost position. In relation to levels, high and low refer to level numbers, not high and low as in north and south in the graphical layout.
- location – Will always refer to the (x, y) screen coordinates of an individual.
- parent-child relationship – An edge in the pedigree between a child and one of its parents.
- family – A composite object containing one or two parents and their one or more children. This entity has been referred to as a 'subtree' by Coulsen et. al. [**?**].
- sibling set – A composite object containing a set of individuals who are all siblings of one another. The children part of a family object is a sibling set.

3 Related Work

3.1 RAGS

A. S. Coulson et. al. introduced a layout algorithm for simple pedigrees. [**?**]. Their algorithm consists of two phases:

- Derive families from known parent-child relationships.
- Repeatedly join each family visually, assigning locations to individuals based on their family placement until locations of all individuals have been assigned.

4 Our Algorithm

Our pedigree layout algorithm takes as input a set of individuals with an id, a mother id, and a father id, and yields as output a set of individuals annotated with (x, y) locations, as well as the set of derived families.

References

[1] A. S. Coulson, D. W. Glasspool, J. Fox, and J. Emery. Rags: A novel approach to computerized genetic risk assessment and decision support from pedigrees. In *Methods of Information in Medicine*, pages 315–322, 2001.